Доклады по компьютерным наукам и информационным технологиям **01**

Издается с 2012 года

Редакционный совет

Дмитрий Игнатов
Ростислав Яворский (редакторы)

Доклады всероссийской научной конференции АИСТ'12

Модели, алгоритмы и инструменты анализа данных; результаты и возможности для анализа изображений, сетей и текстов

Екатеринбург, 16 – 18 марта 2012 года

ИНТУИТ
НАЦИОНАЛЬНЫЙ ОТКРЫТЫЙ УНИВЕРСИТЕТ

Редакторы тома

Дмитрий Игнатов

Ростислав Яворский

УДК [004.738.5+004.9](063)
ББК 32.973.202я431(2Рос)+32.973.26-018я431(2Рос)
Д63

ISSN

ISBN 978-5-9556-0132-8

Доклады Всероссийской научно-практической конференции «Анализ Изображений, Сетей и Текстов» (АИСТ, Екатеринбург, 2012). Рассматриваются проблемы в области компьютерного зрения, анализа изображений и видео, анализа форумов, блогов и социальных сетей, анализ сетевых (графовых) и потоковых данных, компьютерной обработки текстов, гео-информационных систем, математических моделией и методов анализа данных, машинного обучения и разработки данных (Data Mining), рекомендательных систем и алгоритмов, Semantic Web, онтологии и их приложений.

Для студентов, аспирантов и специалистов в области компьютерной графики, машинного зрения и обработки изображений.

.

Предисловие

В сборнике представлены работы участников Всероссийской научно-практической конференции «Анализ Изображений, Сетей и Текстов» (АИСТ 2012). Это мероприятие стало площадкой, которая позволила студентам, аспирантам, специалистам и ученым математических, технических, лингвистических, географических, социологических и иных специальностей представить результаты своих работ и расширить познания в области анализа данных, обменяться опытом.

Конференция проводилась с 16 по 18 марта 2012 года в столице Урала – Екатеринбурге. Все статьи можно условно разбить на несколько групп по темам:

- Компьютерное зрение, анализ изображений и видео
- Анализ форумов, блогов и социальных сетей
- Анализ сетевых (графовых) и потоковых данных
- Компьютерная обработка текста
- Гео-информационные системы
- Математические модели и методы анализа данных
- Машинное обучение и разработка данных (Data Mining)
- Рекомендательные системы и алгоритмы
- Semantic Web, онтологии и их приложения

Всего было получено 45 заявок, каждая из которых была оценена минимум двумя рецензентами. По итогам рецензирования 23 работы были отобраны для секционных докладов и 17 для постерных сессий. В программу конференции включены три мини-курса и две лекции, прочитанные приглашёнными докладчиками, а также презентации компаний организаторов и спонсоров конференции.

Пользуясь этой возможностью, мы выражаем благодарность всем организаторам, членам программного комитета, рецензентам, докладчикам, спонсорам и партнёрам конференции, благодаря которым эта конференция состоялась. Мы благодарны Национальному Открытому Университету «ИНТУИТ» за помощь в издании тома трудов конференции.

Март 2012

Дмитрий Игнатов
Ростислав Яворский

Программный комитет конференции

Координаторы

Дмитрий Игнатов, НИУ ВШЭ, Россия
Ростислав Яворский, Witology, Россия

Члены

Ольга Баринова, МГУ, Россия
Виктор Бочаров, СПбГУ, Россия
Павел Браславский, СКБ Контур, Россия
Александр Вохминцев, ИИТ ЧелГУ, Россия
Борис Галицкий, Университет Жироны, Испания
Дарья Гончарова, Witology, Россия
Дмитрий Грановский, Яндекс, Россия
Леонид Дворянский, НИУ ВШЭ, Россия
Максим Дубинин, NextGIS, Россия
Виктор Ерухимов, ЦКЗ Аргус, Нижний Новгород
Леонид Жуков, НИУ ВШЭ, Россия
Вадим Канторов, École Normale Supérieure de Cachan, Франция
Юрий Катков, СПб НИУ ИТМО, Россия
Никита Козин, Университет Райса, США
Андрей Константинов, НИУ ВШЭ, Россия
Дмитрий Корнев, УрФУ, Россия
Сергей Кузнецов, НИУ ВШЭ, Россия
Алексей Лахно, НИУ ВШЭ, Россия
Виктор Лемпицкий, Яндекс, Россия
Алексей Незнанов, НИУ ВШЭ, Россия
Сергей Объедков, НИУ ВШЭ, Россия
Йонас Пульманс, Католический Университет Левена, Бельгия

Сергей Рогожкин, Microsoft, Россия
Александра Савельева, НИУ ВШЭ, Россия
Александр Семенов, НИУ ВШЭ, Россия
Павел Сердюков, Яндекс, Россия
Никита Спирин, Университет Иллинойса, США
Алексей Станкевичус, НИЯУ МИФИ, Россия
Рустам Тагиев, Технический университет Фрайберга, Германия
Олег Ушмаев, Институт проблем информатики РАН, Россия
Михаил Хачай, ИММ УрО РАН и УрФУ, Россия

Приглашенные рецензенты

Лидия Пивоварова, СПбГУ, Россия
Константин Блинкин, НИУ ВШЭ, Россия
Наталия Жукова, СПбГТУ «ЛЭТИ», Россия
Александра Каминская, НИУ ВШЭ и Witology, Россия
Никита Ромашкин, НИУ ВШЭ, Россия
Федор Строк, НИУ ВШЭ, Россия
Екатерина Черняк, НИУ ВШЭ, Россия
Ольга Чугунова, НИУ ВШЭ, Россия

Организационный комитет конференции

Секретарь

Александра Каминская, НИУ ВШЭ и Witology, Россия

Члены

Дарья Гончарова, Witology, Россия
Ирина Войчитская, Яндекс, Россия
Мария Рудниченко, СКБ Контур, Россия
Никита Спирин, Университет штата Иллинойс, США

Спонсоры и партнеры конференции

Национальный исследовательский университет Высшая школа экономики

Национальный Открытый Университет «ИНТУИТ»

Witology

Яндекс

СКБ Контур

Уральский федеральный университет имени первого Президента России Б.Н. Ельцина

Исследовательский центр моделирования, анализа и тестирования «Моданте»

3DiVi Company (ООО «ТРИДИВИ»)

Издательство «Открытые системы»

Оглавление

Приглашенные доклады

Секционные доклады

Постерные доклады

Компьютерное зрение

Ольга Баринова

v-olbari@microsoft.com

119992 ГСП-2, Москва, Воробьевы горы, МГУ им. М.В.Ломоносова

Аннотация. Современные системы компьютерного зрения позволяют распознавать жесты, восстанавливать трехмерную структуру сцены по двумерному изображению, выделять объекты переднего плана от фона, надежно обнаруживать объекты определенного класса. Эти достижения во многом обязаны появлению графических моделей, которые позволяют с одной стороны объединять информацию из различных частей и элементов изображения в единую модель, а с другой стороны использовать глобальные ограничения реального мира. В мини-курсе мы коснёмся следующих вопросов: Что такое компьютерное зрение, какие задачи оно позволяет решать? Что такое графические модели? Какие графические модели используются в современных системах компьютерного зрения? Для иллюстрации мы рассмотрим последние совместные проекты МГУ и Microsoft Research по компьютерному зрению.

Ключевые слова: компьютерное зрение, графические модели, системы компьютерного зрения.

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16-18 марта, 2012.

Геоинформационные системы

Максим Дубинин

sim@gis-lab.info

NextGIS, 117312, Москва, Вавилова 41

Аннотация. В докладе будут рассмотрены следующие вопросы, касающиеся геоинформационных систем. 1) Геоданные: особенности, основные источники, ПО для работы с ними. 2) Непараметрические классификаторы для анализа данных дистанционного зондирования. 3) Методы максимальной энтропии для пространственного нишевого анализа.

Ключевые слова: геоинформационные системы, геоинформатика, анализ геоданных.

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16-18 марта, 2012.

Анализ формальных понятий: от теории к практике

Д. И. Игнатов

dignatov@hse.ru

НИУ ВШЭ, Россия, 101000, г. Москва, ул. Мясницкая, д. 20

Аннотация. В работе даются основные определения анализа формальных понятий (АФП), рассказывается о его роли в математике и компьютерных науках, а также приводится краткий обзор его основных приложений.

Ключевые слова: анализ формальных понятий, разработка данных (Data Mining), приложения.

Введение

Анализ формальных понятий (АФП) является прикладной ветвью алгебраической теории решеток, в рамках которой предложен математический формализм, описывающий на языке алгебры понятие и иерархии понятий. Основные идеи АФП был сформулированы Рудольфом Вилле в его работе [50], а наиболее полной монографией по АФП является книга Гантера и Вилле [23].

Фактически анализ формальных понятий имеет дело с данными в объектно-признаковой форме, а формальные понятия, определенные с помощью соответствия Галуа, представляют собой пары множеств вида (объем, содержание), им в точности до перестановки строк и столбцов соответствуют максимальные прямоугольники в таблице объект-признак. Основными достоинствами такого определения понятия являются соответствие традиционным представлениям о понятиях исполь-

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16-18 марта, 2012.

зуемым в философии: 1) понятие — это пара вида (объем, содержание), 2) при уменьшении объема понятия увеличивается его содержание и наоборот, 3) понятия иерархически упорядочены по отношению «быть более общим понятием».

За последние 30 лет АФП прошел значительный путь от первоначальных теоретических изысканий к разнообразным многочисленным приложениям (только на английском языке издано около 900 научных работ по тематике АФП, более половины из которых посвящены приложениям), что позволяет полноправно назвать его прикладной математической дисциплиной. Основными приложениями АФП, которым мы уделим внимание в этой работе, являются анализ данных (машинное обучение и разработка данных), представление знаний (онтологии и таксономии), информационный поиск, анализ неструктурированных данных (в частности, текстов), программная инженерия, социология и образование. В настоящее время существуют три наиболее репрезентативных международных конференции по тематике АФП: International Conference on Formal Concept Analysis, International Conference on Concept Lattices and Their Applications и International Conference on Conceptual Structures. Первая в списке конференция является наиболее представительной и служит для обсуждения значительных теоретических и практических результатов в области, вторая посвящена преимущественно приложениям АФП, а третья, помимо АФП-сообщества, призвана собрать исследователей в области представления знаний и онтологического моделирования (например, сооснователем этой серии конференций является создатель понятийных графов Джон Сова).

Основные определения анализа формальных понятий

Контекстом в АФП называют тройку $K = (G, M, I)$, где G — *множество объектов*, M — *множество признаков*, а отношение $I \subseteq G \times M$ говорит о том, какие объекты какими признаками обладают. Для произвольных $A \subseteq G$ и $B \subseteq M$ определены *операторы Галуа*:

$$A' = \{m \in M \mid \forall g \in A\ (g\ I\ m)\};$$

$$B' = \{g \in G \mid \forall m \in B\ (g\ I\ m)\}.$$

Оператор '' (двукратное применение оператора ') является *оператором замыкания*: он идемпотентен ($A'''' = A''$), монотонен ($A \subseteq B$ влечет $A'' \subseteq B''$) и экстенсивен ($A \subseteq A''$). Множество объектов $A \subseteq G$, такое, что $A'' = A$, называется *замкнутым*. Аналогично для замкнутых множеств признаков — подмножеств множества M. Пара множеств (A, B), таких, что $A \subseteq G$, $B \subseteq M$, $A' = B$ и $B' = A$, называется *формальным понятием* контекста K. Множества A и B замкнуты и называются *объемом* и *со-*

держанием формального понятия (A, B) соответственно. Для множества объектов A множество их общих признаков A' служит описанием сходства объектов из множества A, а замкнутое множество A'' является кластером сходных объектов (с множеством общих признаков A'). Отношение «быть более общим понятием» задается следующим образом: (A, B) ≥ (C, D) тогда и только тогда, когда A⊇C. Понятия формального контекста K = (G, M, I), упорядоченные по вложению объемов образуют решетку $\underline{B}(G, M, I)$, называемую *решеткой понятий*. Для визуализации решеток понятий используют т.н. диаграммы Хассе, т.е. граф покрытия отношения «быть более общим понятием».

АФП в машинном обучении и разработке данных

В этом разделе мы кратко опишем основные приложения и методы на основе АФП в области современного анализа данных, в частности в разработке данных (Data Mining).

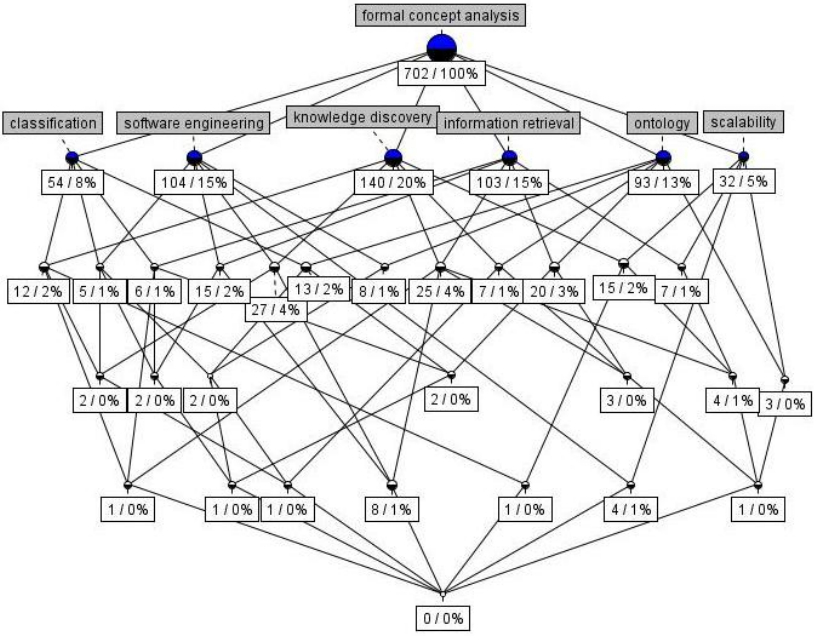


Рис. 1. Диаграмма решетки понятий для 702 статей по АФП, охватывающих 2003 — 2009 годы.

Классификация

Соответствия Галуа также использовались некоторыми исследователи в СССР, например, ДСМ-метод автоматического порождения гипотез, применяемый для решения задач классификации довольно естественно формулируется в терминах решеток понятий как метод машинного обучения по положительным и отрицательным примерам [11]. Переход в данном случае с языка математической логики к алгебраическим формулировкам позволил разработать эффективные программные реализации ДСМ-метода. На языке формальных понятий и соответствий Галуа переформулированы такие методы машинного обучения как пространства версий и деревья решений (см. [22]). Более поздняя работа [14] по применению решеток понятий для порождения деревьев решений на основе решеток формальных понятий показала улучшение результатов по сравнению с известными методами, такими как C4.5 и IB1.

Отбор признаков

Отбор признаков (feature selection), сокращение или редуцирование признаков на этапе предварительного анализа данных существенно помогают сократить не только вычислительные затраты, но и улучшить точность классификации. Сплав идей анализа формальных понятий и неточных множеств (Rough Sets) убедительно демонстрирует такое сокращение [25].

Частые (замкнутые) множества признаков

Поиск частых множеств признаков возник как направление в разработке данных в начале 90-х годов для решения задач анализа корзины покупок в крупных продуктовых супермаркетах. Анализ формальных понятий и поиск (замкнутых) множеств признаков (Frequent Itemset Mining) объединяет идея замыкания. Фактически решетка понятий некоторого формального контекста в АФП изоморфна решетке замкнутых множеств, если требование быть частым множеством не учитывать. В АФП было введено понятие решетки-айсберга [47], которое позволило максимально сблизить терминологию двух сообществ FIM и FCA.

Поиск закономерностей: импликации, ассоциативные правила и функциональные зависимости

Импликацией формального контекста $K = (G, M, I)$ в анализе формальных понятий называется признаковая зависимость вида $A \rightarrow B$, где $A, B \subseteq M$, при условии, что все объекты, обладающие A, также облада-

ют всеми признаками из B, т.е. A' $\subseteq$ B'. Импликация в АФП является частным случаем такой признаковой зависимости как ассоциативное правило в разработке данных, это в точности ассоциативное правило с достоверностью (confidence) равной 1. В свою очередь, ассоциативные правила изучались в АФП задолго до их появления в сообществе разработки данных под названием частичные импликации [37]. Связь импликаций и функциональных зависимостей позволила использовать т. н. базис импликаций Дюкена-Гига для компактного представления функциональных зависимостей (см. теорию баз данных) виде их ограниченного множества, из которого все оставшиеся функциональные зависимости данного многозначного контекста (таблицы данных) выводимы по правилам Армстронга [23]. Достаточно полный обзор по поиску ассоциативных правил на основе АФП можно найти в работе [36].

Модели мультимодальной кластеризации

Недостатки традиционных методов кластеризации, связанные с потерей признакового описания сходства объектов, при установлении факта их числового сходства требуют новых методов кластерного анализа во многих приложениях, таких как анализ данных генной экспрессии и Интернет-данных. Формальные понятия могут быть рассмотрены как своего рода бикластеры, в которых описание сходства объектов сохраняется в признаковой компоненте бикластера — содержании [6, 8]. Стоит отменить многочисленные попытки ослабления определения формального понятия и его обобщения на многомерный случай. Одними из успешных таких попыток являются разработка метода поиска мультимодальных кластеров DataPeeler [38] и плотных би- [6, 8] и трикластеров [4, 5, 30].

Рекомендательные системы

Рекомендательные системы также потенциальные кандидаты для применения АФП, первые шаги в этом направлении были сделаны в работах [1, 7, 28].

Приложения в анализе текстов

Анализ формальных понятий помогает также в анализе неструктурированных данных. Например, для выявления (почти) дубликатов по большим коллекциям веб-документов [9, 10, 29] и анализа текстов полицейских отчетов [41]. Основное преимущество перед методами кластеризации на основе попарного сравнения документов в хорошей эмпирической временной сложности при кластеризации текстовых коллекций

благодаря разреженности данных. Во втором приложении важным для экспертов являются таксономические возможности решеток понятий, позволяющих удобно изучать коллекции полицейских отчетов по диаграмме решетки понятий, построенной по таблице отчеты – ключевые слова [41].

Приложения в программной инженерии

Пожалуй, впервые систематическое обсуждение приложений АФП в программной инженерии было дано в книге [25]. В основном АФП применяется для поддержки разработки ПО и объектно-ориентированного моделирования иерархий классов на ранних стадиях проекта, а также для улучшения и рефакторинга кода на более поздних этапах (см. статьи [26, 27, 46, 49]). Позднее появился обзор 47 статей по программной инженерии на основе АФП [48]. Авторы разбили эти статьи по 10 категориям на основании стандарта программной инженерии ISO 12207 и визуализировали результаты анализа с помощью диаграммы решетки понятий.

АФП в онтологическом моделировании и представлении знаний

Таксономические свойства решеток понятий, представление множества понятий в виде иерархии с отношением «быть более общим понятием» ставят естественный вопрос насколько тесно АФП связан с онтологиями. Ответ на него был дан достаточно давно в работах Ф. Симиано и А. Хотхо (исследователь из университета Касселя, Германия) [17]. Было установлено как можно получить частичный порядок менее строгий, чем решеточный, из решеток понятий, и, наоборот, как по имеющейся онтологии, представленной в виде частичного порядка на понятиях, построить решетку понятий. АФП тесно связан с описательными логиками (Descriptive Logic), например, так называемое исследование признаков (Attribute Exploration), как метод пополнения баз знаний был позаимствован сообществом DL из АФП [13].

Важной темой в работах по АФП является вопрос построения онтологий эффективным образом. Этой теме посвящено около 30% всех статей по АФП (всего 93 статьи за период с 2003 по 2009 год). Авторы используют АФП преимущественно как средство извлечения онтологических понятий и их иерархий. Большинство из них имеют дело с неструктурированными текстами, такими как медицинские отчеты, RSS потоки, научные статьи и т.п. Анализируя неструктурированные тексты авторы как правило используют средства обработки естественного языка (NLP). С помощью NLP они извлекают из текстовых коллекций клю-

чевые слова, фразы, лексико-синтаксический контекст и т.п. По таким данных можно построить решетки понятий и извлекать онтологические классы ключевых слов, иерархически упорядочивать эти понятия, выявлять зависимости между классами и т.п. В итоге новое онтологическое знание может быть сохранено, например, в формате OWL, а новые тексты могут быть классифицированы с использованием уже этой онтологии. Именно с появлением работ Симиано, Хотхо и др. (см. [17] и [18]) АФП стал популярным инструментом для построения онтологий. Работа [17] обсуждает как АФП может быть использован для поддержки построения онтологий и как онтологии могут быть использованы в приложениях АФП. Ричардс [43] предлагает использовать АФП для построения небольших персональных и ad hoc, которые могут помочь пониманию области исследований.

Таксономические свойства АФП оказались удобными для представления знаний, например, при анализе посещаемости сайтов в сети Интернет для построения таксономий аудиторий веб-сайтов [33].

Информационный поиск

Среди приложений АФП по информационному поиску можно отметить мета-поисковые системы для Интернета [16, 19, 32]. Для более детального знакомства с предметом рекомендуется обратиться к книге Карпинето и Романо [15] или еще вполне актуальному обзору Уты Присс [42].

Социологические приложения и анализ образовательных данных

Ключевыми фигурами по приложениям АФП в социологии являются Линтон Фриман и Винсент Дюкен. Линтоном Фриманом изучались возможности решеток понятий для определения сообществ в анализе социальных групп и сетей [21], а Винсентом Дюкеном сделано немало для социологических и антропологических исследований на основе опросных данных [20, 39 и 40]. Исследованием эпистемических сообществ интенсивно занимались Сергей Объедков и Камий Рот [45]. Анализу результатов социологических опросов и данных в области образования посвящены работы автора этой статьи [2, 3, 31 и 44]. Работа [4] посвящена изучению три-сообществ в социальных Интернет-сервисах.

Заключение

Таким образом, можно сделать вывод, что АФП является бурно развивающейся дисциплиной на стыке прикладной математики и компью-

терных наук, а математическая формализация понятия оказала свое благотворное влияние на анализ данных, представление знаний и различные разделы информатики, породив при этом в исследователях желание экспериментировать и находить все новые интересные и востребованные приложения.

Благодарности

Работа выполнена в рамках проектно-учебной группы НИУ ВШЭ «Алгоритмы интеллектуального анализа данных (Data Mining) для Интернет-форумов обсуждения инновационных проектов».

Список источников

1. Игнатов Д.И., Кузнецов С.О. Методы разработки данных (Data Mining) для рекомендательной системы Интернет-рекламы // Одиннадцатая национальная конференция по искусственному интеллекту с международным участием (КИИ-2008, 28 сентября – 3 октября 2008 г., г. Дубна, Россия): Труды конференции. Т.2. – М.: Ленанд, 2008. – 392 с.

2. Игнатов Д.И., Кононыхина О.Н. Решетки формальных понятий для анализа данных социологических опросов// Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник научных трудов V-й Международной научно-технической конференции (Коломна, 20-30 мая 2009 г.). В 2-х томах. Т1. – М.: Физматлит, 2009. – 546 с.

3. Игнатов Д.И., Хавенсон Т.Е. Изучение ресурсной обеспеченности российских школ с помощью методов, основанных на решетках понятий// Социологические методы в современной исследовательской практике: Сборник статей, посвященный памяти первого декана факультета социологии НИУ ВШЭ А.О. Крыштановского / Отв. ред. и вступит. ст. О.А. Оберемко; НИУ ВШЭ, ИС РАН, РОС. М.: НИУ ВШЭ, 2011.

4. Игнатов Д.И., Магизов Р.А. Анализ тримодальных данных на примере Интернет-сервисов социальных закладок// Социологические методы в современной исследовательской практике: Сборник статей, посвященный памяти первого декана факультета социологии НИУ ВШЭ А.О. Крыштановского / Отв. ред. и вступит. ст. О.А. Оберемко; НИУ ВШЭ, ИС РАН, РОС. М.: НИУ ВШЭ, 2011.

5. Игнатов Д. И., Кузнецов С. О., Пульманс Й. Разработка данных систем совместного пользования ресурсами: от трипонятий к трикластерам //Математические методы распознавания образов: 15-я Всероссийская конференция. г. Петрозаводск, 11–17 сентября 2011 г.: Сборник докладов. — М.: МАКС Пресс, 2011. — 618 с. (ISBN 978-5-317-03787-1)

6. Игнатов Д.И., Кузнецов С.О. Бикластеризация объектно-признаковых данных на основе решеток замкнутых множеств// Труды 12-й национальной конференции по искусственному интеллекту, М., Физматлит, Т. 1., С.175-182, 2010.

7. Игнатов Д.И., Каминская С.Ю., Магизов Р.А. Метод скользящего контроля для оценки качества рекомендательных Интернет-сервисов// Труды 12-й национальной конференции по искусственному интеллекту, М., Физматлит, Т. 1., С.183-191, 2010.

8. Игнатов Д.И., Каминская А.Ю, Кузнецов С.О., Магизов Р. А. Метод бикластеризации на основе объектных и признаковых замыканий// Интеллектуализация обработки информации: 8-я международная конференция. Республика Кипр, г. Пафос, 17-24 октября 2010 г.: Сборник докладов. – М.: МАКС Пресс, 2010. – С. 140 – 143.

9. Игнатов Д.И., Кузнецов С.О. О поиске сходства Интернет-документов с помощью частых замкнутых множеств признаков // Труды 10-й национальной конференции по искусственному интеллекту с международным участием (КИИ'06). – М.:Физматлит, 2006, Т.2, стр.249-258

10. Кузнецов С.О., Игнатов Д.И., Объедков С.А., Самохин М.В. Порождение кластеров документов дубликатов: подход, основанный на поиске частых замкнутых множеств признаков. Интернет-математика 2005. Автоматическая обработка веб-данных. Москва: «Яndex», 2005, стр. 302 – 319

11. С.О. Кузнецов, ДСМ-метод как система автоматического обучения, Итоги науки и техники. Сер. Информатика. 1991, Т. 15, С.17-54.

12. С.О. Кузнецов, Формальный анализ понятий с помощью ДСМ-метода, 6-я Национальная Конференция по Искусственному Интеллекту (КИИ-98), т.2, Пущино, АИИ, 1998,С. 591-592.

13. F. Baader and B. Sertkaya. Applying formal concept analysis to description logics. In P. Eklund, editor, Proceedings of the 2nd International Conference on Formal Concept Analysis (ICFCA 2004), volume 2961 of Lecture Notes in Computer Science, pages 261-286. Springer-Verlag, 2004.

14. Belohlavek, Radim and De Baets, Bernard and Outrata, Jan and Vychodil, Vilem. Inducing decision trees via concept lattices. J. International Journal of General Systems, 2009, Volume 38, 4, Pages 455–467(2011)

15. Carpineto, C., Romano, G. (2004a) Concept data analysis: Theory and applications. John Wiley & Sons.

16. Carpineto, C., Romano, G. (2004b) Exploiting the Potential of Concept Lattices for Information Retrieval with CREDO. J. of Universal Computing, 10, 8, 985-1013.

17. Philipp Cimiano, Andreas Hotho, Gerd Stumme, and Julien Tane. Conceptual Knowledge Processing with Formal Concept Analysis and Ontologies. Proceedings of the The Second International Conference on Formal Concept Analysis ICFCA 04, (2961) Springer, 2004.

18. Cimiano, P.; Hotho, A. & Staab, S. Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis. Journal of Artificial Intelligence Research, 2005, 24, 305-339

19. Dau, F., Ducrou, J., Eklund, P. (2008) Concept Similarity and Related Categories in SearchSleuth. P. Eklund et al. (Eds.): ICCS. LNAI 5113, 255-268. Springer.

20. Vincent Duquenne: Latticial Structures in Data Analysis. Theor. Comput. Sci. 217(2): 407-436 (1999)

21. L. Freeman Cliques, Galois Lattices, and the Structure of Human Social Groups. *Social Networks*, 18, 1996, 173-187

22. B. Ganter and S.O. Kuznetsov, Hypotheses and Version Spaces, Proc. 10th Int. Conf. on Conceptual Structures, ICCS'03, A. de Moor, W. Lex, and B.Ganter, Eds., Lecture Notes in Artificial Intelligence, vol. 2746 (2003), pp. 83-95.

23. B. Ganter and R. Wille, Formal Concept Analysis: Mathematical Foundations, Springer, 1999.

24. Bernhard Ganter, Sergei O. Kuznetsov, Scale Coarsening as Feature Selection, In: R.Medina, S.Obiedkov, Eds., Proc. International Conference on Formal Concept Analysis, Lecture Notes in Artificial Intelligence, vol. 4933, pp. 217-228.

25. Bernhard Ganter, Gerd Stumme, Rudolf Wille: Formal Concept Analysis, Foundations and Applications Springer 2005

26. R. Godin, P. Valtchev. Formal Concept Analysis-Based Class Hierarchy Design in Object-Oriented Software Development. In.: B. Ganter, G. Stumme, and R. Wille. (Eds.) Formal Concept Analysis, Foundations and Applications, LNAIvolume 3626, pages 209–231. Springer Berlin / Heidelberg, 2005.

27. W. Hesse, T. Tilley. Formal Concept Analysis Used for Software Analysis and Modelling. In.: B. Ganter, G. Stumme, and R. Wille. (Eds.) Formal

Concept Analysis, Foundations and Applications, LNAI volume 3626, pages 259–282. Springer Berlin / Heidelberg, 2005.

28. D.I. Ignatov, S.O. Kuznetsov. Concept-based Recommendations for Internet Advertisement// In proceedings of The Sixth International Conference Concept Lattices and Their Applications (CLA'08), Radim Belohlavek, Sergei O. Kuznetsov (Eds.): CLA 2008, pp. 157–166 ISBN 978–80–244–2111–7, Palacky University, Olomouc, 2008.

29. D.I. Ignatov, S.O. Kuznetsov. Frequent Itemset Mining for Clustering Near Duplicate Web Documents// In proceedings of The 17th International Conference on Conceptual Structures, S. Rudolph, F. Dau, and S.O.Kuznetsov (Eds.): ICCS 2009, LNCS (LNAI) 5662, pp. 185–200, Springer-Verlag Berlin Heidelberg, 2009

30. Dmitry I. Ignatov, Sergei O. Kuznetsov, Ruslan A. Magizov and Leonid E. Zhukov. From Triconcepts to Triclusters// In proceedings of 13th International Conference on ROUGH SETS, FUZZY SETS, DATA MINING AND GRANULAR COMPUTING, Kuznetsov et al. (Eds.): RSFDGrC 2011, LNCS/LNAI Volume 6743/2011, Springer-Verlag Berlin Heidelberg, 257-264, 2011

31. Dmitry Ignatov and Serafima Mamedova and Nikita Romashkin and Ivan Shamshurin. What can closed sets of students and their marks say?// In proceedings of 4th International Conference on Educational Data Mining, Mykola Pechenizkiy et al. (Eds.), EDM-2011, TU/e Eindhoven, 223-228, 2011

32. Koester, B. (2006) Conceptual Knowledge Retrieval with FooCA: Improving Web Search Engine Results with Contexts and Concept Hierarchies. P. Perner (Ed.): ICDM, LNAI 4065, 176-190. Springer.

33. Sergei O. Kuznetsov, Dmitrii I. Ignatov, Concept Stability for Constructing Taxonomies of Web-site Users// Proc. Satellite Workshop «Social Network Analysis and Conceptual Structures: Exploring Opportunities» at ICFCA'07, Clermont-Ferrand, France, P. 19-24.

34. S.O. Kuznetsov and S.A. Obiedkov, Comparing Performance of Algorithms for Generating Concept Lattices, Journal of Experimental and Theoretical Artificial Intelligence, vol. 14 (2002), pp. 189-216.

35. S.O. Kuznetsov, Galois Connections in Data Analysis: Contributions from the Soviet Era and Modern Russian Research, in Formal Concept Analysis: Foundations and Applications, B. Ganter, G. Stumme, R. Wille, Eds., Lecture Notes in Artificial Intelligence, State-of-the Art Ser. (2005), vol. 3626, pp. 196-225.

36. Lakhal, L., Stumme, G. (2005) Efficient Mining of Association Rules Based on Formal Concept Analysis. B. Ganter et al. (Eds.): Formal Concept Analysis, LNAI 3626, 180-195. Springer

37. Michael Luxemburger. Partielle Implikationen und partielle Abhängigkeiten zwischen Merkmalen. Diplomarbeit, TH Darmstadt, 1988.

38. Loïc Cerf, Jérémy Besson, Céline Robardet, Jean-François Boulicaut: Data Peeler: Contraint-Based Closed Pattern Mining in n-ary Relations. SDM 2008: 37-48

39. Mohr, J., Duquenne, V.: The duality of culture and practice: Poverty relief in New-York City, 1888-1917. Theory and Society 26, 305–356 (1997)

40. Mohr, J., Bourgeois, M., Duquenne, V.: The Logic of Opportunity: A Formal Analysis of the University of California's Outreach and Diversity Discourse. Center for Studies in Higher Education, UC Berkeley, Research and Occasional Papers Series (2004)

41. Jonas Poelmans, Paul Elzinga, Stijn Viaene, Guido Dedene: A Case of Using Formal Concept Analysis in Combination with Emergent Self Organizing Maps for Detecting Domestic Violence.ICDM 2009: 247-260

42. Priss, U. (2000) Lattice-based Information Retrieval. Knowledge Organization, 27, 3, 132-142.

43. Richards, D. (2006) Ad-Hoc and Personal Ontologies: A Prototyping Approach to Ontology Engineering. A. Hoffmann et al. (Eds.): PKAW, LNAI 4303, 13-24. Springer.

44. Nikita Romashkin, Dmitry Ignatov and Elena Kolotova. How university entrants are choosing their department? Mining of university admission process with FCA taxonomies// In proceedings of 4th International Conference on Educational Data Mining, Mykola Pechenizkiy et al. (Eds.), EDM-2011, TU/e Eindhoven, 229-234, 2011

45. Roth, C., Obiedkov, S., Kourie, D. (2008a) Towards Concise Representation for Taxonomies of Epistemic Communities. S.B. Yahia et al. (Eds.): CLA 2006, LNAI 4923, 240-255. Springer.

46. G. Snelting. Concept Lattices in Software Analysis. In.: B. Ganter, G. Stumme, and R. Wille. (Eds.) Formal Concept Analysis, Foundations and Applications, LNAIvolume 3626, pages 151–167. Springer, 2005.

47. Stumme, G., Taouil, R., Bastide, Y., Pasqier, N.and Lakhal, L. Computing Iceberg Concept Lattices with Titanic. J. on Knowledge and Data Engineering, (42)2:189–222, 2002

48. Tilley, T., Eklund, P. (2007) Citation analysis using Formal Concept Analysis: A case study in software engineering. 18th int. conf. on database and expert systems applications (DEXA).

49. T. Tilley, R. Cole, P. Becker, P. Eklund A Survey of Formal Concept Analysis Support for Software Engineering Activities. In.: B. Ganter, G. Stumme, and R. Wille. (Eds.) Formal Concept Analysis, Foundations and Applications, LNAI volume 3626, pages 250–271. Springer, 2005.

50. Wille R. Restructuring Lattice Theory: an Approach Based on Hierarchies of Concepts // Ordered Sets / Ed. by I. Rival. — Dordrecht; Boston: Reidel, 1982.— P. 445–470.

Система анализа данных коллаборативных платформ CrowDM

Д. И. Игнатов[1], А. Ю. Каминская[2], А. А. Беззубцева[3], К. Н. Блинкин[4]

[1]dignatov@hse.ru, [2]skam90@gmail.com, [3]nstbezz@gmail.com
[4]xkonstantinx@gmail.com

НИУ ВШЭ, Россия, 101000, г. Москва, ул. Мясницкая, д. 20

Аннотация. В работе описывается система анализа данных коллаборативной платформы компании Witology. Проект находится в состоянии разработки, поэтому в статье отражены в основном методологические аспекты и результаты первых экспериментов. В основу системы положен ряд моделей и методов современного анализа объектно-признаковых и неструктурированных данных (текстов), таких как Анализ Формальных Понятий, мультимодальная кластеризация, поиск ассоциативных правил и извлечение ключевых словосочетаний и слов из текстов.

Ключевые слова: коллаборативные и краудсорсинговые платформы, разработка данных (Data Mining), анализ формальных понятий, мультимодальная кластеризация.

Введение

Успехи современной индустрии коллаборативных технологий ознаменовались появлением ряда новых платформ для проведения распределенных мозговых штурмов или осуществления так называемой общественной экспертизы, например, на Российском рынке такие продукты выпускают компании Witology [1] и Wikivote [2]. И, хотя до технологического прорыва еще далеко, несколько крупных проектов уже успешно завершены. Среди них «Сбербанк-21», анализ форумов Агент-

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16-18 марта, 2012.

ства Стратегических Инициатив и др. Массивы данных нового типа систем, ядро которых составляют так называемые социосемантические сети, требуют новых подходов к анализу данных. В рамках данной статьи мы предлагаем новую методологическую базу для анализа данных коллаборативных систем, опирающуюся на современные модели и методы разработки данных (Data Mining) и искусственного интеллекта.

Как правило, в рамках одного проекта пользователи таких краудсорсинговых платформ [3] решают некую общую задачу, выдвигают идеи, оценивают идеи друг друга как эксперты, а в итоге по результатам обсуждений и рейтингования определяются лучшие идеи и люди – генераторы идей. Для более глубокого понимания поведения пользователей, выработки адекватных критериев оценки, анализа динамики и статистики в ходе развития проекта необходимы особые средства. Традиционные методы кластеризации, поиска сообществ и анализа текстов нуждаются в адаптации, а иногда и в полной переработке, требуют изобретательности для их результативного применения, т.е. получения действительно полезных и нетривиальных результатов. Мы кратко описываем модели данных, используемых в проекте, в терминах Анализа Формальных Понятий (АФП) [4]. Также мы приводим описание системы анализа данных CrowDM (Crowd Data Mining), ее архитектуру и методы, лежащие в основе ключевых этапов анализа данных.

Математические модели и методы

На начальном этапе анализа данных коллаборативной платформы были выявлены два типа данных такой платформы, напрямую соответствующие двум составляющим социосемантической сети: данные без использования ключевых слов (связи, оценки, действия пользователей) и данные с ключевыми словами (наполнение всего создаваемого контента на платформе).

Для анализа данных без ключевых слов предлагается применять методы анализа социальных сетей (Social Network Analysis), кластеризации (а также би- и трикластеризации [5, 6, 7, 8], спектральной кластеризации), анализ формальных понятий (решетки понятий, импликации, ассоциативные правила) и его расширения для случая мультимодальных данных, например, триадических [9]; рекомендательные системы [10, 11, 12] и статистические методы анализа (анализ распределений и средних значений).

Для методов анализа текстовых данных с использованием ключевых слов, основным является этап выделения ключевых слов и словосочетаний. Это направление компьютерной лингвистики заслуживает отдельного рассмотрения, поэтому в данной статье мы остановимся на некоторых методах анализа данных без использования ключевых слов.

На схеме анализа (см. рис. 2) синим цветом выделены методы, описанные в данной статье.

Главными действующими лицами в краудсорсинговых проектах, а значит и в коллаборативных платформах, созданных для этих проектов, являются пользователи платформы, они же участники проекта. Будем рассматривать их в качестве *объектов* для анализа. Вместе с тем, каждый объект может обладать (или не обладать) определенным набором *признаков*. В качестве признаков пользователей коллаборативной платформы могут выступать темы, в обсуждении которых пользователь принимал участие, идеи, которые он выдвигал или за которые голосовал, и даже другие пользователи. Основным инструментом для анализа данных объектно-признаковой природы является анализ формальных понятий (АФП). Дадим формальные определения.

Контекстом в АФП называют тройку $K = (G, M, I)$, где G — *множество объектов*, M — *множество признаков*, а отношение $I \subseteq G \times M$ говорит о том, какие объекты какими признаками обладают. Для произвольных $A \subseteq G$ и $B \subseteq M$ определены *операторы Галуа*:

$$A' = \{m \in M \mid \forall g \in A\ (g\ I\ m)\};$$
$$B' = \{g \in G \mid \forall m \in B\ (g\ I\ m)\}.$$

Оператор '' (двукратное применение оператора ') является *оператором замыкания*: он идемпотентен ($A'''' = A''$), монотонен ($A \subseteq B$ влечет $A'' \subseteq B''$) и экстенсивен ($A \subseteq A''$). Множество объектов $A \subseteq G$, такое, что $A'' = A$, называется *замкнутым*. Аналогично для замкнутых множеств признаков — подмножеств множества M. Пара множеств (A, B), таких, что $A \subseteq G$, $B \subseteq M$, $A' = B$ и $B' = A$, называется *формальным понятием* контекста K. Множества A и B замкнуты и называются *объемом* и *содержанием* формального понятия (A, B) соответственно. Для множества объектов A множество их общих признаков A' служит описанием сходства объектов из множества A, а замкнутое множество A'' является кластером сходных объектов (с множеством общих признаков A'). Отношение «быть более общим понятием» задается следующим образом: $(A, B) \geq (C, D)$ тогда и только тогда, когда $A \supseteq C$. Понятия формального контекста $K = (G, M, I)$, упорядоченные по вложению объемов образуют решетку $\underline{\mathfrak{B}}(G, M, I)$, называемую *решеткой понятий*. Для визуализации решеток понятий используют т.н. диаграммы Хассе, т.е. граф покрытия отношения «быть более общим понятием».

Так как в худшем случае (булева решетка понятий) количество понятий равно $2^{\min\{|G|,|M|\}}$, то для больших формальных контекстов разумно применять АФП, если данные разрежены. Так же можно использовать различные способы сокращения количества формальных понятий, такие как отбор понятий по индексу устойчивости или размеру объема. Аль-

тернативным подходом является ослабление определения формального понятия, как максимального прямоугольника в объектно-признаковой матрице все элементы которого принадлежат отношению инцидентности. Одним из таких ослаблений является определение объектно-признакового бикластера [2,3].

Если $(g, m) \in I$, то (m', g') называется *объектно-признаковым бикластером* с плотностью $\rho(m',g')=|I \cap (m' \times g')|/(|m'| \cdot |g'|)$.

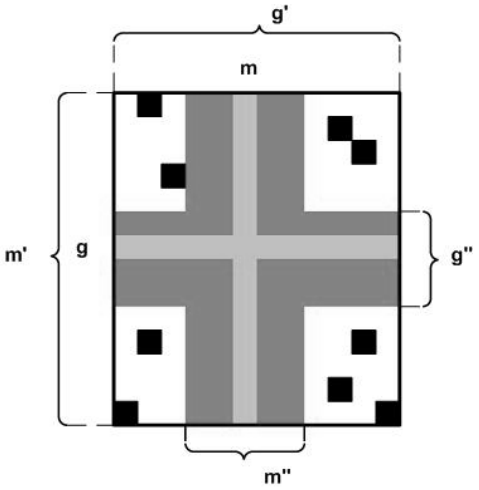


Рис. 1. оп-бикластер

Приведем основные свойства оп-бикластеров:

1. для любого бикластера $(A,B) \in 2^G \times 2^M$ выполняется $0 \le \rho(A,B) \le 1$.

2. оп-бикластер (m',g') является формальным понятием тогда и только тогда, когда $\rho=1$.

3. Если (m', g') – бикластер, то $(g'', g') \le (m', m'')$.

Пусть $(A,B) \in 2^G \times 2^M$ будет бикластером и ρ_{min} неотрицательное действительное число такое, что $0 \le \rho_{min} \le 1$, тогда (A, B) называется *плотным*, если он удовлетворяет ограничению $\rho(A,B) \ge \rho_{min}$.

Из вышеописанного следует, что оп-бикластеры отличаются от формальных понятий тем, что в них не обязательно наблюдается единичная плотность. Графически это означает, что не обязательно все «ячейки» на пересечении объектов и признаков бикластера должны быть заполнены (см. рис. 1).

Помимо построения решеток понятий и их визуализации с помощью диаграмм Хассе используются импликации и ассоциативные правила для выявления признаковых зависимостей в данных. Далее на ос-

нове полученных результатов, можно формировать рекомендации, например, предлагать пользователям наиболее интересные для них обсуждения. Кроме того, можно произвести структурный анализ сети и применить методы кластеризации для поиска сообществ, а также статистические методы для частотного анализа различной активности пользователей.

Почти все вышеперечисленные методы можно применять и к данным с использованием ключевых слов, отличие состоит лишь в том, что в качестве признаков будут выступать ключевые слова, например, употребляемые конкретным пользователем или группой пользователей.

Схема анализа

Схема анализа данных системы CrowDM, создаваемой в данный момент проектно-учебной группой НИУ ВШЭ, представлена на рисунке 2. Ранее упоминалось, что после выгрузки данных из базы, мы получаем формальные контексты и коллекции текстов. Последние в свою очередь тоже преобразуются в формальные контексты после выделения ключевых слов. Далее анализируются полученные контексты.

Результаты экспериментов

Для проведения первых двух экспериментов были отобраны формальные контексты, в которых в качестве объектов выступают пользователи платформы, а в качестве признаков – идеи, которые они предлагали в рамках одной из пяти тем проекта («Сбербанк и частный клиент»). Из всех идей были также отобраны лишь те, которые дошли почти до самого конца проекта. Считается, что объект «пользователь» обладает признаком «идея», если данный пользователь внес любой вклад в обсуждение идеи: является автором идеи, комментировал идею, оставил комментарий в ветке этой идеи, выставил оценку этой идее или комментариям к ней. Таким образом, найденные формальные понятия вида (U,I), где U – множество пользователей, I – множество идей, соответствуют так называемым эпистемическим сообществам (проще говоря, сообществам по интересам) из множества людей U, которые интересуются множествами идей I.

На рисунке 3 представлена диаграмма полученной решетки понятий.

Каждому узлу диаграммы решетки соответствует одно формальное понятие (в данной решетке всего 198 понятий). Также каждый узел помечен множеством объектов и признаков, если этот узел является первым, где встречается данный объект (при движении снизу вверх по диаграмме) или признак (при движении сверху вниз) соответственно. Оче-

видно, что полученная диаграмма решетки является достаточно громоздкой для анализа по ее статическому изображению. Обычно в таких случаях для визуализации используют порядковые фильтры (верхняя часть решетки) или диаграммы множества устойчивых понятий. Мы в свою очередь демонстрируем отдельный фрагмент решетки (см. рис. 4), таким образом, объясняя способ ее «чтения».

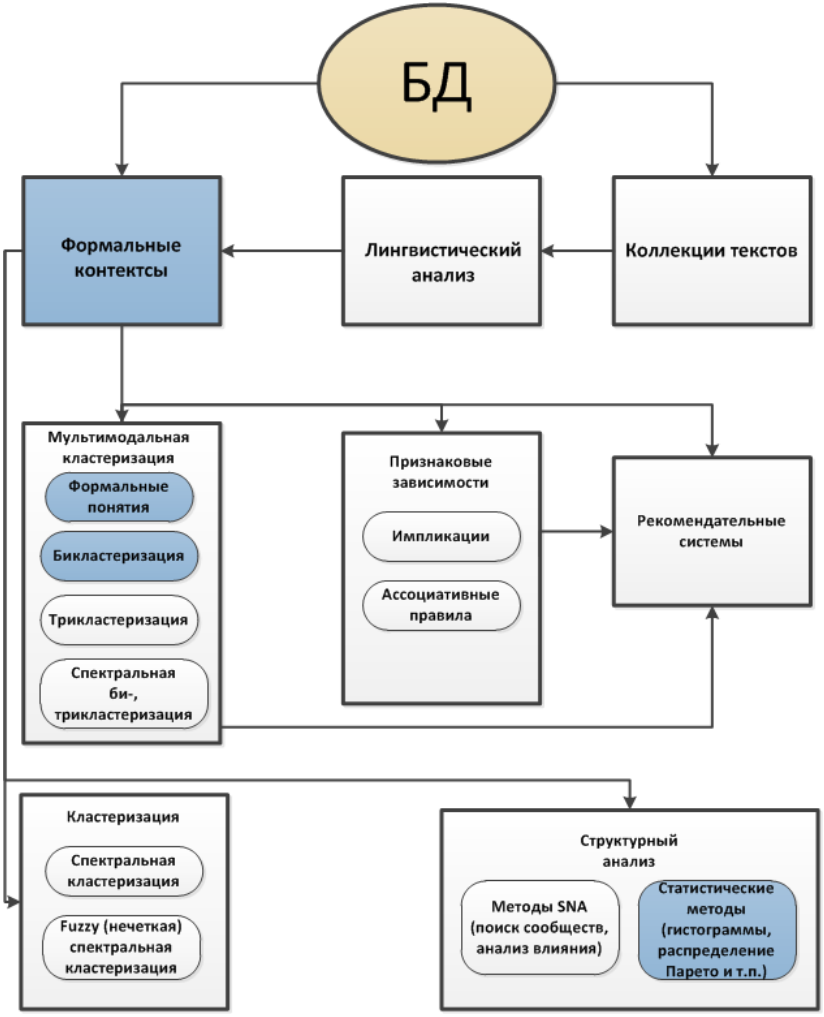


Рис. 2. Схема анализа данных коллаборативных платформ в системе CrowDM

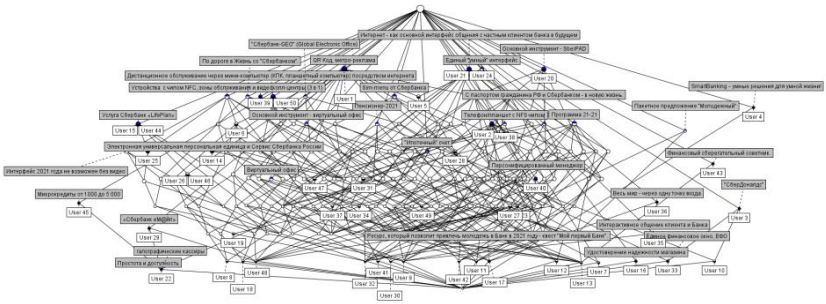


Рис. 3. Диаграмма решетки формальных понятий для контекста пользователи-идеи.

Эксперименты были проведены в программе Concept Explorer, разработанной специально для применения алгоритмов АФП к объектно-признаковым данным. Выделив любой узел решетки, можно увидеть объекты и признаки, соответствующие понятию в этом узле. Объекты «накапливаются» снизу (в данном примере множество объектов состоит из User45 и User22), признаки – сверху (у нас один признак – «Микрокредиты от 1000 до 5000»). Это означает, что пользователи User45 и User22 вместе участвовали в обсуждении идеи с указанным именем и больше ни один из пользователей участия в обсуждении не принимал.

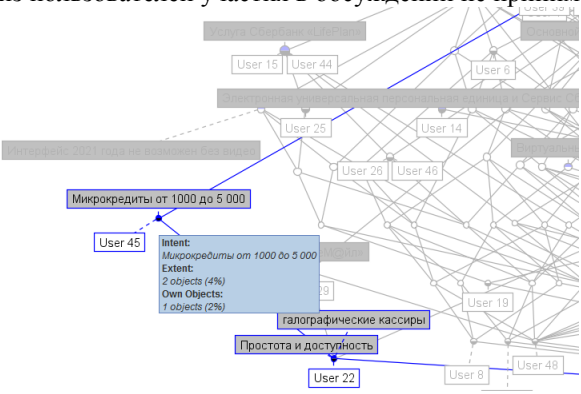


Рис. 4. Фрагмент диаграммы решетки понятий

Ниже представлены результаты применения алгоритмов бикластеризации на тех же самых данных.

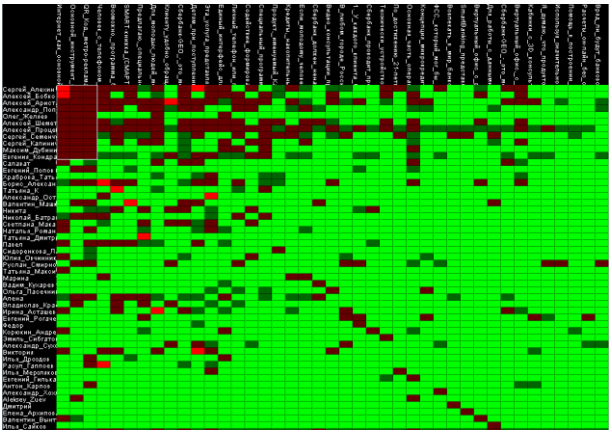


Рис. 5. Результат работы алгоритма бикластеризации BiMax

Поясним рисунок 5. Эксперименты проведены в системе анализа данных генной экспрессии BicAT. Строки соответствуют пользователям, столбцы – идеям в рамках указанной темы, в обсуждении которых пользователи принимали участие. Цвет ячейки на пересечении соответствующей строки и столбца соответствует интенсивности вклада конкретного пользователя в данную проблему. Под вкладом пользователя понимается взвешенная сумма числа его комментариев к этой идее, количества оценок, при этом учитывается, является ли данный человек автором этой идеи, или нет. Самые светлые ячейки соответствуют нулевому вкладу, самые яркие (см. левую верхнюю ячейку на рис.6) – максимальному вкладу. После дискретизации данных (0 соответствовал нулевому вкладу, 1 – ненулевому) к ним был применен алгоритм бикластеризации BiMax, который нашел несколько бикластеров (см. пример на рисунке 6). Поскольку одной из задач проведения краудсорсинговых проектов является поиск людей со схожими идеями, представленный бикластер из 11 пользователей наиболее интересен, в то время как остальные найденные бискластеры содержали в среднем по 4-5 пользователей (с ограничением на количество идей в бикластере строго больше двух).

Далее, чтобы более полно увидеть картину оценивания в проекте, было построено несколько видов графиков распределения оценок. Одним из примеров является график на рисунке 6, который отображает кумулятивное число пользователей, выставивших больше определенного количества оценок за весь проект.

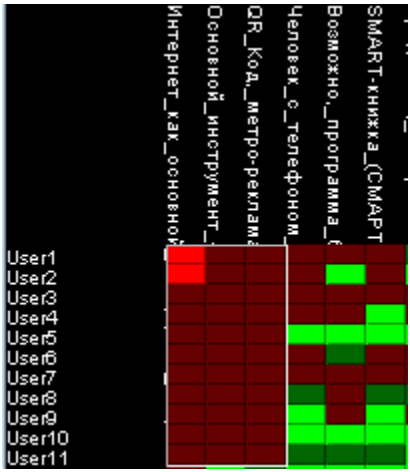


Рис. 6. Бикластер с большим числом пользователей

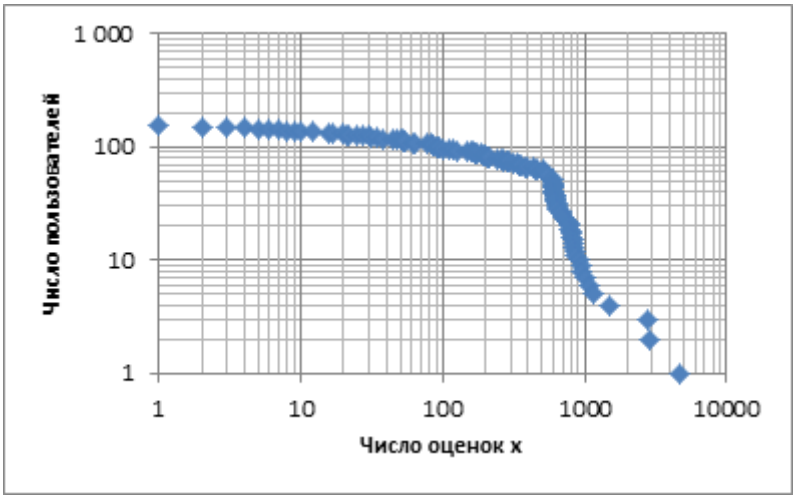


Рис. 7. Распределение числа оценок

По оси абсцисс отложено количество оценок, оставленных пользователем. По оси ординат – число пользователей, которые выставили больше соответствующего числа оценок. Например, больше 5000 оценок поставил один пользователь (крайняя правая точка на оси абсцисс), а больше 4000 – уже упомянутый пользователь и еще один участник. Всего участников, поставивших хотя бы одну оценку, 167. Множество

точек явно разделяется на две части: пологая длинная линия (от x=0 до 544 включительно) и более крутой хвост. Тот факт, что в логарифмических шкалах обе части выглядят похожими на прямые, указывает на то, что обе части, возможно, распределены по Парето.

Целесообразно искать отдельные функции распределения для основной и хвостовой части выборки, потому как если проверить всю выборку на соответствие, например, Парето-распределению, нулевая гипотеза о соответствии отвергается на близком к нулю уровне значимости.

Заключение

Результаты первых экспериментов позволяют утверждать, что разрабатываемая методология окажется полезной для анализа данных коллаборативных систем и систем совместного пользования ресурсами.

Среди направлений дальнейшей работы наиболее приоритетными являются использование текстовой информации генерируемой пользователем и применение методов мультимодальной кластеризации, а также создание рекомендательных сервисов на их основе.

Благодарности

Работа выполнена в рамках проектно-учебной группы НИУ ВШЭ «Алгоритмы интеллектуального анализа данных (Data Mining) для Интернет-форумов обсуждения инновационных проектов».

Список источников

1. http://witology.com/

2. http://www.wikivote.ru/

3. Jeff Howe. The Rise of Crowdsourcing. Wired, 2006.

4. Ganter, B., Wille, R. Formal Concept Analysis. Springer, Heidelberg, 1999.

5. Игнатов Д.И., Кузнецов С.О. Бикластеризация объектно-признаковых данных на основе решеток замкнутых множеств// Труды 12-й национальной конференции по искусственному интеллекту, М., Физматлит, Т. 1., С.175-182, 2010.

6. Игнатов Д.И., Каминская А.Ю., Кузнецов С.О., Магизов Р. А. Метод бикластеризации на основе объектных и признаковых замыканий// Интеллектуализация обработки информации: 8-я международная конференция. Республика Кипр, г. Пафос, 17-24 октября 2010 г.: Сборник докладов.– М.: МАКС Пресс, 2010. – С. 140 – 143.

7. Игнатов Д.И., Магизов Р.А. Анализ тримодальных данных на примере Интернет-сервисов социальных закладок// Социологические методы в современной исследовательской практике: Сборник статей, посвященный памяти первого декана факультета социологии НИУ ВШЭ А.О. Крыштановского / Отв. ред. и вступит. ст. О.А. Оберемко; НИУ ВШЭ, ИС РАН, РОС. М.: НИУ ВШЭ, 2011.

8. Игнатов Д. И., Кузнецов С. О., Пульманс Й. Разработка данных систем совместного пользования ресурсами: от трипонятий к трикластерам //Математические методы распознавания образов: 15-я Всероссийская конференция. г. Петрозаводск, 11–17 сентября 2011 г.: Сборник докладов. — М.: МАКС Пресс, 2011. — 618 с. (ISBN 978-5-317-03787-1)

9. Robert Jäschke, Andreas Hotho, Christoph Schmitz, Bernhard Ganter, Gerd Stumme: TRIAS - An Algorithm for Mining Iceberg Tri-Lattices. ICDM 2006: 907-911

10. Игнатов Д.И., Кузнецов С.О. Методы разработки данных (Data Mining) для рекомендательной системы Интернет-рекламы // Одиннадцатая национальная конференция по искусственному интеллекту с международным участием (КИИ-2008, 28 сентября – 3 октября 2008 г., г. Дубна, Россия): Труды конференции. Т.2. – М.: Ленанд, 2008. – 392 с.

11. D.I. Ignatov, S.O. Kuznetsov. Concept-based Recommendations for Internet Advertisement// In proceedings of The Sixth International Conference Concept Lattices and Their Applications (CLA'08), Radim Belohlavek, Sergei O. Kuznetsov (Eds.): CLA 2008, pp. 157–166 ISBN 978–80–244–2111–7, Palacky University, Olomouc, 2008.

12. Dmitry I. Ignatov, Sergei O. Kuznetsov, Ruslan A. Magizov and Leonid E. Zhukov. From Triconcepts to Triclusters// In proceedings of 13th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing, Kuznetsov et al. (Eds.): RSFDGrC 2011, LNCS/LNAI Volume 6743/2011, Springer-Verlag Berlin Heidelberg, 257-264, 2011.

Прагматическое введение в Semantic Web и Linked Data

Ю.В. Катков

katkov@wikivote.ru

НИУ ИТМО, ООО «ВикиВот!»

Аннотация. Статья представляет собой обзор технологий Semantic Web и Linked Data. Дано краткое введение в технологии и протоколы, рассмотрены некоторые приёмы работы с данными. В помощь разработчику приведены ссылки на литературу и ресурсы сообщества Semantic Web.

Ключевые слова: semantic web, linked data, обзор

Введение

За одиннадцать лет своего существования семантический веб (Semantic Web, SW) прошел путь от одной амбициозной идеи в голове Тима Бернерса-Ли до целого направления в науке, и, кроме того, разработанные технологии и полученные научными группами результаты давно успели найти свое применение на практике.

Сейчас все больше крупных компаний как проявляют интерес к уже существующим проектам в области семантического веба, так и создают свои собственные. Например, проект семантического поиска Powerset был выкуплен компанией Microsoft и использован при создании Bing - поисковой системы, сравнимой по качеству результатов с Yahoo! и Google. В свою очередь Google купил компанию Metaweb, разрабатывающую базу знаний Freebase и средство очистки данных Gridworks (новое название Google Refine). Компании IBM и Oracle занялись разра-

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16-18 марта, 2012.

боткой RDF-хранилищ и библиотек доступа. Наконец, огромным успехом семантического веба можно считать появление ресурса schema.org, на котором размещается общая схема метаданных, которые учитываются поисковыми машинами Google, Yahoo, Bing и Яндекс.

Огромное количество RDF-данных на практически любую тему находится в открытом доступе уже сегодня и эти данные не только связаны с помощью уникальных идентификаторов, но и часто поддерживают вывод из них новых знаний. RDF-метаданные учитываются крупнейшими поисковыми системами, все больше правительств присоединяются к инициативе Открытых государственных данных, доступных с помощью SPARQL-запросов.

Довольно сложно спорить с тем, что Semantic Web - это актуально.

Целью этой статьи будет изложение основ семантических технологий[1] с прагматических позиций. После краткого введения будет рассмотрено несколько примеров задач, ради решения которых стоит браться за изучение стандартов Semantic Web. Затем будет произведен краткий обзор стандартов, используемых сегодня для представления данных и доступа к ним, а также будут показаны некоторые приёмы, использующиеся при изучении источников RDF-данных. Наконец, будут названы основные инструменты исследователя и программиста и вкратце затронут вопрос публикаций собственных данных в общепринятых форматах. Статья завершается кратким обзором литературы, электронных ресурсов и событий, посвященных Linked Data.

Общие сведения

«Семантический веб - это веб данных» - объясняет комитет по стандартизации W3C на официальной странице проекта Semantic Web.[2] Термин был введен Тимом Бернерсом-Ли, создателем World Wide Web [1] и основателем консорциума W3C. Свое видение Тим и его коллеги изложили в публицистической статье The Semantic Web [2]. Центральным элементом проекта являются действующие во всемирной паутине автоматические агенты, оперирующие со структурированными данными. Эти агенты могут выполнять интеллектуальные поисковые запросы, добывать новые знания из уже имеющихся, и таким образом помогать людям принимать важные решения.

В консорциуме W3C начали разрабатываться стандарты для обеспечения жизненного цикла данных во всемирной паутине. Тимом Бер-

[1] Здесь и далее под семантическими технологиями и форматами понимаются те из них, которые имеют отношение к Semantic Web.
[2] http://www.w3.org/2001/sw/

нерсом-Ли была предложена высокоуровневая архитектура, получившая название слоеного пирога семантического веба:[1]

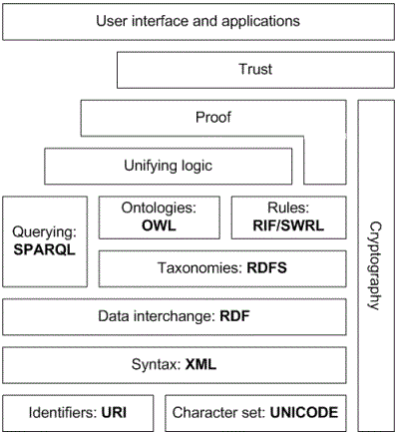


Рисунок 2. Стек семантического веба

1. Все сущности в вебе должны соответствовать т.н. ресурсам, а те, в свою очередь, должны уникальным образом идентифицироваться с помощью URI (Uniform Resource Identifier, [3]), частным случаем которыя являются URL.
2. Для обмена данными должен использоваться язык XML [4]. «Точно так же, как HTML был создан, чтобы любой пользователь мог читать Internet-документы, XML дает нам то эсперанто, на котором любой может читать и писать, невзирая на вавилон несовместимых платформ» [5].
3. Для представления данных используется графовый язык Resourse Description Framework [6], [7], где данные описываются тройками идентификаторов и XML-литералов в последовательности «субъект»-»предикат»-»объект». Мощь подобного представления данных в том, что такие графы легко объединять между собой - для объединения данных из двух RDF-файлов достаточно дописать один файл в конец другого. RDF имеет несколько способов записи (сериализации) в виде троек «субъект - предикат - объект» (Notation3 [8], Turtle [9], N-triples) и в виде XML [10].
4. Простые схемы данных описываются в терминах RDF Schema [11] – аналога XML Schema для RDF. С течением времени схемы могут эволюционировать, при этом не нужно обновлять

[1] из презентации http://www.w3.org/2007/Talks/0130-sb-W3CTechSemWeb

RDF-данные, связанные с этими схемами. RDF Schema позволяет оперировать таксономическими связями, а также задавать области определения и значения предикатов.

5. К данным должны осуществляться запросы - для этого существует язык запросов и протокол, специфицируемые стандартом SPARQL[12][13].
6. Сложные схемы данных описываются языком OWL, по ним доступен логический вывод новых фактов. OWL прочно стоит на логическом фундаменте (его подмножество OWL-DL основывается на формализме дескрипционных логик [14], [15], [16]) и предоставляет гораздо большую выразительность для описания словарей. Для того, чтобы подчеркнуть тот факт, что RDFS- и OWL-документы являются полноценными схемами данных и на их основе можно производить логический вывод, их называют онтологиями [17].
7. Необходима унификация логики, для того, чтобы на факты, выраженные в терминах одной логики интерпретировались в других системах.
8. Данные должны содержать информацию об их источнике и интеллектуальные агенты должны иметь достаточно данных для того, чтобы в большей или меньшей степени доверять тому или иному источнику данных.

Желающим изучить стек семантического веба подробнее следует обратиться к книгам и обзорам, рекомендуемым в последней главе данной статьи. Из русскоязычных работ рекомендуется [18] и вики Semantic Future[1] Для понимания роли семантического веба в контексте развития всемирной паутины следует обратиться к документам по развитию WWW[2] и книге [19].

Область применения Linked Data

Использование технологий семантического веба позволяет улучшить многие приложения, но наиболее успешно применяются для следующих задач:

- получение актуальных структурированных данных из внешних источников,
- публикация собственных массивов данных во всемирной паутине для использования сторонними организациями,
- повышение релеватности поисковой выдачи,
- улучшение структуры публикуемых данных,

[1] http://semanticfuture.net/
[2] http://www.w3.org/DesignIssues/

- поддержка задач компьютерной лингвистики,
- автоматический сбор статистики и анализ данных, удовлетворяющих определенным критериям.

Приведем несколько примеров разработанных приложений, которые могут прояснить то, как используются данные в семантическом вебе:

- Проекты, похожие на Where does my Money Go[1] помогают британскому налогоплательщику понять, как государство тратит его деньги: приложение использует красочную инфографику для того, чтобы показать связь доходов пользователя с теми государственными проектами, которые сейчас активны: работы в области здравоохранения, охраны окружающей среды, науки и т.д. Приложение использует официальные регулярно обновляемые данные о налогах из data.gov.uk в формате RDF. Благодаря тому, что эти данные связаны с тематической онтологией, многочисленные статьи расходов иерархически структурированы и позволяют легко создавать аналитические приложения.
- компания BBC постепенно внедряет достижения семантического веба на своих вебсайтах. И неудивительно - у корпорации есть не только огромные массивы данных о многих музыкальных и научно-популярных теле- и радиопрограммах, но и собственные исследовательские проекты, посвященные биологическому разнообразию. Веб-приложение BBC Wildlife finder[2] увязывает эти данные воедино и соединяет их с внешними источниками: на странице, посвященной львам можно увидеть не только ссылки на все передачи BBC, посвященные этим величественным кошкам. Они включают в себя также динамически обновляемую информацию из Википедии (проекта Dbpedia [20], [21], [22]) и из международных биологических баз данных, предоставляющих их в виде RDF.
- Кембриджский проект True Knowledge[3] позволяет мгновенно получать ответы на вопросы заданные на естественном языке. А компания IBM недавно потрясла мир еще более впечатляющим проектом: их компьютер IBM Watson победил чемпиона мира по игре Jeopardy (российский аналог - передача «Своя игра»). Оба этих проекта заявляют о том, что используют RDF-данные Dbpedia [23].

[1] http://wheredoesmymoneygo.org
[2] http://www.bbc.co.uk/nature/wildlife
[3] http://www.trueknowledge.com

Работа с открытыми связанными данными

Основным стимулом для изучения стандартов является существование большого количества опубликованных в общем доступе данных, для работы с которыми эти стандарты могут быть использованы.

Для того, чтобы быстро обозреть основные источники данных достаточно посмотреть на известное облако связанных данных (Linked Data Cloud)[1]. Видно, что наибольшее количество данных посвящено научным публикациям, затем следуют источники данных по биологии, открытые государственные данные и медиаинформация. Сравнительно небольшое количество датасетов (англ. *dataset*, источник данных) с географическим данными компенсируется большим количеством входящих и исходящих ссылок на них и довольно высоким качеством.

Несмотря на то, что оригинально создание визуализации облака связанных данных преследовало декоративные цели, оно может использоваться как удобный инструмент для того, чтобы понять, из какого источника предполагается черпать данные.

Самым правильным и удобным способ знакомства с данными является изучение их CKAN-описаний. CKAN Project[2] - это хаб, на котором хранятся описания RDF-хранилищ, относящихся к Linked Data. Именно базы, попавшие в CKAN отображаются в облаке Linked Data. CKAN-страницы датасетов содержат RDFS и OWL схемы, лежащие в основе RDF-баз, их машиночитабельные описания в формате VoID [24], ссылки на их SPARQL-точки (SPARQL endpoint), примеры описанных объектов и некоторую статистику.

Приведем несколько приёмов, использующихся для того, чтобы найти нужные данные в облаке Linked Data. В первую очередь, стоит проверить, есть ли база по интересующей вас теме в CKAN.

Затем можно воспользоваться RDF-поисковиком, например Sig.ma[3], Sindice[4] или Swoogle[5]. После этого данные удобно просматривать с помощью RDF-браузера, наглядно показывающего объекты и их RDF-свойства. Иногда держатель данных предоставляет RDF-браузер на сайте (например, Dbpedia), но это не так, то можно воспользоваться браузерами Marbles[6] или Operator[7].

---

[1] http://richard.cyganiak.de/2007/10/lod
[2] http://thedatahub.org
[3] http://sig.ma
[4] http://sindice.com
[5] http://swoogle.umbc.edu
[6] http://marbles.sourceforge.net
[7] https://addons.mozilla.org/en-US/firefox/addon/operator

Другим эффективным способом исследовать данные являются проверочные SPARQL-запросы. В таблице (Таблица 1) приведены примеры таких запросов.

Таблица 1. Пробные SPARQL-запросы.

Запрос	Значение
SELECT * WHERE {?s ?p ?o} LIMIT 1000	показать тысячу произвольных триплетов
SELECT DISTINCT ?p WHERE {?s ?p ?o} LIMIT 1000	показать не более тысячи свойств
SELECT DISTINCT ?p WHERE { ?s ?p ?o . ?p a rdf:Property. } LIMIT 1000	чуть более узкий запрос - показывает свойства, явно помеченные в онтологии как таковые

SELECT ?p (COUNT ?p as ?countPredicate) WHERE {
?s ?p ?o
}
GROUP BY ?p
ORDER BY DESC (?countPredicate)
LIMIT 100
	вывести сто классов объектов (модифицируется с использованием GROUP BY)

SELECT ?o WHERE { ?s ?p ?o. ?o a rdf:Class } LIMIT 100	вывести сто классов объектов (модифицируется с использованием GROUP BY)

При практическом использовании данных Semantic Web разработчику потребуется ряд инструментов. Перечислим наиболее часто применяющиеся:

- средства конвертации данных в RDF (*RDFizers*), например Google Refine[1] + Rdf Plugin[2]
- RDF-редакторы, например OntoWiki[3], Altova SemanticWorks®[4]
- редакторы онтологий, например Protege[5], NeOn Toolkit[6], TopBraid Suite[7]

[1] http://code.google.com/p/google-refine
[2] http://lab.linkeddata.deri.ie/2010/grefine-rdf-extension
[3] http://ontowiki.net/Projects/OntoWiki
[4] http://www.altova.com/solutions/semantic-web-tools.html
[5] http://protege.stanford.edu
[6] http://neon-toolkit.org
[7] http://www.topquadrant.com/products/TB_Suite.html

- программные библиотеки для доступа к RDF-данным (*RDF libraries and frameworks*), например Jena (Java)[1], Sesame (Java)[2], dotNetRdf (.Net)[3], ARC2 (PHP)[4], Graphite (PHP)[5], rdflib (Python)[6], (Python)[6], Redland (мультиязычная)[7]
- движки логического вывода (*reasoners, inference engines*), например Pellet[8], Fact++[9], Hermit[10]
- RDF-хранилища (*RDF storages, triple storages*), например OpenLink Virtuoso[11], 4Store[12], Sesame[13].

Помимо этих инструментов программиста, существуют также готовые к использованию семантические платформы, например многочисленные семантические вики[14] и CMS (англ. *Content Management Systems*, системы управления содержимым) с поддержкой RDF (модули для Joomla[15] и Drupal[16]).

Подводя итог, можно без преувеличения можно сказать, что сегодня существуют программные средства (зачастую с открытым программным кодом) и интернет-сервисы, использование которых упростит разработку на всех стадиях построения семантического проекта. В [25] и [26] приведён подробный обзор инструментов в контексте жизненного цикла связанных данных и архитектуры приложений соответственно. Существует также несколько списков и постоянно пополняемых каталогов подобного программного обеспечения.[17]

Обзор литературы и сообществ

Исследователи и работники организаций, использующие в своей работе семантические технологии, формируют сообщество, и количест-

[1] http://incubator.apache.org/jena
[2] http://www.openrdf.org
[3] http://www.dotnetrdf.org
[4] https://github.com/semsol/arc2
[5] http://graphite.ecs.soton.ac.uk
[6] http://code.google.com/p/rdflib
[7] http://librdf.org
[8] http://clarkparsia.com/pellet
[9] http://owl.man.ac.uk/factplusplus
[10] http://hermit-reasoner.com
[11] http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main
[12] http://4store.org
[13] http://www.openrdf.org
[14] автором статьи поддерживается страница Семантические вики в Википедии
[15] http://swm.deri.org/jsyndication
[16] http://semantic-drupal.com
[17] http://www.w3.org/2001/sw/wiki/Tools

во участников этого сообщества постоянно растёт. Для того, чтобы помочь начинающему исследователю или разработчику сориентироваться, ниже приведен обзор событий, публикаций и ресурсов, связанных с семантическим вебом.

Книги и статьи

С момента выхода статьи «The Semantic Web» в 2001 году тема семантического веба породила огромное количество исследований, и, как следствие, научных статей и книг. Заметим, однако, что чтение литературы, выпущенной до 2006-го года, вряд ли приблизит разработчика к пониманию современных подходов и инструментов семантического веба. Литература этого периода - это пионерские работы, записи проб и ошибок - её стоит читать только тем, кто глубоко погрузился в проблематику Semantic Web, но она не годится для тех, кто собирается использовать семантические технологии как подспорье для своей прикладной деятельности.

Книга «Semantic Web Programming» [25] служит отличным стартом для практической работы. Хебелер и соавторы отталкиваются от практических задач и дают прекрасный обзор с примерами на Jena. Работа «Programming the Semantic Web» [27] также прекрасно подходит для разработчиков ПО и содержит большое количество простых примеров на языке Python. Она перекликается с другой книгой Тоби Серагана «Программируем коллективный разум» [28] - замечательным введением в машинное обучение, data mining и социальные алгоритмы. Из бесплатных книг, касающихся темы практического использования Linked Data (англ. *consuming Linked Data*) стоит отметить книгу [29]. Кроме этого рекомендуются материалы лекций школ по Linked Data и Semantic Web (обзор таких школ приведён ниже).

Книги [30] и [31] не ориентированы на разработчика, а скорее представляют собой обзоры того, как устроен современный семантический веб, какие инициативы в нем реализуются, в то время как [32] рассказывает о математических основаниях проекта. «Handbook...» также содержит множество идей семантических приложений на основе существующих данных.

Тем, кто занимается публикацией массивов данных в вебе, нужно ознакомиться с бесплатными онлайн-книгами [33] и [34], статьёй [26], а также заметкой Тима Бернерса-Ли [35], позволяющую оценить предоставляемые данные по шкале от одного до пяти.

Списки рассылки и форумы

Новичок в вопросах связанных данных всегда может найти достаточное количество ресурсов, дающих краткое введение в тему, а также имеет возможность задать интересующий его вопрос на одном из фору-

мов или списках рассылки. Главные списки рассылки, Semantic Web[1] и public-lod[2], служат как для общения участников, так и для информирования сообщества о предстоящих мероприятиях. Каждый из стандартов разрабатывает своя рабочая группа, имеющая свои списки рассылки; аналогично поступают университетские лаборатории и команды, занимающимися разработкой ПО. Помимо этого, набирает популярность ресурс Semantic Overflow[3] - форум экспертов, построенный на технологии StackExchange. Из русскоязычных можно выделить рассылки Веб Данных[4] и Open Government[5], рассылку школы KESW[6] а также форум Semantic Future[7].

Конференции и семинары

Научные и технические новинки обсуждаются на тематических конференциях и семинарах. Наиболее престижной считается World Wide Web Conference[8], затем следуют International Semantic Web Conference[9] и Extended Semantic Web Conference[10]. Примечательно то, что на этих событиях часто проводятся так называемые Doctoral Symposium, цель которых - помочь аспирантам определиться с темами их диссертаций. Русскоязычные конференции, в которых поднимается тема связанных данных и семантического веба это KESW[11], RCDL[12], WebConf[13], OSTIS[14], КИИ[15].

Журналы

Зачастую в научных журналах предъявляются более высокие требования к качеству статей, нежели на конференциях. Поэтому публикация в уважаемом журнале высоко ценится, а статьи содержат полную и хорошо описанную информацию о проведённых исследованиях. Среди

[1] http://lists.w3.org/Archives/Public/semantic-web
[2] http://lists.w3.org/Archives/Public/public-lod
[3] http://answers.semanticweb.com
[4] http://groups.google.com/group/webofdata-russian
[5] http://groups.google.com/group/opengovdataru
[6] https://groups.google.com/group/kesw-school
[7] http://forum.semanticfuture.net
[8] http://www2012.wwwconference.org
[9] http://iswc2012.semanticweb.org
[10] http://2012.eswc-conferences.org
[11] http://kesw.ifmo.ru
[12] http://rcdl.ru/
[13] http://www.webconf.bsu.by
[14] http://conf.ostis.net
[15] http://www.isa.ru/cai

журналов по семантическим технологиям стоит выделить SWJ[1], IJSWIS[2], ETAI[3], IJSWIS[4], JWS[5].

Школы

Школы являются прекрасным способом систематизации знаний и изучения нового на практике за короткий промежуток времени. На момент написания статьи на тему семантических технологий регулярно проводится несколько летних школ: Reasoning Web[6], ESWC Summer School[7], SSSW[8], ASWS[9], SSSC[10]. Лекторами на европейских и американских школах зачастую становятся признанные учёные с большим опытом и разработчики известных приложений. Из русскоязычных школ можно выделить KESW[11] и Russir[12]. Помимо самого процесса обучения, на школе есть возможность проконсультироваться по теме дипломной или диссертационной работы с признанными экспертами. Чтение материалов таких школ тоже крайне полезно, так как обычно это качественные обзоры.

Новостные ресурсы

Для того, чтобы быть в курсе последних событий, стоит посещать не только подписаться на рассылки, но и посещать новостные ресурсы (либо подписаться на обновления с них с помощью RSS-агрегаторов). В первую очередь это сайты w3.org, semanticweb.com и semanticweb.org (последний вебсайт является вики-системой и туда стоит добавлять информацию о своих разработках). Раздел, посвященный семантическим технологиям есть на сайте ReadWriteWeb[13]. Сайт AI3[14] также занимается отслеживанием последних новостей. Ресурс Cloud of Data[15] предоставляет новости и интервью в виде аудиоподкастов.

---

[1] http://www.semantic-web-journal.net
[2] http://www.igi-global.com/journal/international-journal-semantic-web-information/
[3] http://www.etaij.org/seweb
[4] http://www.ijswis.org
[5] http://www.websemanticsjournal.org
[6] http://reasoningweb.org
[7] http://summerschool2012.eswc-conferences.org
[8] http://sssw.org
[9] http://asws2011.semsphere.com
[10] http://sssc2011.sti2.org
[11] http://kesw.ifmo.ru
[12] http://romip.ru/edbt-russir2011
[13] http://www.readwriteweb.com/archives/semantic-web
[14] http://www.mkbergman.com
[15] http://cloudofdata.com

Сообщество

Наконец, стоит сказать о некоторых группах в сообществе Semantic Web. Одни исследователи ориентированы на данные и исповедуют подход снизу вверх (bottom-up): сюда входят создатели крупнейших общедоступных баз данных и исследователи, работающие в области ubiquitous computing. Они признают важность использования онтологий для схем данных, но эти онтологии используют довольно мало возможностей OWL (в основном owl:sameAs), могут содержать противоречия (быть несогласованными, англ. *inconsistent*), а для их обработки помимо алгоритмов логического вывода могут применяться и структурные подходы вроде вывода по графам. Другая группа вышла из сообщества логиков и меньше заботится о публикации и связывании данных, ориентируясь на мощный логический вывод: таковы проекты из области биоинформатики. Наконец, существует прослойка, занимающаяся системным анализом и моделированием - с помощью OWL и языков представления правил они создают модели предметных областей и бизнес-процессов. Их онтологии должны быть понятны людям, а потому большое значение придаётся визуализациям (в том числе на UML). И, хотя формально граница между этими фракциями не проводится, крайне важно правильно расставить свои приоритеты при выборе данных и онтологий для повторного использования.

Заключение

Использование семантических технологий перестало быть уделом исследовательских лабораторий - стандарты, форматы, библиотеки и программы для работы со связанными данными успешно используются для создания коммерческих приложений. Постоянно растущее количество RDF-данных, доступных через SPARQL-интерфейсы, открывает перспективы для создания функциональных и мощных приложений в гораздо более короткие сроки, чем это было возможно раньше. В этой статье было дано введение в технологии Linked Data и Semantic Web с точки зрения прикладного программиста. Автор надеется, что предоставленный обзор и ссылки окажут помощь разработчику в написании приложений нового поколения.

Благодарности

Автор хочет поблагодарить Починок Ирину за ценные советы и замечания относительно текста статьи.

Список источников

1. Berners-Lee T. et al. World-Wide Web: The Information Universe // Internet Research. 1992. Vol. 2, № 1. P. 52–58.

2. Berners-Lee T., Hendler J., Lassila O. The Semantic Web // Scientific American / ed. Gómez-Pérez A., Yu Y., Ding Y. Citeseer, 2001. Vol. 284, № 5. P. 34–43.

3. Berners-Lee T., Fielding R., Masinter L. RFC 3986 - Uniform Resource Identifier (URI): Generic Syntax // Technical report httptoolsietforghtmlrfc3986. Network Working Group, 2005. P. 1–62.

4. Bray T., Paoli J., Sperberg-McQueen C. Extensible Markup Language (XML) // W3C recommendation. World Wide Web Consortium, 2000. Vol. 2004, № 31-05. P. 1–7.

5. Bosak J., Bray T. XML and the Second-Generation Web // Scientific American. Scientific American, 1999. Vol. 280, № 5. P. 89–93.

6. Lassila O., Swick R.R. Resource Description Framework (RDF) Model and Syntax Specification // World Wide Web Internet And Web Information Systems / ed. Lassila O., Swick R.R. 1999. Vol. 2004, № October. P. 1–54.

7. Manola F., Miller E. RDF Primer // W3C Recommendation / ed. Manola F., Miller E. W3C, 2004. Vol. 10, № February. P. 1–107.

8. Berners-Lee T. Notation3 (N3) A readable RDF syntax // Design Issues. W3C, 1998.

9. Beckett D., Berners-Lee T. Turtle - Terse RDF Triple Language // W3C Team Submission. Chapman and Hall, 2008. Vol. 28, № January. P. 3–11.

10. Beckett D. RDF/XML Syntax Specification (Revised) // W3C recommendation / ed. Beckett D. World Wide Web Consortium, 2004. Vol. 10. P. 1–37.

11. Brickley D., Guha R.V. RDF Vocabulary Description Language 1.0: RDF Schema // W3C Recommendation / ed. McBride B. W3C, 2004. Vol. 2009, № 10 February 2004.

12. Prud'hommeaux E., Seaborne A. SPARQL Query Language for RDF // W3C Recommendation / ed. Prud'hommeaux E., Seaborne A. W3C, 2008. Vol. 2009, № January. P. 1–106.

13. Harris S., Seaborne A. SPARQL 1.1 Query Language // W3C Working Draft / ed. Harris S., Seaborne A. W3C, 2010. № May.

14. Nardi D., Brachman R.J. An Introduction to Description Logics.

15. Baader F. Basic Description Logics // The Description Logic Handbook / ed. Baader F. et al. Cambridge University Press, 2003. Vol. 25, № 1. P. 43–95.

16. Schulz S., Hahn U. Description Logics // Studies In Health Technology And Informatics / ed. Van Harmelen F., Lifschitz V., Porter B. Springer--Verlag, 2004. Vol. 101, № 07. P. 137–141.

17. Gruber T.R. A translation approach to portable ontology specifications // Knowledge Acquisition. Citeseer, 1993. Vol. 5, № 2. P. 199–220.

18. Андон Ф.И., Гришанова И.Ю., Резниченко В.А. Semantic Web как новая модель информационного пространства Интернет | Щербак Сергей [Online]. 2009. URL: http://shcherbak.net/semantic-web-kak-novaya-model-informacionnogo-prostranstva-internet/ (accessed: 21.02.2012).

19. Berners-Lee T. Weaving the Web // The original design and ultimate destiny of the world wide web by its inventor. Harper Collins, 1999. P. chapter 12.

20. Kobilarov G. et al. DBpedia - A Linked Data Hub and Data Source for Web and Enterprise Applications // Knowledge Creation Diffusion Utilization. 2009. P. 1–3.

21. Bizer C. et al. DBpedia - A crystallization point for the Web of Data // Web Semantics Science Services and Agents on the World Wide Web. Elsevier, 2009. Vol. 7, № 3. P. 154–165.

22. Auer S. et al. Dbpedia: A nucleus for a web of open data // The Semantic Web / ed. Aberer K. et al. Springer, 2007. Vol. 4825, № Springer. P. 722–735.

23. Ferrucci D. et al. Building Watson : An Overview of the DeepQA Project // AI Magazine. Association for the Advancement of Artificial Intelligence, 2010. Vol. 31, № 3. P. 59–79.

24. Alexander K. et al. Describing Linked Datasets with the VoID Vocabulary // Knowledge Creation Diffusion Utilization. 2010. № March.

25. Hebeler J., Fisher M., Blace R. Semantic web programming. 2011.

26. Auer S., Lehmann J., Ngonga Ngomo A.C. Introduction to linked data and its lifecycle on the web // Reasoning Web Semantic Technologies for the Web of Data / ed. Polleres A. et al. Springer, 2011. Vol. 6848. P. 1–75.

27. Segaran T., Evans C., Taylor J. Programming the Semantic Web // Semantic Web Services Processes and Applications / ed. Treseler M. O'Reilly Media, 2009. Vol. 54, № 2. P. 300.

28. Segaran T. Programming Collective Intelligence // Computational intelligence the experts speak. O'Reilly, 2007. P. 368.

29. Heath T., Bizer C. Linked Data: Evolving the Web into a Global Data Space // Synthesis Lectures on the Semantic Web Theory and Technology / ed. Van Harmelen F., Hendler J. Morgan & Claypool, 2011. Vol. 1, № 1. P. 1–136.

30. Pollock J. Semantic Web for dummies // Production. 2009.

31. Year C. et al. Handbook of Semantic Web Technologies // World Wide Web Internet And Web Information Systems / ed. Domingue J., Fensel D., Hendler J.A. Springer Berlin Heidelberg, 2011. P. 157–190.

32. Hitzler P., Krötzsch M., Rudolph S. Foundations of Semantic Web Technologies // Chapman Hall CRC Press. Chapman & Hall/CRC, 2009. P. 427.

33. Bizer C., Cyganiak R., Heath T. How to Publish Linked Data on the Web // Publish. 2007. Vol. 20, № October. P. 43.

34. Dodds L., Davis I. Linked Data Patterns [Online]. 2011. URL: http://patterns.dataincubator.org/book/ (accessed: 21.02.2012).

35. Berners-Lee T. Linked Data // TED. 2009.

Сходимость эмпирических случайных процессов и обобщающая способность алгоритмов обучения

Михаил Юрьевич Хачай

mkhachay@imm.uran.ru

Институт математики и механики УрО РАН, 620219, Россия, г.Екатеринбург, ГСП-384, ул. Софьи Ковалевской, 16

Уральский государственный университет, 620000, Россия, г. Екатеринбург, ул. Тургенева, 4

Аннотация. Классический подход к обоснованию алгоритмов обучения распознаванию и восстановления эмпирических закономерностей более общей природы связан с выводом гарантированных оценок математического ожидания подходящей функции потерь. Построение подобных оценок на основе материала обучения традиционно связано с рассмотрением сходимости подходящих эмпирических случайных процессов. При этом получаемые результаты опираются либо на условия равномерной сходимости (родственные теореме Гливенко-Кантелли), либо слабой сходимости (близкие к ЦПТ и теореме Донскера) исследуемых процессов. В докладе наряду с известными будут приведены новые результаты, касающиеся достаточных условий как первого, так и второго рода для специального класса эмпирических процессов, порождаемых процедурами обучения.

Ключевые слова: машинное обучение, обобщающая способность, оценки сходимости.

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16-18 марта, 2012.

Влияние метрики на эффективность сжатия видеоизображения

Е. А. Альтман, Е. И. Захаренко

ОмГУПС, Омск, Россия

Аннотация. Сжатие видео и повышение качества изображения являются актуальными задачами видео кодирования и средством повышения эффективности анализа видеоизображений. Основным этапом кодирования, влияющим на степень сжатия, является оценка движения. В статье представлен анализ существующих метрик оценки движения и предложен новый метод, обеспечивающий лучшие показатели сжатия при том же качестве изображения.

Ключевые слова: метрика, оценка движения, сжатие видео, алгоритм сопоставления блоков, метод полного перебора, метод бриллиантового поиска, стандарт MPEG-4, квантование, дискретное косинусное преобразование, зигзаг-преобразование.

Введение

Одна из наиболее актуальных задач анализа видеоизображений – это распознавание объектов. Повысить эффективность реализации этой задачи можно посредством улучшения качества видео при неизменном размере видеофайла. Этого можно добиться путем применения алгоритмов эффективного сжатия.

Также актуальным для систем видеонаблюдения, которые осуществляют анализ видеоконтента, является быстрая передача и компактное хранение полученных данных.

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16-18 марта, 2012.

Основная сложность при работе с видео – недостаточная пропускная способности канала связи, до 100 Мбит/с. Тогда как для передачи видео 720×480 пикселей в формате RGB с частотой 25 кадров в секунду необходим канал с пропускной способностью 198 Мбит/с, а для видео HDTV 1280×720 пикселей – 633 Мбит/с. При этом, принимая во внимание значительную избыточность, присущую видеоизображениям, используют алгоритмы сжатия.

Оценка и компенсация движения являются основными этапами сжатия видео во многих телекоммуникационных системах, например, телевидение высокой и стандартной четкости (HDTV и SDTV), видео конференции и мультимедиа сервисы для Web приложений. Международные стандарты, такие как MPEG, ATSC, и ITU в качестве метода оценки движения регламентируют применять алгоритм сопоставления блоков (Block Matching Algorithm – BMA) [1]. Основной частью алгоритма BMA является метод поиска остаточных блоков и векторов смещения или как его принято называть – метрика. По причине того, что в стандарте не регламентируется выбор этого метода, существует несколько метрик, многие из которых не обеспечивают наилучшее сжатие видео в условиях неизменного качества изображения.

Развитие систем телекоммуникаций требует все большего сжатия видео при том же качестве изображении. Эта задача может быть решена путем использования более эффективной метрики оценки движения.

В статье предложен новый критерий, использование которого при той же степени четкости изображения приводит к более эффективному сжатию видео по сравнению с наиболее популярными метриками.

Популярные метрики

Алгоритм BMA состоит в разбиении текущего кадр на непересекающиеся блоки одного размера и поиске наиболее схожего блока текущего кадра с блоком из заданной области на предыдущем кадре. Наиболее популярными метриками этого алгоритма являются энергия остатка (SSD) и суммарная абсолютная разность (SAD) [1].

Суммарная квадратичная ошибка (Sum of Square Difference – SSD) вычисляется по формуле (1).

$$SSD = \sum_{p \in Obj} (Y_{Cur}(p) - Y_{Rf}(p))^2 \quad (1)$$

Суммарная абсолютная разница (Sum of Absolute Difference – SAD) – по формуле (2).

$$SSD = \sum_{p \in Obj} \left| Y_{Cur}(p) - Y_{Rf}(p) \right| \quad (2)$$

где: Obj – прямоугольный блок размером N×N пикселей
Y_{Cur} и Y_{Rf} – яркость текущего и предыдущего кадров, соответственно, в точке $p=(x,y)$.

Схема кодирования с использованием метрики SAD представлена на рис. 1.

Рис. 1. Схема алгоритма кодирования одного блока кадра с использованием метрики SAD

При применении подобных метрик блок текущего кадра C сравнивается с каждым блоком-кандидатом S_i из области поиска S на предыдущем кадре по формуле (1) или (2). Получается n (количество блоков зависит от размера области поиска) значений метрики для блока $SAD(C,S_i)$. Наименьшее значение определяет координаты вектора движения блока $vec(min\ SAD(C,S_k))$. После оценки движения осуществляется его компенсация на основе полученных координат вектора, вычисляется скомпенсированный блок. Далее этот блок переводится в частотную область путем применения дискретного косинусного преобразования (ДКП). Полученные ДКП-коэффициенты квантуются. К квантованному блоку применяются коды переменной длины. Результатом такого кодера является битовая последовательность.

Реализация каждого блок схемы рис. 1 описана в стандарте ISO/IEC 14496 Part 2 («MPEG4 Visual») [2].

SAD и SSD являются упрощенными, и их применение снижает вычислительную сложность алгоритма. При этом точность оценки движения такими метриками также снижается, что не позволяет максимально сжать видео при неизменном качестве видеоизображения.

Новая метрика

В статье предложена новая метрика. Особенностью этого метода является оценка движения не по яркостным компонентам, как в описанных выше метриках, а по количеству бит на выходе кодера. Алгоритм оценки движения с использованием разработанной метрики описан ниже, схема его работы показана на рис. 2.

1. Для фиксированного блока C текущего кадра вычисляются все возможные в области поиска S разностные блоки-кандидаты $C\text{-}S_i$ и определяются координаты всех векторов движения.
2. Каждый разностные блок переводится в частотную область путем применения дискретного косинусного преобразования (ДКП) . Таким образом, для одного фиксированного блока получается множество ДКП блоков-кандидатов $DCT(C\text{-}S_i)$, i=1,2..n.
3. Все ДКП-блоки квантуются. Вычисляется $QUANT_i$, i=1,2..n.
4. Применяется зигзаг-преобразование внутри каждого блока. Зигзаг-преобразование – это перестановка пикселей блока с целью более эффективного кодирования.
5. Квантованные коэффициенты кодируются с использованием заранее фиксированной системы кодов переменной длины. Про-

изводится подсчет количества бит для каждого блока-кандидата VLC_i.

6. Из всех полученных значений количества бит выбирается минимальное. Наименьшее значение определяет вектор движения и остаточный блок.

В рассмотренной литературе [3] подобная метрика не встречалась, поэтому назовем предложенный метод New Method (NM).

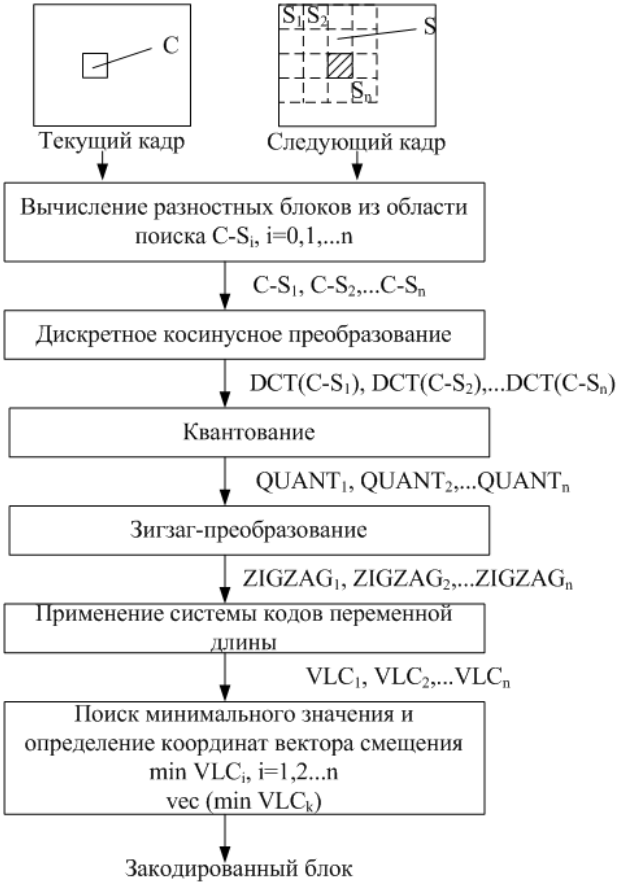


Рис. 2. Схема алгоритма кодирования одного блока кадра с использованием новой метрики

Модель кодера

Для исследования эффективности применения NM разработана модель кодера в соответствии со стандартом ISO/IEC 14496 Part 2 («MPEG4 Visual») [2]. Стандарт предполагает выбор следующих параметров кодирования: коэффициент квантования от 1 до 31 и степень билинейной интерполяции для увеличения (масштабирования) изображения в 2 и в 4 раза по каждому из измерений. В модели выбран коэффициент квантования, равный 1, т.к. при таком значении наименьшие потери качества изображения. С целью увеличения скорости моделирования не используется интерполяция. Также стандарт определяет систему кодов переменной длины (VLC).

Модель разработана на языке C++ на основании открытого кода кодека XviD [4]. Размер видеокадра задается вручную. Модель осуществляет покадровое кодирование: преобразование кадра из формата RGB в YCbCr [5], оценка движения, компенсация движения, дискретное косинусное преобразование (ДКП), квантование, зигзаг-преобразование и применение системы кодов переменной длины. Выходными данными модели является количество бит и среднее время обработки одного кадра.

Движение на видео оценивается по яркостным компонентам Y, не используя цветоразностные (формат кадра YCbCr). Для цветоразностных выполняется только компенсация движения и кодирование. Исследование влияния метрики на эффективность сжатия видеоизображения позволяет не использовать информацию о цвете, а ограничится яркостной компонентой кадра.

Оценка движения в модели осуществляется двумя видами алгоритмов: полный перебор [1] и бриллиантовый поиск. Полный перебор для оценки движения использует все возможные блоки в области поиска в соответствии с выбранным уровнем интерполяции. Бриллиантовый поиск представляет собой упрощенный алгоритм выбора блоков-кандидатов в области поиска. Бриллиантовый поиск представляет собой перебор по шаблону в форме ромба [6].

Исследование влияния метрики на сжатие

Исследование проводилось на тестовом видео foreman.avi с расширением 176x144 пикселя и количеством кадров, равным 400, с использованием персонального компьютера на основе процессора AMD Athlon 1,60 GHz. Тестовое видео foreman.avi имеет как статический характер сцен – это движение губ и руки, так и динамический – движение камеры по горизонтали. На видео четко различимы объекты, т.е. яркость изображения позволяет произвести оценку движения. Использование одно-

го тестового видео является достаточным в рамках проводимого исследования по причине того, что оно имеет необходимые типы движения в кадре и яркость.

Был произведен анализ результатов шести способов оценки движения: метода бриллиантового поиска с применением SAD, SSD, NM и полного перебора с этими же метриками. Также был промоделирован алгоритм полного перебора с метриками после ДКП (DCT), квантования (QUANT) и зигзаг-преобразования (ZIGZAG).

Оценка и компенсация движения по формулам SSD и SAD и дискретное косинусное преобразование не влияют на качество изображения. В модели потери качества на других стадиях кодирования одинаковы для разных способов оценки движения. Поэтому качество изображения после кодирования для всех шести исследуемых методов остается постоянным.

Для анализа полученных результатов используется единственный численный критерий, определяющий степень сжатия видео, – среднее количество бит одного кадра. Чем меньше значение выбранного критерия, тем точнее оценка движения и больше степень сжатия. Результаты исследования приведены в табл. 1. Обозначения в таблице: DS – метод бриллиантового поиска, FS – метод полного перебора, SAD – суммарная абсолютная разница, SSD – суммарная квадратичная ошибка, NM – новый метод, DCT – метрика после ДКП, QUANT – метрика после квантования, ZIGZAG – метрика зигзаг- преобразования.

Табл. 1. Количественный критерий качества сжатия и время работы алгоритмов

Метод	**Средний размер кадра, Кбит**	**Среднее время обработки кадра, с**
DS-SAD	205,769	0,60
DS-SSD	205,396	0,59
DS-DCT	202,802	4,14
DS-NM	200,885	3,59
FS-SAD	194,615	1,11
FS-SSD	193,658	0,98
FS-DCT	183,085	531,68
FS-QUANT	181,866	611,33
FS-ZIGZAG	181,866	613,00

Метод	Средний размер кадра, Кбит	Среднее время обработки кадра, с
FS-NM	174,831	518,00

На основании полученных результатов можно сделать вывод о том, что применение новой метрики позволяет повысить степень сжатия видео на 9,7% лучше по сравнению с SSD при алгоритме полного перебора и на 2,2% при бриллиантовом поиске с теми же метриками. Алгоритм полного перебора дает лучшие показатели сжатия на 5,7 % относительно бриллиантового поиска при одинаковых условиях моделирования, например при SSD. Полный перебор и метрика после DCT на 5,5% увеличивает степень компрессии видео по сравнению с SSD. Метрики после квантования (QUANT) и зигзаг-преобразования (ZIGZAG) при полном переборе дают одинаковый результат и на 6,1% лучше сжимают видео по сравнению с SSD. При этом качество изображения для всех случаев исследования остается постоянным.

На практике наиболее популярным является применение упрощенного блочного алгоритма, например бриллиантового поиска, и метрики SAD с целью снижения вычислительной сложности. При сравнении степени сжатия DS-SAD и FS-MN выявлено, что FS-NM на 15,0% эффективнее сжимает видео при одинаковом качестве изображения, т.е. применение такого метода позволит повысить качество изображения при одинаковом размере видеофайла. Полученный результат является существенным в условиях растущих требований к качеству видео.

Недостатком NM и полного перебора является большое количество арифметических операций по сравнению со стандартными метриками и бриллиантовым поиском, следовательно, и большее время обработки кадра относительно бриллиантового поиска и SAD или SSD (см. таблицу 1). Поэтому данный метод представляет только научный интерес и применение его на практике не допустимо.

Выводы

В результате исследования было продемонстрировано, что сжать видео можно эффективнее, чем стандартными алгоритмами оценки движения на 15,0%.

Применение на практике предложенного в статье эффективного метода сжатия не допустимо по причине большого времени обработки кадра по сравнению с популярными методами.

Результаты представленного в статье исследования станут основой для дальнейшей исследовательской работы в области усовершенствования стандартных методов или разработки нового алго-

ритма оценки движения. Новый алгоритм должен быть сравним по времени обработки видео со стандартными, но гарантировать большую степень сжатия видео и точность оценки, как алгоритм полного перебора с NM.

Список источников

1. Кубасов, Д. Обзор методов компенсации движения *[Электронный ресурс]* / Компьютерная Графика и Мультимедиа: сетевой журнал. – Электрон. текстовые дан. – М.: computergraphics.ru, 2005. – Режим доступа: http://cgm.computergraphics.ru/content/view/76.

2. International standard ISO/IEC 14496-2:2001(E). Information technology – Coding of audio-visual objects – Part 2: Visual.

3. *Сжатие видео – Motioninfo [Электронный ресурс] / Всё о сжатии данных, изображений и видео* – Электрон. текстовые дан. – М.: compression.ru, 2001. – Режим доступа: http://www.compression.ru/download/video_motioninfo.html.

4. Xvid codec. *Developer Downloads [Электронный ресурс]* / Xvid codec – Электрон. текстовые дан. – www.xvid.org, 2011. – Режим доступа: http://www.xvid.org/Downloads.43.0.html.

5. Ричардсон, Я. Видеокодирование. H.264 и MPEG-4 – стандарты нового поколения / Ян Ричардсон. М.: Техносфера, 2005, 368 с.

6. Jo Yew Tham, Surendra Ranganath, Maitreya Ranganath, Ashraf Ali Kassim. A Novel Unrestricted Center-Biased Diamond Search Algorithm for Block Motion Estimation, IEEE transactions on Circuits and Ыystems for Video Technology, Vol. 8, No. 4, August 1998.

Идентификация пользователей социальных сетей в Интернет на основе социальных связей

С. Бартунов[1], А. Коршунов[2]

[1] sbartunov@gmail.com, [2] korshunov@ispras.ru

Институт системного программирования РАН, Москва, Россия

Аннотация. В настоящее время мы переживаем бум социальных интернет-сервисов. Каждый год появляется множество как общенаправленных, так и нишевых социальных сервисов, и для активных пользователей Интернет типично иметь несколько профилей в различных социальных сетях. Обнаружение профилей, принадлежащих одному человеку, в нескольких социальных сетях, позволяет получить более полный социальный граф, что может быть полезно во многих задачах, таких как информационный поиск, интернет-реклама, рекомендательные системы и т. д. В данной работе предлагается оригинальная «JLA-модель» идентификации пользователей, основанная на модели условных случайных полей и совместно использующая как аттрибуты пользовательских профилей, так и социальные связи. Предложенный подход особенно полезен в случаях, когда информация о пользовательских профилях малополезна, недоступна или скрыта из соображений приватности. Эксперименты на данных из двух популярных в настоящий момент социальных сетей «Facebook» и «Twitter» показали, что данный подход работает эффективнее существующих решений и способен сопоставить профили, которые невозможно сопоставить, используя только информацию об аттрибутах.

Ключевые слова: идентификация пользователей; анализ социальных сетей; условные случайные поля; графические модели; обработка графов; машинное обучение.

Игнатов Д. И., Яворский Р. Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16–18 марта, 2012

Введение

Еще несколько лет назад было трудно предположить, каким огромным будет присутствие социальных приложений в нашей жизни. Тем не менее, сейчас мы живем в эпоху онлайновых социальных сетей. Ввиду беспрецедентного масштаба социальных сервисов и, как следствие, большого количества информации, заключенной в них, привлечение социальной составляющей при решении многих задач может значительно улучшить результаты.

Основной проблемой при задействовании социальной информации является её фрагментированность среди множества различных онлайновых социальных сетей. Несмотря на то, что существуют попытки по обеспечению единого способа взаимодействия между различными социальными платформами (например, Google Open-Social [1]), они не получили широкого использования, а новые социальные сервисы продолжают появляться. Процесс идентификации пользователей необходим для объединения различных социальных сетей и получения более полной картины о социальном поведении данного пользователя в «Интернет».

В данной работе мы фокусируемся на задаче идентификации пользователей в т.н. *локальной перспективе*. Это подразумевает сопоставление профилей в рамках списка контактов некоторого центрального пользователя. Такая задача часто возникает при работе с контактами в социальных мета-сервисах, которые, в частности, могут служить для объединения новостных потоков в поддерживаемых социальных сервисах (таких как «Path»[2]) или предоставления единой системы обмена сообщениями (сервисы «Meebo» и «imo»[3]). Другая область, в которой возникает подобная задача, это функция автоматического объединения контактов, часто присутствующая в современных мобильных устройствах (например, в смартфонах на платформе «Android»).

Постановка задачи. Рассмотрим два социальных графа $\langle A, B\rangle$. Под социальным графом будем понимать граф, узлы которого представлены пользовательскими профилями с различными аттрибутами (например, имя, день рождения, родной город и т. д.), а ребра социальными связями между профилями. Эти связи могут быть как направленными, так и ненаправленными в зависимости от семантики отношений, которые они представляют.

Задача идентификации пользователей заключается в поиске как можно большего числа правильно определенных пар профилей (v, u) таких,

[1] http://code.google.com/apis/opensocial/
[2] http://path.com/
[3] http://imo.im/

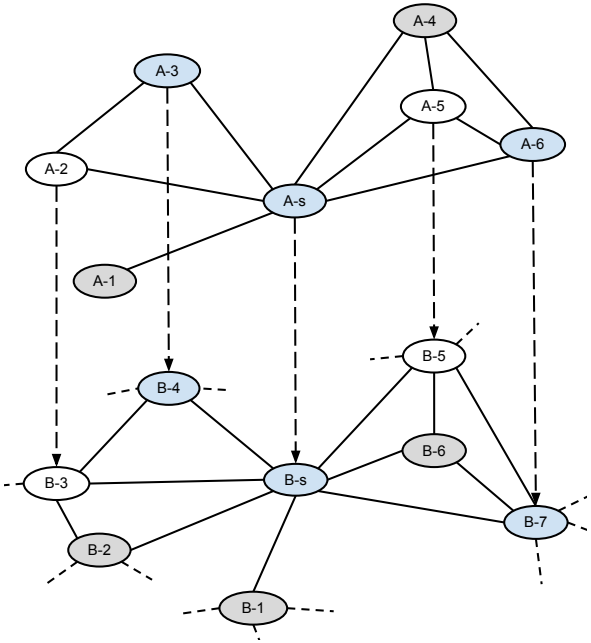


Рис. 1. Результат идентификации пользователей. Пунктирные стрелки обозначают проекции между профилями. Для вершин, закрашенных синим, проекции были известны заранее, проекции для незакрашенных вершин были установлены алгоритмом, для вершин, закрашенных серым, проекции не были найдены

что $v \in A, u \in B$, принадлежащих одному и тому же реальному человеку. Сопоставленный профиль для профиля $v \in A$ мы будем обозначать как $\mathrm{pr}(v) \in B$ и называть *проекцией* профиля $v \in A$ в B, а множество всех проекций $\{\mathrm{pr}(v)\}_{v \in A}$ профилей из A в B как $\mathrm{PR}(A)$. Если же для профиля $v \in A$ не найдено подходящей проекции, то проекцию для v будем называть *нейтральной* и обозначать как $\mathrm{pr}(v) = \mathbf{N}$. Пример двух таких социальных графов $\langle A, B \rangle$ и сопоставленных пар профилей изображен на рис. 1.

Так как в нашей работе мы рассматриваем задачу идентификации в *локальной перспективе*, то подразумевается, что графы A и B имеют структуру *эго-сетей* (англ. *ego-network*) некоторого пользователя. Эго-сеть вершины v представляет из себя граф, состоящий из вершины v и всех вершин, расстояние от которых до v не превышает двух. Такое ограничение отражает реальные ограничения использования социальных

приложений, в которых для предоставлении какой-либо информации о социальных связях требуется непосредственное разрешение пользователя.

Обзор существующих методов

Идентификация пользователей. На момент написания данной работы наиболее значительным трудом по идентификации пользователей является работа Veldman [12], в которой представлено множество эвристик, использующих как информацию о профилях, так и связей между ними. Похожие исследования описаны в [8, 4, 9, 13]. Motoyama и др. [8] сопоставляли пары профилей между сетями «Facebook» и «MySpace», в свою очередь Gaewon и др. [4] решали аналогичную задачу для «Twitter» и «EntityCube». Raad и др. [9] в своем исследовании генерировали случайные социальные графы со случайно сформированными профилями и применяли к ним многочисленные сложные эвристики с целью не упустить ни одного потенциально полезного источника информации, доступного в социальной сети. В работе Vozecky и др. [13] профили из «Facebook» и «StudiVZ» представлялись как векторы признаков, к которым в последствии применялись операции точного, частичного и нечеткого сравнения, по результатам которых проводилась идентификация.

Общая схема работы систем идентификации пользователей. Несмотря на то, что описанные выше исследования применялись к данным самых разных социальных сервисов, не составляет труда выделить и проанализировать общую схему работы этих систем:

1) Приведение данных из полей профилей из двух социальных сетей к некоторому общему виду (например, вектору, элементами которого являются поля профилей)
2) Попарное применение техник нечеткого сравнения между профилями из одной сети и профилями из другой
3) Подсчет результирующего показателя *похожести* между профилями и отсечение всех парных результатов, для которых этот показатель ниже некоторого порогового значения

После этого все оставшиеся пары считаются сопоставленными между собой и принадлежащими одному пользователю.

Несмотря на относительно неплохое качество работы этих систем они все имеют общий недостаток — слишком простую модель сравнения текстовых аттрибутов профилей. При этом социальная информация не учитывается, либо учитывается слишком слабо. При этом информация, содержащаяся в профилях, достаточно ненадежна, так как данные, указанные пользователям, в разных социальных сетях могут сильно отличаться,

быть скрытыми из-за настроек приватности или не поддерживаться в актуальном состоянии.

Для улучшения этого общего подхода необходимо привлечь дополнительные источники данных, в частности информацию о социальных связях. В некоторых работах для этого применяется техника сравнения *частично сопоставленных* списков контактов [12], которая заключается в подсчете показателя похожести между множествами профилей, которые ранее были сопоставлены по именам. Очевидно, что подобная эвристика может привести к *смещению* в результатах. Подход, представленный в этой работе, активно использует социальные связи обеих рассматриваемых социальных сетей путем сравнения оригинальных списков контактов, естественным образом комбинируя их с информацией аттрибутов профилей, благодаря чему лишен многих недостатков существующих систем идентификации пользователей.

Разрешение сущностей. Помимо описанной выше задачи идентификации пользователей, существует также ряд близких задач, результаты которых могут быть использованы и в применении к объединению социальных графов. Одной из таких задач является *разрешение сущностей* (англ. *entity resolution*), которая заключается в определении записей базы данных, относящихся к одному и тому же объекту реального мира (не обязательно описывающие его). В работе [10] авторы строят сеть марковской случайной логики, узлами которой являются атомарные утверждения о записях базы данных с весом от 0 до 1 в зависимости от истинности или ложности утверждения, а ребрами логические связи между ними, после чего ищут оптимальную (*наиболее правдоподобную*) конфигурацию истинности утверждений, используя информацию о логических зависимостях. Подобный подход был также применен и в работе [11] для задачи устранения дубликатов (англ. record deduplication) в графе цитирования авторов научных статей, где вместо марковской случайной логики применены условные случайные поля [6].

Аналогичный подход, наиболее похожий на представленный в данной работе, описан в [11], где для устранения дубликатов (англ. record deduplication) в графе цитирования авторов научных статей используются условные случайные поля [6]. Основной идеей было построить условное случайное поле и представить узлами атомарные утверждения вида “являются ли эти две записи дубликатами?” с возможными значениями “да” или “нет” и узлы-улики с информацией о близости аттрибутов рассматриваемых объектов. Ребра же между утверждениями обозначали бы *условную зависимость* между ними, в то время как ребра между утверждениями и уликами — правдоподобность данного утверждения при известном значении похожести аттрибутов. После чего из данной

модели возможно сделать *вывод* оптимальной конфигурации ответов на утверждения.

Две данные работы демонстрируют эффективность решения задачи, похожей на идентификацию пользователей как совокупности нескольких взаимозависимых задач, а также применения графических вероятностных моделей. Тем не менее, «JLA-модель» хоть и также основана на условных случайных полях, но значительно отличается от описанных выше подходов следующими аспектами:

— Более компактное и в то же время более естественное представление модели условных случайных полей, которая строится на основе одного из социальных графов, а не графе связанных утверждений
— Помимо информации о строковой похожести аттрибутов объектов используется информация о графовой близости вершин
— Размер и подробность графической модели в описанных выше подходах делают вывод ответа неэффективным на относительно больших данных, в то время как предлагаемый в данной работе подход использует более компактное представление и при помощи описанных техник оптимизации делает возможным вывод решения даже для больших социальных графов, в том числе параллельно.

«JLA-модель»

«JLA-модель» (от англ. joint link-attribute), представленная в данной работе, основывается на следующих соображениях:

1) Необходимо совместно использовать как аттрибуты профилей, так и социальные связи между ними
2) Задачи выбора проекций для связанных вершин в графе A взаимосвязаны, иначе говоря, выбор проекции для некоторой вершины зависит от значений проекций связанных с ней вершин.
3) Если две вершины в графе A связаны, их проекции должны иметь как можно меньшие расстояния в графе B.

В данной работе из соображений простоты и эффективности в качестве функции расстояния в графе B используется коэффициент Дайса:

$$\mathsf{network\text{-}distance}(v,u) = 1 - \frac{2 \cdot w(L_v \cap L_u)}{w(L_v) + w(L_u)}, v, u \in B,$$

где L_v и L_u — множества вершин, связанных с v и u соответственно, а $w(L) = |L|$ — вес этих множеств. Причем $0 \leqslant \mathsf{network\text{-}distance} \leqslant 1$.

Предполагается, что один из графов $\langle A, B\rangle$ является ненаправленным. В дальнейшем без ограничения общности будем считать таким графом A.

На основе графа A строится модель условных случайных полей [6], в которой множество наблюдаемых переменных представлено вершинами графа A: $\mathbf{X} = \{\mathbf{x}_v \mid v \in A\}$, с каждой из которых ассоциирована одна скрытая переменная $\mathbf{Y} = \{\mathbf{y}_v \mid v \in A\}$, определяющая проекцию данной вершины: $\mathbf{y}_v = \mathrm{pr}(v) \in B$. Эти пары переменных связаны фактором унарной энергии Φ. Две скрытые переменные $\mathbf{y}_v$ и $\mathbf{y}_u$ связаны фактором бинарной энергии Φ тогда и только тогда, когда связаны вершины v и u в графе A.

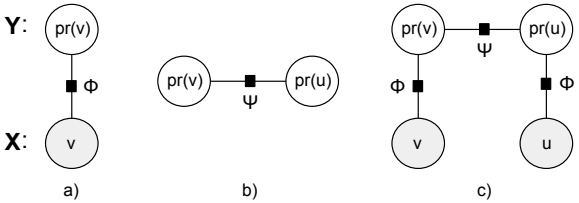


Рис. 2. Структура «JLA-модели». a) идентификация, основанная на аттрибутах профилей b) идентификация, основанная на социальных связях c) полная модель

Совместная природа модели выражается в том, что похожесть аттрибутов профилей учитывается при помощи унарной энергии Φ, а социальные связи — через бинарную энергию Ψ (см. рис. 2). Это делает модель адаптивной по отношению к данным, которые доступны для использования. Так, если отсутствует информация о социальных связях, то $\Phi \equiv 0$ и модель приобретает форму стандартной системы идентификации. В то же время, если данные анонимизированны, то есть, социальные связи присутствуют, но вся полезная для идентификации информация в профилях убрана, то $\Psi \equiv 0$ и используется только структура социальных графов.

Таким образом, модель порождает следующее вероятностное распределение:

$$p(\mathbf{Y}|\mathbf{X}) = \exp(-E(\mathbf{Y}|\mathbf{X})),$$
$$E(\mathbf{Y}|\mathbf{X}) = \sum_{v \in V} \Phi(\mathbf{y}_v|\mathbf{x}_v) + \sum_{(v,u)\in E} \Psi(\mathbf{y}_v, \mathbf{y}_u),$$

где E это *функционал энергии*, моделируемый функцией *унарной энергии* Φ и функцией *бинарной энергии* Ψ. Обе энергетические функции вещественны и неотрицательны.

Унарная энергия отвечает за похожесть профиля в A и его проекции в B с точки зрения полей профилей:

$$\Phi(\mathbf{y}_v|\mathbf{x}_v) = \alpha(v) \cdot \mathsf{profile\text{-}distance}(v, \mathrm{pr}(v)),$$

а бинарная энергия отвечает за близость между проециями вершин v и u в графе B:

$$\Psi(\mathbf{y}_v, \mathbf{y}_u) = \mathsf{network\text{-}distance}(\mathrm{pr}(v), \mathrm{pr}(u)).$$

Здесь $0 \leqslant \mathsf{profile\text{-}distance} \leqslant 1$, и $\alpha(v) = \log(degree(v)) \geqslant 0$ — коэффициент баланса между унарной и бинарной энергией.

Для двух данных графов $\langle A, B\rangle$ существует оптимальная конфигурация проекций:

$$\mathbf{Y}^* = \underset{Y}{\mathrm{argmin}} E(\mathbf{Y}|\mathbf{X}), \tag{1}$$

которая минимизирует функционал энергии, максимизируя при этом правдоподобие модели.

Похожесть профилей. Для определения похожести профилей из разных социальных сетей необходимо учесть все доступные поля, содержащиеся в них, с использованием различных функций нечеткого сравнения. Для рассматриваемых в данной статье сетей «Twitter» и «Facebook» использовалась схема сравнения, изображенная в табл. 1.

Табл. 1. Схема сравнения полей профилей в сетях Facebook и Twitter

Facebook	**Twitter**	**Функция сравнения**
Name	Name	VMN
	Screen name	Screen Name measure
Website	URL	URL measure

«URL measure» проверяет, упоминается ли в одном профиле URL второго профиля. «Screen Name measure» проверяет на полное совпадения имени в «Facebook» и отображемого в адресе имени пользователя в «Twitter». VMN это функция близости, заимствованная из [13].

Путем применения функций близости к соответсвующим полям двух профилей, формируется *вектор похожести* $V(v, \mathrm{pr}(v))$. Причем если хотя бы одном из профилей поле отсутствует или недоступно, то соответствующий элемент ветока V неопределен. Вектор V используется как набор признаков, на которых обучается специальный бинарный классификатор, определяющий принадлежат ли профили v и $\mathrm{pr}(v)$ одному и тому же человеку.

Таким образом, можно определить:
$\text{profile-distance}(v, \text{pr}(v)) = \text{P}(\text{разные люди}|V(v, \text{pr}(v))$, поскольку и унарная энергия, и вероятность, возвращаемая классификатором принадлежат интервалу $[0, 1]$. Сравнение различных алгоритмов классификации при помощи кросс-валидации с 3-я разбиениями представлено в табл. 2.

Табл. 2. Сравнение классифакторов похожести унарной энергии

алгоритм	полнота	точность	F_1
Naive Bayes	**0.862**	0.308	0.453
C4.5	0.569	0.86	0.685
C4.5 с MultiBoosting	0.669	**0.879**	**0.76**

Алгоритм C4.5 с MultiBoosting был выбран для дальнейший экспериментов, как показавший наибольшую эффективность. Следует отметить, что ни один из классификаторов не смог „объяснить" принадлежность профилей на основании только полей профилей.

Заранее известные проекции

«JLA-модель», также как и многие другие алгоритмы, использует информацию о заранее известных проекциях (в зарубежной литературе *anchor nodes* или *seed nodes*). Такие проекции могут быть сообщены алгоритму перед началом работы, или просто могут считаться таковыми если $\text{profile-distance}(v, \text{pr}(v)) \leqslant \Delta$.

Для каждой вершины v с заранее известной проекцией $\text{anchor}(v)$ значения энергий зафиксированы:

$$\begin{aligned} &\Phi(\mathbf{y}_v \mid \mathbf{x}_v) = \infty \\ &\Psi(\mathbf{y}_v, \mathbf{y}_u) = \Psi(\mathbf{y}_u, \mathbf{y}_v) = \infty \end{aligned} \qquad \text{if } \mathbf{y}_v \neq \text{anchor}(v)\ \forall u$$

Заранее известные проекции повышают точность полученных результатов, а также значительно уменьшают вычислительное время работы алгоритма. Информация распространяется от вершин с заранее известными проекциями (см. рис. 3), и для подграфов графа A, связанных с остальным графом только посредством таких вершин, проекции могут быть установлены независимо, что позволяет параллельно обрабатывать большие графы.

Нейтральные проекции и очистка результатов. Поскольку выбрать разумные фиксированные значения функций близости для нейтральных проекций не представляется возможным, для очистки результатов (1) от

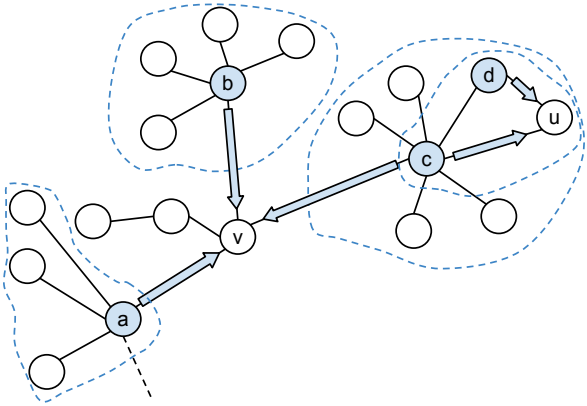


Рис. 3. Распространение информации от вершин с заранее известными проекциями. Проекции для вершин внутри областей, размеченных пунктирами, могут быть установлены независимо

неправильно выбранных проекций необходимо привести соответствующую процедуру.

Очевидной методикой очистки результатов может служить повторный вывод ответа (1) при построении модели на графе B (то есть, с противоположным направлением проецирования) с последующим удалением всех вершин, для которых выбранная проекция не совпала. Иначе говоря, в результаты попадают только вершины, которые были *взаимно* размечены при обратном направлении проецирования. Несмотря на простоту и интуитивность, данная процедура очистки требует повторогого вывода ответа, что может быть слишком затратным с точки зрения времени выполнения для относительно больших графов, а также достаточно груба, так как не учитывает *причину*, по которой была допущена ошибка.

В качестве более продвинутого решения предлагается схема обучения бинарного классификатора (C4.5 с MultiBoosting), который, используя информацию о контекстке каждой вершины в A, решает, правильно ли для неё выбрана проекция. Для этого используются следующие признаки:

1) $\mathsf{profile\text{-}distance}(v, \mathrm{pr}(v))$

2) Средняя графовая близость к проекциям смежных вершин

3) Доля заранее известных проекций среди смежных вершин

4) Взаимо-согласованность смежных вершин с заранее известными проекциями:

$$\frac{1}{n} \cdot \sum_{v} \frac{1}{n-1} \sum_{u \neq v} \mathsf{network\text{-}distance}(\mathrm{pr}(v), \mathrm{pr}(u))$$

Сравнение различных алгоритмов классификации при кросс-валидации с 3-я разбиениями приведено в табл. 3.

Табл. 3. Эффективность классификаторов очистки

алгоритм	полнота	точность	F_1
Naive Bayes	0.762	0.256	0.383
Support Vector Machine	0.662	0.935	0.775
C4.5	0.715	**0.939**	0.812
C4.5 с MultiBoosting	**0.844**	0.902	**0.872**

Результаты работы

Предложенный в данной работе подход был протестирован на данных из двух наиболее популярных на данный момент социальных сетей «Facebook» и «Twitter». Для 16 центральных пар профилей в обоих социальных сетях были загружены и размечены „эго-сети", преимущественно самими владельцами профилей. Эти основные данные были использованы для настройки всех алгоритмов машинного обучения и для тестирования точности всех алгоритмов с использованием кросс-валидации с 3-я разбиениями.

Кроме того, для тестирования «JLA-модели» была составлена дополнительная тестовая выборка. Поскольку без привлечения владельцев аккаунтов социальных сервисов не возможно достоверно разметить данные, то дополнительная выборка использовалась в полуавтоматическом режиме для задачи *повторной идентификации*.

Полная статистика по использованным данным содержится в табл. 4.

Поскольку связи в социальной сети «Twitter» направленные и имеют семантику подписки, а не дружбы как в «Facebook», то при построении эго-сети было решено рассматривать только вершины, которые взаимно подписаны друг на друга, для симуляции отношений дружбы. Таким образом, при построении модели условных случайных полей на графе «Twitter» использовался ненаправленный граф, как того требует модель. При расчете расстояний на графе «Twitter» в качестве списка контактов также использовались списки взаимно подписанных друг на друга профилей.

Табл. 4. Экспериментальные данные

	Twitter	**Facebook**
Основная выборка		
# центральных пользователей	16	
# профилей	398	977
# связей	1 728	10 256
# сопоставленных профилей	141	
# заранее известных проекций	71	
Дополнительная выборка		
# центральных пользователей	17	
# профилей	1 499	7 425
# связей	15 943	172 219
# сопоставленных профилей	161	

Базовые алгоритмы. В качестве базовых алгоритмов, с которыми проводилось сравнение «JLA-модели», были выбраны алгоритмы, использующих только информацию о полях профилей, поскольку именно так работает большинство систем идентификации пользователей. Базовые алгоритмы сопоставляют каждому профилю из графа A не более одного профиля из графа B, так чтобы с одной стороны максимизировалась некоторая функция близости между профилями и их проекциями, а с другой стороны не было двух и более профилей, спроецированный в один и тот же профиль в графе B. Таким образом, базовые алгоритмы решали задачу *оптимального парасочетания*.

Рассматриваемые алгоритмы использовали следующие функции близости:

1) Взвешенная сумма элементов вектора похожести $V(v, \mathrm{pr}(v))$. Веса подбирались при помощи линейной регрессии, исходя из предположения, что для правильно выбранных проекций, сумма должна равняться 1.

2) $\mathsf{profile\text{-}distance}(v, \mathrm{pr}(v))$. Иначе говоря, данный алгоритм использовал ту же функцию похожести, что и «JLA-модель».

Базовые алгоритмы также имели пороговые значения, ниже которых значения похожести не рассматривались. Эти значения были настроены таким образом, чтобы достигалась максимальная точность, так как именно это ожидается от реальной системы идентификации.

Оценка качества алгоритмов. Рассматриваемые алгоритмы оценивались с точки зрения общеизвестных метрик *точности* и *полноты*:

$$\text{полнота} = \frac{\text{true-positives}}{\text{true-positives} + \text{false-negatives}}$$

$$\text{точность} = \frac{\text{true-positives}}{\text{true-positives} + \text{false-positives}}$$

При тестировании «JLA-модель» использовалась по умолчанию с процедурой очистки результатов при помощи обученного классификатора, а также с наивной техникой взаимной проекции при противоположных направлениях. Для получения списка заранее известных проекций использовались результаты второго базового алгоритма.

Результаты оценок качества алгоритмов отображены в табл. 5. Практически все алгоритмы достигли максимальной точности. Таким образом, основной трудностью было, сохраняя высокий показатель точности, сопоставить как можно большую часть профилей, тем самым достигнув максимального показателя полноты.

Табл. 5. Оценка качества алгоритмов на основной выборке

алгоритм	полн.	точн.	F_1
безразличные к направлению проекции			
Базовый 1 (взвешенная сумма)	0.45	0.94	0.61
Базовый 2 (вероятностная похожесть)	0.51	**1.0**	0.69
JLA, взаимн. проекц., аноним.	0.6	**1.0**	0.76
JLA, взаимн. проекц.	0.66	0.99	0.79
Twitter $\rightarrow$ Facebook			
JLA, анонимн. ($\Phi \equiv 0$)	0.62	**1.0**	0.77
JLA	0.79	**1.0**	0.89
Facebook $\rightarrow$ Twitter			
JLA, анонимн. ($\Phi \equiv 0$)	0.61	**1.0**	0.76
JLA	**0.8**	**1.0**	**0.89**

Второй базовый алгоритм незначительно обогнал первый алгоритм и обозначил предел возможностей систем идентификации, использующих только аттрибуты профилей.

Подход, использующий «JLA-модель», в среднем на 29 % превзошел второй базовой алгоритм по показателю полноты, сохранив при этом максимальную точность. При этом включенная техника взаимного проецирования значительно занизила полноту, тем самым эмпирически подтвердив целесообразность использования классификатора для очистки результатов.

«JLA-модель» даже в условиях анонимизированных данных, но при наличии 50 % заранее известных проекций, позволила дополнительно сопоставить около 10 % процентов профилей, иначе говоря, *деанонимизировать* их. Это подтверждает как значимость социальных связей в задачах идентификации и деанонимизации пользователей, так и высокую для них пригодность «JLA-модели».

Повторная идентификация

Несмотря на то, что направление проекции на основной выборке практически не отразилось на результатах, некоторые отличия можно наблюдать при тестировании «JLA-модели» на дополнительной выборке.

Поскольку дополнительная выборка не была размечена вручную, на ней невозможно адекватно оценить качество работы алгоритма. Тем не менее, можно оценить, как хорошо алгоритм способен повторно идентифицировать профили, которые ранее были сопоставлены по профилям при помощи второго базового алгоритма.

Некоторая часть таких профилей фиксируется случайным образом, после чего у всех возможных проекций удаляется вся информация из профилей. Таким образом, правильные проекции для них могут быть найдены только благодаря связым и информации в виде оставшихся идентифицированных профилей.

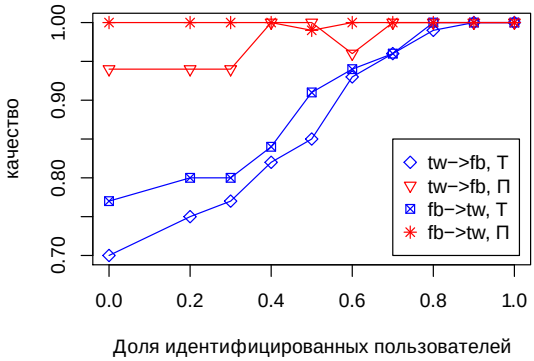


Рис. 4. Влияние доли известных идентифицированных пользователей на качество «JLA-модели»

На рис. 4 отражена зависимость показателей точности и полноты среди идентифицированных профилей в зависимости от числа профилей

с известными проекциями в среднем. Проекции, которые алгоритм определил для всех остальных профилей не учитываются. Поскольку социальный граф «Facebook» сильно более связный чем «Twitter» (согласно табл. 4), то при построении модели на графе «Facebook» даже при малой доле заранее известных проекций, информация от них распространялась лучше, и таким образом в среднем достигалось более высокое качество. Этот эксперимент демонстрирует значимость связности при выборе графа для построения вероятностной модели, а также показывает, что при знании 80 % проекций «JLA-модели» удалось найти остальные 20 %.

Заключение

Результаты экспериментов на данных актуальных и в то же время небогатых по возможностям сравнения профилей социальных сетей показали важность социальных связей в задаче идентификации пользователей, а также их эффективное использование в предложенной «JLA-модели».

Несмотря на успешное применение модели для локальной перспективы, перенос её для глобальной перспективы нетривиален и не представляется возможным без использования достаточно большого числа заранее известных проекций. Это связано в первую очередь с большой вычислительной сложностью процесса вывода решения в условных случайных полях, который потребует значительных оптимизаций. Одна из таких оптимизаций — разбиение задачи на независимые подзадачи была предложена в данной статье, однако помимо этого также необходимо сужение множества возможных проекций для каждой вершины.

Открытым вопросом также является, как работает предложенный подход с социальными графами различных топологий, а не только с эго-сетями, а также насколько «JLA-модель» устойчива к потере или намеренному искажению информации о социальных связях.

Список источников

1. S. Bortoli, H. Stoermer, P. Bouquet (2007). *Foaf-O-Matic — Solving the Identity Problem in the FOAF Network*. In: Proceedings of the Fourth Italian Semantic Web Workshop (SWAP2007), Bari, Italy, Dec.18-20, 2007.
2. P. Bouquet, S. Bortoli (2010). *Entity-centric Social Profile Integration*. In: Proceedings of the International Workshop on Linking of User Profiles and Applications in the Social Semantic Web (LUPAS 2010) 52-57.

3. F. Fouss, A. Pirotte, J.-M. Renders, M. Saerens. *Random-walk computation of similarities between nodes of a graph, with application to collaborative recommendation*. IEEE Transactions on Knowledge and Data Engineering, vol. 19, No. 3, March 2007.
4. Gae-won Y., Seung-won H., Zaiqing N., Ji-Rong W. *SocialSearch: Enhancing Entity Search with Social Network Matching*. EDBT 2011.
5. H. Kopcke, E. Rahm. *Frameworks for entity matching: A comparison*. Data & Knowledge Engineering, Vol. 69, No. 2. (2010), pp. 197-210.
6. J. D. Lafferty, A. McCallum, P. McCallum, C. N. Fernando. *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data*. Proceedings of the Eighteenth International Conference on Machine Learning, 2001.
7. M. Lenzerini. *Data Integration: a Theoretical Perspective*. In PODS '02: Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of Database Systems, pages 233-246. 2002.
8. Motoyama, M., Varghese, G. *I Seek You — Searching and Matching Individuals In Social Networks*. WIDM '09: Proceeding of the eleventh international workshop on Web information and data management.
9. Raad, E., Chbeir, R., Dipanda, A. *User Profile Matching in Social Networks*. 13th International Conference on Network-Based Information Systems (NBiS), 2010.
10. P. Singla, P. Domingos. *Entity Resolution with Markov Logic*. In Proc. of the Sixth International Conference on Data Mining (ICDM'06).
11. P. Singla, P. Domingos. *Multi-relational Record Linkage*. KDD Workshop on Multi-Relational Data Mining (pp. 31-48), 2004.
12. Veldman, I. (2009) *Matching Profiles from Social Network Sites*. Master's thesis, University of Twente.
13. Vosecky, J., Dan Hong, Shen, V.Y. *User identification across multiple social networks*. In Proc. of First International Conference on Networked Digital Technologies, 2009.

Типология пользователей коллаборативных платформ

А. А. Беззубцева

nstbezz@gmail.com

НИУ-ВШЭ, Москва, Россия

Witology, Москва, Россия

Аннотация. В данной работе приведен обзор существующих типологий пользователей интернет-сервисов. Особое внимание уделяется социальным сетевым сервисам, в том числе, блогам и краудсорсинговым сообществам. На основе рассмотренных типологий была выведена собственная типология пользователей отдельного класса сервисов – коллаборативных платформ инноваций. Для ее разработки применялся кластерный анализ показателей активности (создание идей, комментирование, оценивание) более 500 участников одного из проектов российской площадки инноваций Witology. Деятельность получившихся групп пользователей (неактивные, прохожие, создатели, критики, спорщики, звезды) примерно распределена по правилу «90 – 9 – 1».

Ключевые слова: краудсорсинг, типология, классификация, коллаборативная платформа, инновации, социальная сеть, сообщество, блог.

Введение

Коллаборативные платформы – сравнительно недавнее явление, особенно в России. Они набирают популярность несколько меньшими темпами (например, ср. [1] и [2]), чем социальные сети или блоги, одна-

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16-18 марта, 2012.

ко от этого интерес к участвующей в краудсорсинговых проектах аудитории не убавляется. Существующие исследования потребительского и медиа-поведения пользователей в сети нельзя полностью распространить на их поведение на платформе инноваций, а общие социологические и психологические типологии людей без привязки к вебу тем более упускают многие важные особенности, присущие только сетевому взаимодействию участников краудсорсингового проекта.

Классификация участников таких платформ не просто интересна, она также может помочь в разработке и улучшении правил сообщества, выступить в качестве упрощенной математически обоснованной системы определения компетентности участников или дополнительного мотивационного инструмента.

Основная цель данного исследования – выяснить, как обстоит дело с типированием пользователей коллаборативных платформ и проверить типологии на практике. Достигалась цель путем анализа, в том числе, с помощью математических инструментов, данных по одному из проектов российской площадки инноваций Witology.

Терминология

В статье в качестве предмета анализа часто упоминаются не только сами коллаборативные платформы, но и все социальные сетевые сервисы и даже все интернет-сервисы, поскольку типологии последних могут быть распространены на коллаборативные платформы. Терминология в данной области пока не устоялась, и приведенные ниже определения не претендуют на неоспоримость и полноту и призваны лишь дать общее представление о сфере исследования.

Под интернет-сервисом мы подразумеваем любой веб-сайт, предоставляющий любые услуги (блоги, файлообменные сети, чаты, многопользовательские игры, онлайн-магазины и т.д.). Интернет-сервисы, обеспечивающие взаимодействие людей, называют социальными сетевыми сервисами (social networking services, SNS). Они включают социальные сети (Facebook, MySpace, last.fm, LinkedIn, Orkut), блоги (LiveJournal, Tumblr, Twitter), вики (Wikipedia), медиа-хранилища (Flickr, Picasa, YouTube) и пр. [3], [4]. Социальные сетевые сервисы обычно порождают онлайн-сообщества (online communities) – группы людей, которых объединяют сходные интересы и общение через определенный интернет-сервис. Некоторые авторы [5], [6], [7] понимают термин community в более широком смысле – как всю аудиторию конкретного социального сетевого сервиса (т.е. сообществами считаются и пользователи социальных сетей, и участники вики-проектов, и даже зарегистрированные на YouTube), что, по мнению Michael Wu [8], не-

корректно. При описании типологий мы сохранили словари авторов, в остальном использовалось первое определение сообщества.

Краудсорсинговые платформы (платформы краудсорсинга) – социальные сетевые сервисы, предполагающие получение необходимых услуг, идей или контента от участников платформы, т.е. ее сообщества, в отличие от обычных сотрудников или поставщиков [9]. Краудсорсинговые (коллаборативные) платформы инноваций нацелены именно на получение идей. Часто работа на таких платформах осуществляется в рамках ограниченных по времени проектов в виде создания идей, комментирования, оценивания и других активностей, по качеству и количеству которых определяются лучшие идеи и участники (эксперты). По такой схеме работает платформа инноваций Witology [10], однако множество коллаборативных сайтов функционируют иным образом (см. список [11]). Работа на краудсорсинговых платформах (в том числе инноваций) идет обычно в рамках краудсорсинговых проектов.

Под типологией (классификацией) участников платформы мы подразумеваем как процесс выделения типов участников, так и результат этого процесса – дерево типов участников. В данной работе термины «типология» и «классификация» используются как синонимы, что не совсем правильно [12], однако не критично в данном случае.

Обзор существующих типологий

Несмотря на молодость таких явлений, как интернет-сервис и онлайн-сообщество, уже были предприняты десятки попыток классифицировать их пользователей. Приведенный обзор так или иначе затрагивает чуть больше двадцати статей последнего десятилетия, не исключено, что есть еще столько же, не учтенных в обзоре. Впрочем, Brandtzæg в «Towards a Unified Media-User Typology…» (2010) [6] утверждает, что результатом трехмесячного поиска по словам «media use», «online community», «social networking», «user», «typology», «patterns», «profiles» и т.п. в различных комбинациях по 4 библиотекам публикаций (ISI Web of Knowledge, Springer, ScienceDirect, ACM), а также с помощью Google и Google Scholar были всего 22 релевантные типологии медиаповедения пользователей (классификации пользователей сетевых игр к таковым не относились). Среди найденных нами типологий есть лишь одна [9], подходящая под временной промежуток поиска, но не указанная Brandtzæg, а также пара более поздних [13], [14] типологий, что позволяет предположить, что на самом деле число не попавших в обзор релевантных статей не так уж велико.

Некоторые из исследований [15], [16], [17] рассматривают только медиа-поведение детей или студентов, другие [18] – только поведение пользователей как клиентов интернет-магазинов. Значительная часть

ранних (до 2007 г.) типологий (напр., [19], [20]) разрабатывалась на основе данных по частоте и разнообразию использования всемирной сети и новых гаджетов, результатом чего были весьма тривиальные схожие типологии (обычно пользователей делили на «опытных», «обычных» и «неактивных», иногда разбавляя их «функционалами» и «развлекающимися»). Позже, с увеличением числа доступных пользователю действий, усложнялись и сами классификации (Horrigan [21], в частности, обнаружил десять подклассов в трех классах). Далеко не все авторы при построении моделей поведения опирались на существующие социологические или психологические теории (вероятно, это объясняется их желанием по-новому взглянуть на различия в поведении пользователей) или же упоминали уже существующие типологии других авторов. Исследование одного из них [7] не просто описательное, но еще и считается несколько неформальным, и, несмотря на это, его классификация и правило «90 – 9 – 1» пользуются большим авторитетом и популярностью.

Что касается типологий пользователей сайтов, близких по устройству к платформе инноваций, то большая часть статей сосредотачивается на анализе поведения участников социальных сетей, также есть исследования, посвященные блогам и группам новостей. Информации по типам пользователей самих платформ пока обнаружено не было, однако некоторые из найденных типологий других социальных сетевых сервисов весьма интересны и полезны как базис для изучения поведения участников краудсорсинговых проектов.

Анализ типологий позволяет предположить, что, несмотря на порой сильные различия в устройстве социальных сетевых сервисов, есть какой-то общий набор типов пользователей, из которого формируются выборки с типами для каждого сервиса, что подтверждается исследованием [6] (хотя в некоторых источниках [22] возможность создания метатипологии исключается). Неожиданностью не стало то, что некоторые типы встречаются почти во всех типологиях (например, прослойки неактивных пользователей/наблюдателей и активных пользователей/деятелей в том или ином виде есть в [5], [7], [23], [24] и частично в [25]). Далее приведены подробные описания релевантных классификаций и выделяемых в них классов.

Brandtzæg and Heim (2010)

- Sporadics (19%) – «случайные прохожие», заходят в сообщество время от времени, чтобы узнать, не писал ли им кто, и более ничего там не делают.
- Lurkers (27%) – «бездельники», наибольшая категория, также ничего не создают, а распространяют или потребляют уже соз-

данное. Отличаются от sporadics тем, что заходят в сети еще и для того, чтобы убить время.

- Socializers (25%) – «общительные», их немного меньше, чем бездельников, они заходят в сообщество, чтобы общаться, заводить новых друзей, комментировать фотографии старых, писать всем поздравления на стены и т.д.
- Debaters (11%) – «спорщики», более зрелый и образованный вариант общительного пользователя. Помимо коммуникации, менее поверхностной, чем у socializers, в социальных сетях им интересны новости и другая полезная и не очень информация.
- Actives (18%) – «активные» пользователи, которые делают все, что можно: общаются, читают, создают и смотрят видео и фотографии, заводят группы.

Исследование [5] по большей части описательное, однако в отдельном разделе его перечисляются существующие теории и другие исследования, с учетом которых выводилась новая классификация. Кластерному анализу подвергались результаты онлайн-опроса пользователей 4 норвежских социальных сетей. Выделенные группы действительно отличались по нескольким видам активности (всего 18 видов).

Budak, Agrawal, Abbadi (2010)

- Connectors – «связные», быстро сходятся с людьми и имеют очень много знакомых. С их помощью что-то новое может широко распространяться. В терминах теории графов – узел с высокой центральностью (в данном случае она определялась как out-degree).
- Mavens – «знатоки», очень информированы в силу своей любознательности, бескорыстно делятся знаниями, в верности которых обычно никто не сомневается. От них часто исходят новые идеи. Выявить знатоков на графе несколько сложнее. Они определяются с помощью данных о каскадах в графе (серия постов, содержащих ссылки на хронологически более ранний пост) – сначала по всем успешным или неуспешным инициациям каскадов вычисляется индекс влияния, затем из самых влиятельных выбираются те, кто действительно являлись оригинальными источниками информации.
- Salesmen – «продавцы», у них естественно получается убеждать людей, налаживать с ними эмоциональный контакт. Могут аргументированно доказать, что идея хороша. В статье предполагается, что причина замечательной убедительности продавцов в блогах – то, что они не сдаются, то есть чаще других пытаются

заставить других людей продолжить каскад, в котором сами продавцы участвуют.

- Translators – «переводчики», «мосты» между разными группами по интересам внутри социальной сети или блога. Переводчики могут убрать преграды на пути идеи, проинтерпретировав ее по-другому. Например, рассматривая решение некой политической проблемы в экономическом аспекте, они включат в круг заинтересованных в вопросе, до того состоящий, преимущественно, из политологов, еще и экономистов.

В статье [13] в терминах теории графов в контексте современных сетевых сообществ (особенно блогов) рассматриваются три типа людей, предложенные в 2002 году канадским журналистом Малькольмом Гладуэллом [26]. Присутствие таких людей в достаточных количествах, по мнению Гладуэлла, является причиной оглушительности успеха некоторых инноваций.

Авторами также вводится четвертая группа «переводчиков», которая, по результатам анализа более 53 миллионов постов, предоставленных MemeTracker [27], наравне с «продавцами» больше других влияет на распространение информации и срабатывание новых идей в социальных сетях и блогах.

Li, Bernoff, Fiorentino, and Glass (2007)

- Creators – пишут в блогах, создают и публикуют видео и свои веб-страницы; обычно из молодого поколения.
- Critics – отбирают полезный медиа-контент (комментируют блог или пишут обзоры и выставляют оценки); старше предыдущей группы.
- Collectors – отличаются склонностью сохранять закладки на специальных сервисах, агрегировать информацию в новостных лентах, ставить тэги.
- Joiners – много внимания уделяют социальным сетям; самая молодая группа.
- Spectators – читают блоги, смотрят видео, слушают подкасты, основные потребители пользовательского контента.
- Inactives – неактивны в социальных сервисах.

Еще одна классификация [25] пользователей социальных и других интернет-сервисов без научной теории в основе. Группы выделялись с помощью кластерного анализа результатов анкетирования жителей США и Европы. Разница между процентным объемом групп, выделенных в США и европейских странах, не превышает 7%.

Nielsen (2006)

- Lurkers (90%) – «бездельники», только читают, никогда не создают.
- Sporadic contributors (9%) – «случайные прохожие», читают и иногда создают.
- Active participants (1%) – «активные участники», вносят наибольший вклад в сообщество.

Эта описательная типология [7] предполагает, что три группы существуют в соотношении «90 – 9 – 1», т.е. процент активных участников наименьший. Правило выполняется на больших открытых сообществах и социальных сетях. В некоторых случаях доли смещены в сторону бездельников еще больше (в Wikipedia их 99,8%). Никаких особенных аналитических инструментов при разработке типологии не использовалось, однако в статье упоминается степенной закон в виде кривой Ципфа (Zipf curve), по которому распределена активность участников сообщества.

Jepsen (2006)

- Tourists – «туристы», не слишком задерживаются в сообществе и имеют поверхностный интерес к происходящему в нем.
- Minglers – общительные, не потребляют содержимое, предпочитают коммуникацию с другими членами сообщества.
- Devotees – «фанаты», наоборот, не общаются, но очень заинтересованы в материалах сообщества.
- Insiders – и общаются, и потребляют информацию.

Классификация [23] пользователей онлайн-сообществ (датских новостных групп) построена на основе теории сегментации пользователей виртуального сообщества (Kozinets, 1999) [28]. Участники относились к тому или иному сегменту по средним и медианным значениям опросов.

Golder and Donath (2004)

- Celebrities – центральные фигуры в сообществе, вносят наибольший вклад в него
- Newbies – новички, задают много вопросов, не разбираются в терминологии сообщества, не знают, как в нем принято общаться.
- Lurkers – пользователи, читающие группу, но не участвующие в обсуждениях.
- Flamers, Trolls, Ranters – флеймеры, тролли, флудеры – отличаются своим негативным поведением в сообществах.

В описательном исследовании [24] рассматривались 16 немодерируемых групп новостей (Usenet newsgroups), в которых шли обсуждения самых различных вопросов. По данным о частоте публикации сообщений и чтения групп была построена таксономия социальных ролей онлайн-сообществ.

Постановка задачи

Чтобы положить начало классификации пользователей коллаборативных платформ, мы планируем выполнить следующие задачи:

1. *С помощью математических инструментов проанализировать деятельность участников одного из проектов одной из платформ.* С подобного шага начинались многие исследования типов пользователей, описанные выше. Обычно анализируемые данные представляли собой результаты опроса участников или фактические показатели активности из базы.
2. *Выявить и описать классы пользователей.* На основе выполненного анализа литературы и деятельности участников требуется сформировать итоговую типологию.
3. *Сравнить доли групп с существующими типологиями.* Проведя аналогии между группами полученной типологии и классами других исследований, можно определить, отличается ли состав участников платформы от типичного для сообществ, а в перспективе – выяснить, хорошо ли это (например, путем подсчета индекса здоровья сообщества [29]).

Ход работы

Исходные данные

В качестве исходных данных для анализа были взяты результаты одного из проектов [30] коллаборативной платформы Witology. Из реляционной базы данных по этому проекту были извлечены количественные показатели активности участников – число созданных каждым участником текстов (комментариев, идей) и число поставленных каждым участником оценок. Также для вспомогательных целей были выгружены даты последних входов пользователей в систему.

В итоге анализируемые данные представляли собой таблицу со столбцами «Идентификатор пользователя», «ФИО пользователя», «Создал идей», «Написал комментариев», «Поставил оценок». В базе было зарегистрировано 519 человек, из которых 15 человек – администраторы и модераторы, остальные – приглашенные эксперты, т.е. участники, непосредственно занятые в генерации и обсуждении идей, и сторонние наблюдатели, идентифицировать которых сложно. Из анализа были ис-

ключены администраторы и модераторы, по сути, не являющиеся участниками проекта.

Специфика проекта

Для более полного восприятия дальнейшего анализа данных и типологии пользователей разумно привести описание работы платформы Witology в рамках проекта «Сбербанк-21».

Во-первых, следует отметить, что для того, чтобы стать участником проекта, желающим необходимо было подать заявку, пройдя тест; и лишь десятая часть (450 из 5198 человек) прошедших тест были приглашены к участию в проекте. Во-вторых, помимо приглашенных экспертов на платформе было 54 наблюдателя (представители компании-заказчика, Witology и др.), которых сложно выявить и исключить из анализа. В роль наблюдателей также вошли многие приглашенные эксперты.

Деятельность экспертов на разных этапах проекта включала следующие активности: генерация предложений идей, их обсуждение, фильтрация схожих предложений, доработка идей, генерация контр-идей, разного рода оценивание идей и комментариев (этапы общего голосования, ревю и пр.), игра на бирже идей, личная переписка с другими пользователями. Три наиболее понятных и важных показателя были взяты в качестве характеристик объектов для дальнейшего анализа.

Создание идей. В рамках любой из 15 задач, сгруппированных по 5 темам, эксперты могли генерировать идеи по решению задач. Существует разница между терминами «предложение идеи» и «идея». В описании исходных данных и далее нередко первый заменяется вторым, однако по факту идеями становятся не все предложения идей (в данном проекте только треть – 589 из 1581 предложений). Неочевидно, какой из этих показателей лучше в качестве характеристики участников, однако в данном исследовании мы учитывали число всех предложений по решению проблемы, поданных пользователем.

Комментирование. Каждый пользователь может свободно открыто комментировать любое сообщение другого пользователя. К таким сообщениям относятся организационные объявления администраторов, идеи, задачи, комментарии и пр. Учитывались все комментарии пользователей по любому поводу.

Оценивание. Так или иначе, участники оценивают идеи и комментарии на разных этапах. Мы рассматривали наиболее доступный всем участникам способ – выставление пары оценок «качество» - «отношение» по шкале от -3 до 3 идее или комментарию. Характеристикой участника выступало число поставленных им пар оценок, число же полученных им оценок – скорее, показатель осведомленности или популяр-

ности, но не активности участника – в перспективе можно также учитывать при выделении классов участников.

На основе этих и других счетчиков действий участников вычисляются значения рейтингов участников. Восемь существующих рейтингов («Судья», «Игрок», «Болельщик», «Деятель» и четыре рейтинга, измеряющих влияние, значимость, популярность и репутацию) в какой-то степени относят пользователей платформы к одному или нескольким типам. Тем не менее, рейтинговая система никак не учитывается в данной работе, поскольку видится нам немного субъективной и спорной.

Выделение сегментов

Сначала среди 504 участников были найдены те, кто не написал ни одного сообщения, не создал ни одной идеи и не поставил ни одной оценки. Эти 248 человек явно относятся к группе неактивных и незаинтересованных пользователей (165 из них заходили на платформу в последний раз в первые три дня ее работы) и не нуждаются в кластерном анализе.

Далее для деления пользователей по нескольким параметрам было решено воспользоваться кластеризацией k-means [31]. Выбор кластерного анализа обусловлен доказанной эффективностью и популярностью данного метода в задаче выделения групп схожих объектов. В широчайшем обзоре [6] ровно в половине рассмотренных исследований применялась кластеризация, там же утверждается, что такая статистика не случайна – кластеризация, действительно, удобнее, чем, например, второй по популярности (5 из 22 исследований) метод – факторный анализ – для выполнения такого рода задачи. Одним из аргументов в пользу кластерного анализа был тот факт, что он оперирует реальными переменными, а не абстрактными факторами, что упрощает интерпретацию результатов. Алгоритму k-means было отдано предпочтение из-за его универсальности и распространенности для базового, необязательно очень точного (размер выборки позволяет скорректировать результаты вручную) деления объектов на группы.

Результаты кластеризации 256 объектов изображены на рис. 1 (для кластерного анализа использовалась надстройка XLSTAT 2011 [32] над MS Excel, для трехмерной визуализации – ее пакет XLSTAT-3DPlot).

Первый кластер (кольца) – участники, не проявившие особой активности в комментировании и оценке. Из-за разницы в порядках числа созданных идей, комментариев и оценок участники, сгенерировавшие много (более 10) идей также оказались в этой группе. (Кластеризация нормированных показателей избавляет результаты от этого недостатка, однако сильно путает все остальные кластеры, поэтому здесь приводятся результаты первичной наивной кластеризации.)

Второй кластер (круги) отличается от первого немного большей активностью участников как оценщиков. Вероятно, у таких людей интерес к проекту все же присутствовал, и за развитием событий они следили, однако на деятельное преобразование мира (комментирование и создание идей) большинство не было достаточно мотивировано. Часть этого кластера целесообразно объединить с предыдущим.

Третий кластер (треугольники) целиком сложно охарактеризовать. Его представители не пассивны, они порой оставляют комментарии и иногда создают идеи, но всегда ставят оценки. Только пять активных генераторов идей слева мешают с определенной долей уверенности считать их типичными критиками [25].

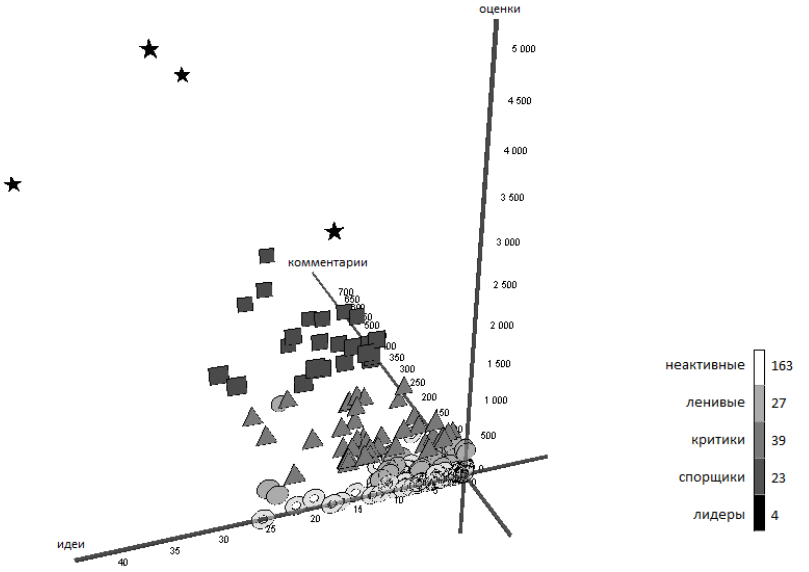


Рис. 3. Кластеризация пользователей (справа от цветовой шкалы – число объектов в кластере)

Четвертый кластер (квадраты) немного отрывается от предыдущего по числу оценок и сильно отличается общительностью (числом комментариев). Кластер также следует разделить на активно создающих и остальных пользователей. Без двух создателей напрашивается сравнение группы со спорщиками [5], связными/продавцами [13], или же с селебрити (звездами) [24].

Последний, самый немногочисленный кластер (звезды) – безусловные лидеры, они и несколько человек из предыдущей группы оказались

авторами победивших идей проекта. Эту группу можно сопоставить знатокам [13] или, опять же, селебрити (звездам) [24].

Для большей правдивости классификации полученные кластеры были модифицированы: часть колец, кругов, треугольников и квадратов образовали класс создателей, а оставшиеся круги примкнули к бездельникам; также были сделаны некоторые мелкие перестановки (рис. 2 дает примерное представление о характере модификаций). Из всех пяти только класс создателей неоднороден – кольца, два круга и треугольник отличаются от остальных практически нулевыми показателями комментирования и невысокой активностью по оценке идей и комментариев. Здесь мы не будем выделять общительных и необщительных создателей в отдельные классы из-за недостаточного доказательства коренных различий в этих подтипах и малой выборки.

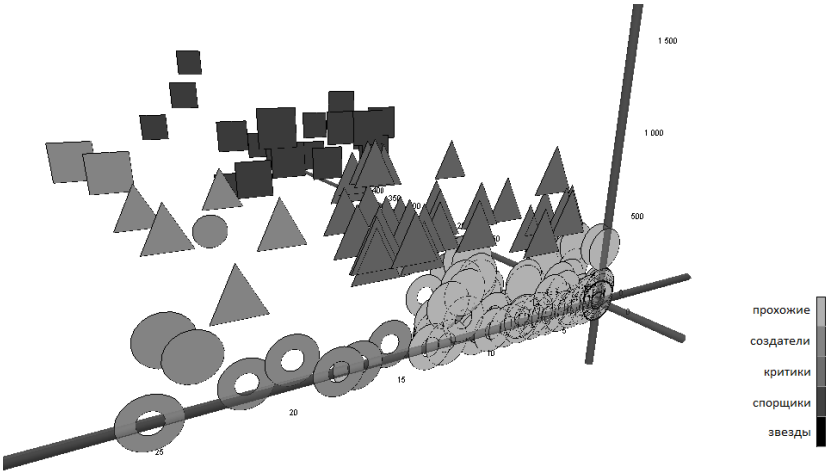


Рис. 4. Модифицированные кластеры

Результаты

В таблице 1 приведены итоговые классы пользователей, их доли, описания и соответствия классам из других типологий. Названия выбирались соответственно характеру группы из встреченных в рассмотренных типологиях.

Итак, как это обычно случается [25], половина пользователей сообщества не проявляет совершенно никакой активности. Последние две группы логично объединить в одну, соответствующую бездельникам (lurkers) типологии Нильсена [7], создателей и критиков вместе можно

сопоставить второй группе (sporadic contributors), а спорщиков и звезд – третьей группе самых активных.

Табл. 1. Классы участников колаборативной платформы Witology

Название	**Число/ доля объектов**		**Описание**	**Соответствие классам других типологий**
Звезды	4	1%	Выдающиеся пользователи, рекордсмены по каким-либо активностям.	Actives [5], mavens [13], active participants [7], insiders [23], celebrities [24]
Спорщики	21	4%	Пользователи, которые пишут очень много комментариев и активно ставят оценки.	Debators/socializers [5], connectors/salesmen [13], active participants [7], minglers [23]
Создатели	20	4%	Участники-генераторы идей. Делятся на энергичных создателей (6 человек), у которых хватает времени и на другие активности, и социопатичных создателей (14 человек), в обсуждениях не участвующих.	Mavens [13], creators [25], active participants/sporadic contributors [7], insiders/devotees [23]
Критики	34	7%	Пользователи, склонные ставить оценки не влезая в дискуссии.	Critics/spectators [25], sporadic contributors [7], lurkers [24]
Прохожие	177	35%	Те, кто делали попытки проявить активность, но редко и без энтузиазма.	Sporadics/lurkers [5], spectators [25], lurkers [7], tourists [23], newbies/lurkers [24]
Неактивные	248	49%	Те, кто ничего не делали.	Sporadics/lurkers [5], inactives [25], lurkers [7], tourists [23]

Подобное сравнение с наблюдениями Нильсена также провел Brandtzæg [5]: к первой группе он отнес sporadics и lurkers, ко второй – debaters и socializers, а к третьей – actives. Соответствие долей классификаций пользователей онлайн-сообществ приведено в табл. 2. Сравнение с другими типологиями не проводилось из-за заметных различий в трактовке классов, а также в силу отсутствия данных по долям каждой группы в некоторых исследованиях.

Табл. 2. Сравнение процентных долей разных типологий

Класс	**Типология Нильсена**	**Данная типология**	**Типология Brandtzæg**
Lurkers	90%	84%	46%
Sporadic contributors	9%	11%	36%
Active participants	1%	5%	18%

Интересно, что процентное соотношение классов полученной типологии сильно ближе к типологии Нильсена. Сам Brandtzæg объясняет несоответствие своих результатов правилу «90 – 9 – 1» тем, что норвежские социальные сети не так распространены, как, например, YouTube и Wikipedia, а также имеют меньшие барьеры для создания своего содержимого. То же можно сказать и о рассматриваемом проекте платформы Witology, и, более того, его участники проходили специальный отбор и, вероятно, обладают большей мотивацией на работу в проекте, чем посетители Wikipedia на написание статей.

Почему же тогда так много бездельников? Можно предположить, что наблюдатели (54 человека), большая часть которых наверняка попала в группу lurkers, как-то портят статистику, однако без них процент бездельников всего на единицу меньше. Другими объяснениями молчания 84% экспертов могут быть их недостаточная компетентность и осведомленность в проблемах Сбербанка, разочаровавший их формат проведения проекта (как и сама платформа, так и работа модераторов и администраторов), изначально низкий интерес к непосредственной работе на проекте и прохождение теста только из любопытства и т.д. Определить причины точнее без сравнения данных по разным проектам или даже разным платформам пока не представляется возможным.

Заключение

В ходе поиска литературы по теме обнаружилось, что на данный момент не существует научной общепринятой классификации пользователей социальных сетевых сервисов, что, скорее всего, обусловлено широким их разнообразием. Однако поиск типологий членов сообществ

рассматриваемого вида также ничего пока не дал. На основе анализа рассмотренных классификаций пользователей социальных сетей, групп новостей, блогов и пр. с помощью кластерного анализа показателей активности на платформе инноваций Witology удалось построить собственную классификацию участников. Из-за различий в устройстве платформ краудсорсинга нельзя утверждать, что полученная классификация будет верна для, например, рынков предсказаний [33] или сервиса по дизайну роботов [34]. Доли классов примерно следуют известному правилу «90 – 9 – 1», по которому большая часть пользователей сервисов только наблюдает, малый процент изредка что-то создает, и лишь единицы действительно активны. Такой результат хорош для больших сообществ, однако не совсем ясно, почему в краудсорсинговом проекте с участием по приглашениям наблюдателей так много.

Таким образом, поставленные задачи были по большей части выполнены.

Перспективы исследования

Полученные классы пользователей (неактивные, прохожие, критики, создатели, спорщики, звезды) далеко не окончательны по многим причинам. Анализировались данные лишь по одному из проектов площадки, а сравнение результатов по нескольким проектам существенно бы уточнило классификацию. Результаты кластерного анализа также можно сделать точнее, введя веса параметров, отнормировав показатели или меняя алгоритм кластеризации. В частности, судя по тому, что разброс размеров полученных классов велик, k-means не подходит для решения рассматриваемой задачи, так как отличается тенденцией выделять равноразмерные кластеры, в отличие от, например, EM-алгоритма. Еще одним аргументом в пользу переосмысления инструментария является то, что k-means, вообще говоря, рекомендован для выборок с числом объектов более 1000 [35], а проект на платформе Witology с таким числом участником пока не представляется возможным найти.

К перспективам исследования также относятся следующие задачи.

- Вовлечение других данных по проекту в анализ. Можно было бы учитывать, сколько раз (дней, часов, сессий) пользователь был на платформе за весь проект, или даже анализировать логи его пребывания на сайте, чтобы явно увидеть наблюдателей, бездельников и безучастных. Также интересны характер оценок, которые ставит участник, количество просмотренных идей на этапе отбора схожих или другие активности, которые появятся на платформе в будущем.
- Разработка новой классификации другими методами. Для выделения групп также подходят факторный анализ [15], всесторон-

ний качественный анализ [36], изучение статистических показателей, анализ средних [23], построение и анализ графов [13], [14], [37] и др.

- Поиск особенных пользователей (троллей, флеймеров, флудеров [24]).
- Построение алгоритма отнесения (классификации) новых объектов к существующим классам.
- Проверка связи между группами и демографическими факторами (возраст, пол).
- Дальнейший поиск литературы.
- Определение здоровья сообщества [29].

По числу приведенных возможных направлений развития работы можно заключить, что данная статья – лишь небольшая пробная вылазка в исследование поведения пользователей в рамках проектов коллаборативных платформ, описывающая только срез одного из проектов и не претендующая на неоспоримость и фундаментальность.

Выводы

На данный момент нет исследований, посвященной типологиям участников краудсорсинговых проектов.

С помощью кластерного анализа данных краудсорсингового проекта «Сбербанк-21» удалось выделить следующие группы пользователей: «звезды», «спорщики», «создатели», «критики», «прохожие», «неактивные».

Доля «прохожих» и «неактивных» участников – 84% – заметно больше ожидаемой. Активность в проекте примерно распределена по правилу «90 – 9 – 1», типичному для глобальных сообществ и сетей.

Результаты требуют доработки и уточнения. Им может поспособствовать использование другого алгоритма кластерного анализа, увеличение выборки и числа признаков, сравнение данных по разным краудсорсинговым площадкам и пр.

Список источников

1. http://wiki.witology.com/index.php/Рынки_предсказаний – Рынки предсказаний. – 2011.

2. http://www.searchenginejournal.com/the-growth-of-social-media-an-infographic/32788/ – The Growth of Social Media: An Infographic. – 2011.

3. Kelsey, T. Social Networking Spaces: From Facebook to Twitter and Everything In Between. – Springer-Verlag, 2010.

4. Boyd, D. M., and Ellison, N. B. Social Network Sites: Definition, History, and Scholarship // Journal of Computer-Mediated Communication, 13 (1). – 2007. (http://jcmc.indiana.edu/vol13/issue1/boyd.ellison.html)

5. Brandtzæg, P. B. and Heim, J. A Typology of Social Networking Sites Users // International Journal of Web Based Communities 2011, 7 (1). – 2010. (http://www.inderscience.com/storage/f916113101287452.pdf)

6. Brandtzæg, P. B. Towards a unified media-user typology (MUT): a meta-analysis and review of the research literature on media-user typologies // Computers in Human Behaviour, 26 (5). – 2010. (http://sintef.academia.edu/PetterBaeBrandtz%C3%A6g/Papers/814028/Towards_a_unified_Media-User_Typology_MUT_A_meta-analysis_and_review_of_the_research_literature_on_media-user_typologies)

7. Nielsen, J. Participation Inequality: Encouraging More Users to Contribute. // Jakob Nielsen's Alertbox, 9 October 2006. – 2006. (http://www.useit.com/alertbox/participation_inequality.html)

8. http://lithosphere.lithium.com/t5/Building-Community-the-Platform/Community-vs-Social-Network/ba-p/5283 – Community vs. Social Network. – 2010.

9. http://wiki.witology.com/index.php/Краудсорсинг – Краудсорсинг. – 2011.

10. http://witology.com/

11. http://www.openinnovators.net/list-open-innovation-crowdsourcing-examples/ – Open Innovation Crowdsourcing Examples.

12. http://www.inventech.ru/lib/analis/analis0013/ – Методы классификации и типологии.

13. Budak, C., Agrawal, D., Abbadi, A. E. Where the Blogs Tip: Connectors, Mavens, Salesmen and Translators of the Blogosphere // In Proc. of Workshop on Social Media Analytics (SOMA 2010), New York, USA. – 2010. (http://snap.stanford.edu/soma2010/papers/soma2010_15.pdf)

14. Chan, J., Hayes, C., Daly, E. M. Decomposing Discussion Forums and Boards Using User Roles // In Proc. of Web Science Conf. (WebSci10), Raleigh, USA. – 2010. (http://www.deri.ie/fileadmin/documents/uimr/jkcchan_icwsm10.pdf)

15. Johnson, G. M., and Kulpa, A. Dimensions of online behavior: Toward a user typology // CyberPsychology & Behavior, 10 (6). – 2007.

16. Heim, J., Brandtzæg, P. B., Endestad, T., Kaare, B. H., and Torgersen, L. Children's usage of media technologies and psychosocial factors // New Media & Society, 9(3). – 2007.

17. Livingstone, S., and Helsper, E. Gradations in digital inclusion: Children, young people and the digital divide // New Media & Society, 9(4). – 2007.

18. Barnes, S. J., Bauer, H., Neumann, M., and Huber, F. Segmenting cyberspace: A customer typology for the Internet // European Journal of Marketing, 41(1). – 2007.

19. Selwyn, N., Gorard, S., and Furlong, J. Whose Internet is it anyway? Exploring adults (non)use of the internet in everyday life // European Journal of Communication, 20(1). – 2005.

20. Heim, J., and Brandtzæg, P. B. Patterns of Media Usage and the Non-Professional Users // In Proc. of the SIGCHI Conference on Human factors in computing systems (CHI 2007), San Jose, California, USA. – 2007.

21. Horrigan, J. B. A Typology of Information and Communication Technology Users // Pew Internet Report, USA. – 2007. (http://pewinternet.org/~/media/Files/Reports/2007/PIP_ICT_Typology.pdf.pdf)

22. Angeletou, S., Rowe, M., Alani, H. Modelling and Analysis of User Behaviour in Online Communities // In Proc. of International Semantic Web Conf. (ISWC 2011), Bonn, Germany. – 2011. (http://people.kmi.open.ac.uk/rowe/files/mrowe-iswc2011.pdf)

23. Jepsen, A. L. Information Search in Virtual Communities: Is It Replacing Use of Off-Line Communication? // Journal of Marketing Communications, 12 (4). – 2006.

24. Golder, S. A., and Donath, J. Social Roles in Electronic Communities // In Proc. of Association of Internet Researchers (AoIR) Conference 5.0, Brighton, UK. – 2004. (http://web.media.mit.edu/~golder/projects/roles/golder2004.pdf)

25. Li, Ch., Bernoff, J., Fiorentino, R., and Glass, S. Mapping Participation In Activities Forms The Foundation Of A Social Strategy // A Forrester Research Report, New York. – 2007. (http://www.icsd.aegean.gr/website_files/proptyxiako/277846938.pdf)

26. Gladwell, M. The Tipping Point: How Little Things Can Make a Big Difference. — Back Bay Books, 2002.

27. http://www.memetracker.org/

28. Kozinets, R. V. E-Tribalized Marketing? The Strategic Implications of Virtual Communities of Consumption // European Management Journal, 17 (3). – 1999. (http://kozinets.net/__oneclick_uploads/2008/06/etribalized_marketing_emj.pdf)

29. Measuring Community Health for Online Communities. Community Health Index White Paper // Lithium. – 2011. (http://pages.lithium.com/community-health-index.html)

30. http://sberbank21.ru/

31. http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html – K-Means Clustering.

32. http://www.xlstat.com/en/

33. http://www.intrade.com

34. http://ldd.lego.com/en-us/subpages/mindstorms/ – Lego Digital Designer.

35. Tan, P., Steinbach, M., and Kumar, V. Introduction to Data Mining. – Pearson Addison-Wesley, 2006.

36. Social networking. A quantitative and qualitative research report into attitudes, behaviours and use // Office of Communication (OFCOM), London. – 2008.

37. Zhang, J., Ackerman, M. S., and Adamic, L. Expertise Networks in Online Communities, Structure and Algorithms // In Proc. of the 16th International Conf. on World Wide Web (WWW 2007), New York, USA. – 2007. (http://www2007.org/papers/paper516.pdf)

Выявление пересекающихся сообществ в социальных сетях

Н. Бузун, А. Коршунов

Институт системного программирования РАН, Москва, Россия

Аннотация. Кластерная структура является одной из главных особенностей социальных графов. Несмотря на большое количество алгоритмов ее выявления, существует необходимость определения области их эффективной применимости при различных значениях конфигурационных параметров сети. В этой статье основное внимание уделено степени пересечения кластеров. Выполнено тестирование как наиболее современных методов нечеткой кластеризации, так и обобщенных классических подходов. В зависимости от величины пересечения сделан вывод о применимости отдельных классов алгоритмов с общей методикой и их представителей.

Ключевые слова: кластеризация; социальные сети; выявление сообществ; community detection.

Введение

Сети являются естественным представлением различных сложных систем в обществе, биологии, технике и других областях. Множество сетей характеризуются мезоскопическим уровнем организации внутри групп узлов, образующих единицы с большим количеством связей. Такие единицы называются *кластерами* (*сообществами* или *модулями*).

В последние годы внимание данной области исследований сфокусировано на социальных и естественных сетях, для обнаружения внутрен-

Игнатов Д. И., Яворский Р. Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16–18 марта, 2012

ней структуры которых не применимы классические алгоритмы кластеризации.

Можно привести несколько вариантов полезной информации, полученной на основании разбиения сети на сообщества: обнаружение функциональных единиц системы, выявление сходства между вершинами из одного сообщества, вершины в одном сообществе могут классифицироваться в соответствии с их позицией (лидеры, связывающие и т. д.), удобный способ визуализации системы, определение атрибутов вершин на основании общих атрибутов сообществ, включающих данные вершины [1].

Рассмотрим некоторые особенности структуры социальных сетей, которые требуется учитывать при выборе алгоритма кластеризации.

1) Вершина может находиться одновременно в нескольких сообществах с разной степенью принадлежности (`fuzzy clusters`) [2–10,24]

2) Сообщества могут иметь *иерархическую структуру* [4, 6, 8, 11, 22, 24], что требуется для эффективного управления в масштабных организациях, а наличие таковой подчеркивает стабильность системы [12]

3) Помимо того, высокая плотность ребер не всегда свидетельствует о наличии кластера. Поэтому для отсечения "псевдосообществ" вычисляется вероятность реализации конкретной конфигурации подграфа ("*статистическая значимость*") в предположении истинности гипотезы случайного распределения ребер (при заданных значениях степеней вершин) [5,9,13].

4) В некоторых случаях (например при определении атрибутов вершин) необходимо присвоить вершинам и ребрам *несколько параметров* [1, 2, 14]. При этом в большинстве своем в настоящее время алгоритмы принимают на вход лишь 1 параметр — веса ребер.

5) Также может дополнительно ставиться задача изучения *динамики сообществ* в сети [15].

В данной статье внимание акцентируется на выявлении пересекающихся сообществ в сетях больших размеров ($n = 10^8,\ m \sim n$) с высоким коэффициентом пересечения ($r \sim 10$). P — множество сообществ, $G = (E, V),\ m = |E|,\ n = |V|,\ r = \sum |P_i|/n$. Приводится несколько современных алгоритмов, которым изначально присуща возможность выявления указанных сообществ. Помимо того, предлагается несколько вариантов обобщения классических алгоритмов на случай графов с пересекающимися кластерами. Целью же данного исследования является

выявление наиболее релевантных методов нечеткой (пересекающейся) кластеризации и способов оценки качества разбиения графа.

Обзор методов выявления сообществ

Модель случайного графа (null model). В методах данного класса заданная конфигурация ребер сравнивается с равномерным их распределением для каждой вершины графа. При этом степени вершин случайного графа в большинстве случаев считаются известными параметрами. Классическим вариантом здесь является максимизация целевой функции `modularity` и ее модификаций [16–21], характеризующей суммарную разность количества ребер в сообществе и их математического ожидания:

$$Q = \frac{1}{2m} \sum_{c \in P} \sum_{i,j \in c} [A_{ij} - Pr(A_{ij} = 1)],$$

где P — множество сообществ, A — матрица инцидентности.

Аналогично вместо ребер можно брать во внимание количество треугольников в сообществе, считая связи между вершинами слабыми, если они не являются ребрами треугольника [8,20].

Изначально `modularity` вводилась для характеристики непересекающихся разбиений, но существуют ее обобщения и на случай *пересекающихся сообществ* [17,21]. Помимо того, стоит упомянуть ее квантовомеханическую модификацию [18,19], позволяющую улучшить *разрешающий предел* и придающую ей энергетический смысл системы вершин с различными спиновыми значениями (`spinglass` [19]).

Более общим методом является обнаружение “значимых” кластеров, имеющих малую вероятность конфигурации в предположении истинности гипотезы случайного графа. (`oslom, moses`) [5,9]

Блуждания (random walk).

`infomap`[10,22]: В данном случае граф разбивается таким образом, чтобы минимизировать длину описания случайного блуждания в данном графе. Одной из оценочных функций для ожидаемой длины кода является энтропия, широко используемая в различных разделах теории информации. Исходя из этого в [22] предлагается в качестве оценочной рассматривать следующую функцию:

$$L(P) = qH(Q) + \sum_i p_i H(P_i),$$

где q — вероятность перехода в другой модуль, p_i — доля переходов внутри i-го модуля, $H(Q)$ — энтропия названий модулей, $H(P_i)$ — энтропия названий вершин внутри модуля.

`walktrap` [23]: Здесь формирование сообществ происходит на основе следующего утверждения: Пусть вершины i, j принадлежат одному кластеру, тогда $Pr(k|i,t) \approx Pr(k|j,t)$ для всех $k \in V$, где Pr — матрица перехода.

`betweenness` [9]: Введение меры "промежуточности" на множестве ребер (чем больше проходов по ребру при случайном блуждании, тем больше величина меры). Ребра с большой "промежуточностью" естественно считать внекластерными `(conga, GN)` [3].

Локальный анализ подграфов. При локальном изучении и формировании кластера (без учета структуры остальной части графа) обычно рассматривается отношение количества внутренних ребер и треугольников к внешнему и максимально возможному их числу (`cohesion` [8]). Также сообщества могут формироваться на основании схожести с полным графом или набором связанных клик различного размера `(CFinder, GCE)` [7]. Помимо того, для локальной характеристики похожести подграфа на сообщество может быть использована упомянутая выше *"статистическая значимость"*. Существует также набор методов из данного раздела, позволяющих независимо выделять подграфы с высокой величиной влияния вершин внутри себя (`moduland` [6]). Для данного класса методов свойственно естественное выделение пересекающихся сообществ, но возникают трудности с последующим формированием конечного разбиения всего графа.

Введение координат. Еще одним изящным подходом является присвоение координат вершинам в графе [26], которыми являются компоненты собственных векторов нормированной матрицы Лапласа L. Данный способ кластеризации является крайне полезным, если требуется использовать уже известные атрибуты вершин.

Подытоживая обзор, можно выделить методы `spinglass`, `infomap`, `wolktrap` обладающие наиболее высокими показателями *Normalized Mutual Information* [24] (для случая *непересекающихся* сообществ) при относительно небольшом времени работы и возможностью параллельного исполнения [25].

Способы обобщения на случай пересекающихся сообществ

Статическое. Используя меру `betweenness` на множестве вершин, можно каждую вершину с высоким значением меры разделить на две, соединенные ребром (`toolteep` [3]). Альтернативный вариант — генерация линейных графов(ребра переходят в вершины, а вершины в ноль или несколько ребер) и последующая кластеризация ребер [4].

Динамическое. Вводя коэффициенты принадлежности для вершин (вероятности нахождения в каждом из сообществ), в процессе работы алгоритма относят вершину одновременно к нескольким кластерам. Как ориентир для коэффициента принадлежности могут быть использованы следующие величины:

Индивидуальный вклад в прирост целевой функции:

$$Pr(V_i \in P_k) \sim Q(V_i \in P_k) - Q(P_k \backslash V_i) = \triangle Q_{ik}.$$

Вероятность нахождения на определенном энергетическом уровне:

$$Pr(V_i \in P_k) = e^{-\beta Q(V_i \in P_k)} / \sum_S e^{-\beta Q(V_i \in P_S)},$$

где Q — целевая функция, β — величина, обратно пропорциональная коэффициенту пересечения.

Интересно заметить, что введение коэффициентов принадлежности часто улучшает разбиение на непересекающиеся сообщества. Основная идея здесь в том, что задавая вероятностей переходов вершины в другие сообщества (оставаясь с определенной вероятностью в исходном) мы "сообщаем" другим вершинам тактику ее поведения. Т. о. для задания коэффициентов в этом случае может быть использовано следующее выражение:

$$Pr(V_i \in P_k) \sim \triangle Q_{ik} - \min_h(\triangle Q_{ih}),\ V_i \in P_h\ Pr(V_i \in P_{hmax}) \sim 0{,}1.$$

Способы реализации

1) Алгоритм жадной оптимизации целевой функции (используется большинством из упомянутых выше алгоритмов): Изначально каждая вершина является сообществом. Далее на каждом шаге каждая вершина выбирает к каким сообществам присоединиться, сравнивая величины прироста целевой функции. Завершающей стадией является объединение сформировавшихся модулей, с большим числом связей.
2) Метод центральных вершин [2]: Задается несколько центральных вершин, к которым постепенно присоединяются остальные, выбирая наиболее "близкий" кластер.
3) Рекурсивное разбиение исходного графа на две и более частей [16]. Вначале вершины разбиваются случайным образом. Затем перемещаются *в первую очередь* те, которые дают максимальный прирост целевой функции.

4) В случае применения локальной оптимизации предлагается следующая схема [9]:

Однокластерный анализ $\mapsto$ Проверка внутренней структуры $\mapsto$
$\mapsto$ Объединение кластеров $\mapsto$ Вычисление коэфф. принадлежности.

Тестирование

Здесь в качестве генератора сетей с пересекающейся кластерной структурой используется `LFR benchmark` алгоритм [9]. С целью исследования работы алгоритмов при различной величине пересечения сообществ были сгенерированы два множества тестовых графов с предопределенным разбиением. В качестве параметров указанному генерарору передавались следующие величины: n — число вершин, k — срернее значение степени вершины, $kmax$ — максимальное значение степени вершины, $|P_i|$ — количество вершин в кластере, τ_1 — значение экспоненты степенного распределения степени вершин, τ_2 — значение экспоненты степенного распределения $|P_i|$, μ — усредненная нормированная степень вершины внутри родительского сообщества, on — число вершин, принадлежащих более чем одному сообществу, om — количество сообществ, содержащих фиксированную вершину. Параметры графов из первого множества отличаются значением om, из второго — значением on.

Для сравнения полученных разными методами разбиений (рис. 1: 1, 2) будем использовать меру *Normalized Mutual Information* (I_{norm}) [24], базирующуюся на следующем предложении: если два разбиения графа похожи, то требуется относительно небольшое количество информации для получения первого разбиения при известном втором.

$$I(X,Y) = H(X) - H(X|Y)$$

$$I_{norm}(X,Y) = \frac{2I(X,Y)}{H(X)+H(Y)},$$

где H — энтропия Шеннона.

В дополнение, для разбиений графов из первого множества (рис. 1: 3) будем вычислять обобщенную на случай нечеткой кластеризации функцию `modularity`.

По результатам экспериментов выявления сообществ с разной величиной пересечения можно заключить, что для случая *значительного пересечения* может быть эффективно применено лишь несколько методов — `oslom` [9], `moses` [5], `gce` [7], которые представляют класс локальной оптимизации. В частности первые два свидетельствуют об эффективности использования “статистической значимости” в качестве индивидуальной (локальной) характеристики выраженной кластерной структуры.

Также стоит отметить результативность подхода пересекающейся кластеризации ребер [4], который может быть применен для сетей средних и малых размеров. Для больших сетей с незначительным пересечением могут быть использованы методы, имеющие сложность не более $O(n^{\alpha})$ $\alpha \in [1,2]$ — `fuzzy infomap` [10], `gce` [7], `fuzzy spinglass` [18].

Помимо того, анализируя графики значений `modularity` (рис. 1: 3), следует подчеркнуть расхождение в оценке качества разбиения с I_{norm} при увеличении on. Откуда следует, что `modularity` дает объективную оценку разбиения только при небольших значениях коэффициента пересечения r.

Заключение

В итоге, в данном исследовании было указано несколько основных свойств социальных и естественных графов, проведено разбиение алгоритмов на четыре класса. Также предложено несколько различных типов их обобщения на случай пересекающихся сообществ и приведены основные варианты их реализации. Из результатов тестирования на искусственно сгенерированных сетях выявлена применимость наиболее современных методов при различных конфигурациях графа.

Возможными направлениями дальнейшей работы является продолжение изучения слабых и сильных сторон приведенных классов алгоритмов в зависимости от свойств графа и поставленных целей. При этом во внимание будут приниматься все отмеченные в начале статьи особенности социальных графов. В частности, довольно значимыми являются задача выявления иерархической структуры, методов ее оценки, а также кластеризация графов с атрибутами (`ordered graphs` [14]) на множестве вершин и ребер — что является первостепенной задачей для предсказания неизвестных атрибутов. Вследствие аккумуляцией результатов проведенных исследований может быть обучающийся анализатор графов, определяющий на каких частях графа (отличающихся, например, величиной пересечения сообществ) может быть эффективно применен конкретный метод.

Список источников

1. Tang L., 2010. Learning with Large-Scale Social Media Networks. Ph. D. Dissertation. Arizona State University, Tempe, AZ, USA. Advisor: Huan Liu. AAI3425805.
2. Zhang S., Wang R. S., Zhang X. S., 2007. Identification of overlapping community structure in complex networks using fuzzy c-means clustering. Physica A 374: 483–490.

3. Gregory S., 2007. An algorithm to find overlapping community structure in networks. Berlin, Germany: Springer-Verlag. Pp 91–102. https://www.cs.bris.ac.uk/∼steve
4. Ahn Y., Bagrow J. P., Lehmann S. 2010. Link communities reveal multiscale complexity in networks. Nature 466, 761–764.
5. McDaid A. F,, Hurley N. J. 2010. Using Model-based Overlapping Seed Expansion to detect highly overlapping community structure. In: ASONAM 2010. http://sites.google.com/site/aaronmcdaid/moses
6. Kovacs I. A., Palotai R., Szalay M. S., Csermely P. 2010. Community landscapes: An integrative approach to determine overlapping network module hierarchy, identify key nodes and predict network dynamics. PLoS ONE 5: e12528
7. Lee C., Reid F., McDaid A., Hurley N. 2010. Detecting highly overlapping community structure by greedy clique expansion. Poster at KDD 2010.
8. Friggeri A., Chelius G., and Fleury E. 2011. Egomunities, Exploring Socially Cohesive Person-based Communities. NRIA, Research Report RR-7535, 02 2011
9. Lancichinetti A., Radicchi F., Ramasco J., Fortunato S. 2011. Finding Statistically Significant Communities in Networks. PLoS ONE 6(4): e18961. http://santo.fortunato.googlepages.com/inthepress2
10. Esquivel A. V., Rosvall M. 2011. Compression of flow can reveal overlapping modular organization in networks. Phys. Rev. X 1, 021025 (2011). https://sites.google.com/site/alcidesve82
11. Clauset A., Moore C., Newman M. E. J. 2008. Hierarchical structure and the prediction of missing links in networks. Nature 453: 98–101.
12. Simon H. 1962. The architecture of complexity. Proc. Am. Phil. Soc. 106: 467–482.
13. Lancichinetti A., Radicchi F., Ramasco J. J. 2010. Statistical significance of communities in networks. Phys. Rev. E 81: 046110
14. Gregory S. 2011. Ordered community structure in networks. Physica A: Statistical Mechanics and its Applications (December 2011)
15. Mucha P. J., Richardson T., Macon K., Porter M. A., Onnela J. 2010. Community Structure in Time-Dependent, Multiscale, and Multiplex Networks. Science 328: 876.
16. Newman M. E. J. 2006. Finding community structure in networks using the eigenvectors of matrices. Phys. Rev. E 74, 036104 (2006)

17. Nicosia V., Mangioni G., Carchiolo V., Malgeri M. 2008. Extending modularity definition for directed graphs with overlapping communities. J. Stat. Mech. P03024 (2009).
18. Reichardt J., Bornholdt S. 2008. Statistical Mechanics of Community Detection. Phys. Rev. E 74 (1) (2006) 016110
19. Ronhovde P., Nussinov Z. 2009. Multiresolution community detection for megascale networks by information-based replica correlations. Phys. Rev. E 80 (1) (2009) 016109
20. Arenas A., Fernandez A., Fortunato S. 2008. Gomez. Motif-based communities in complex networks. J. Phys. A 41 (22) (2008) 224001.
21. Lazar A., Abel D., Vicsek T. 2009. Modularity Measure of Networks With Overlapping Modules. IOP Publishing, Pages: 18001
22. Rosvall M., Bergstrom C. T. 2008. Maps of random walks on complex networks reveal community structure. Proc Natl Acad Sci USA 105: 1118–1123.
http://www.tp.umu.se/~rosvall/code.html
23. Pons P., Latapy M. 2005. Computing communities in large networks using random walks. Sci. 3733 (2005) 284–293.
24. Lancichinetti A., Fortunato A., Kertesz J. Detecting the overlapping and hierarchical community structure in complex networks. New J. Phys. 11, 033015, 2009
25. Fortunato S. Community detection in graphs. 2009. Physics Reports, 486, 75–174
26. Donetti L., Mucoz M. A. Detecting network communities: a new systematic and efficient algorithm. 2004. J. Stat. Mech. P10012 (2004).

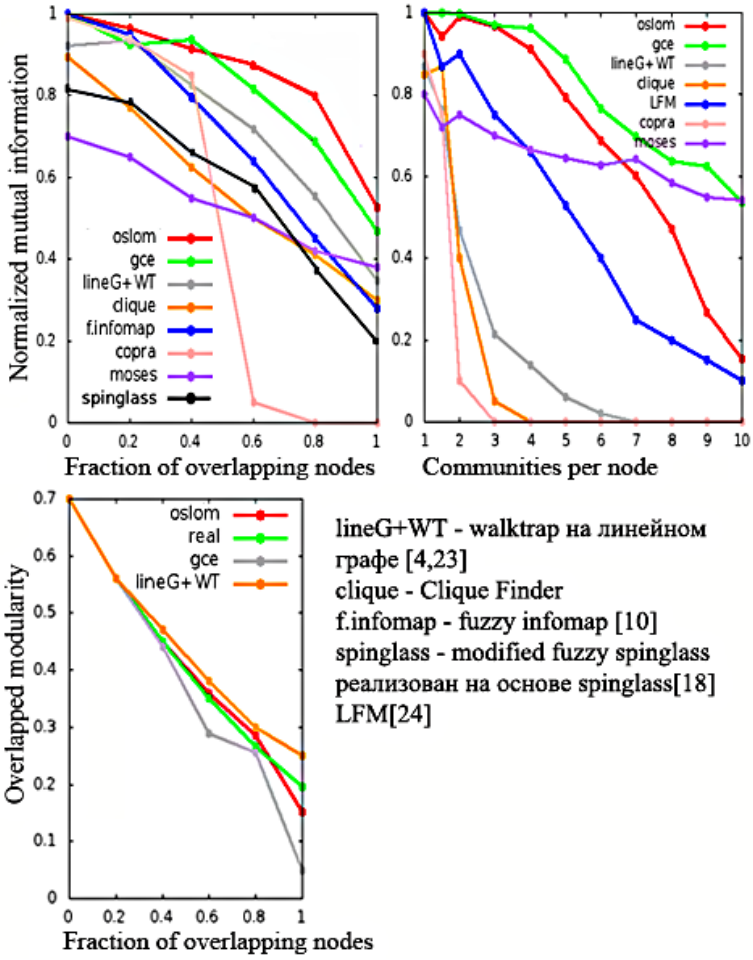


Рис. 1. Тестирование алгоритмов нечеткой кластеризации
1, 3: $n = 2000$, $k = 15om$, $kmax = 45om$, $|P_i| \in [15, 60]$, $\tau_1 = 2$, $\tau_2 = 0$, $\mu = 0{,}2$, $on \in \{0, 1000, 2000\}$, $om \in \{1, 1.5, 2, 3, ..., 9, 10\}$
2: $n = 1000$, $k = 20$, $kmax = 50$, $|P_i| \in [20, 100]$, $\tau_1 = 2$, $\tau_2 = 1$, $\mu = 0{,}3$, $on \in \{0, 200, 400, ..., 1000\}$, $om = 2$

Автоматизация использования таксономий для аннотирования текстовых документов

Е.Л. Черняк[1], О.Н. Чугунова[2], Ю.А.Аскарова, С. Насименто[3], Б.Г. Миркин[4]

[1,2,4]Международная Лаборатория Анализа и Выбора Решений НИУ-ВШЭ, Москва, Россия

[1]Witology, Москва, Россия

[3]FCT, Universidade Nova de Lisboa, Caparica, Portugal

[4]Department of Computer Science, Birkbeck University of London, London, UK

Аннотация. Работа посвящена проблеме автоматического аннотирования текстового документа ключевыми словами. Обычно, в качестве источника ключевых слов для аннотирования документа используют таксономии. Наш метод состоит из двух этапов, названных нами «отображением» и «аннотацией». Процедура отображения соотносит таксономические единицы с рассматриваемым текстом. Ее результатом является четкое или нечеткое множество таксономических единиц, иначе множество запроса к таксономии, которое характеризует содержание документа само по себе. На этапе аннотации требуется найти несколько таксономических единиц на высших уровнях таксономии, покрывающих все или почти все множество запроса. Эту задачу можно решить с помощью процедуры оптимального подъема. Найденные таксономические единицы следует считать искомой аннотацией текстового документа. Предлагаемый метод можно применять не только к одному текстовому документу, но и к коллекции текстовых документов. В таком случае, возникает необходимость в еще одном дополнительном этапе – кластеризации таксономических единиц.

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16-18 марта, 2012.

Ключевые слова: анализ текстов, аннотация и отображение, кластерный анализ

Введение

Понятие онтологии как инструмента для хранения знания в некоторой предметной области – одно из самых популярных понятий в современном искусственном интеллекте. Долгое время интересы исследователей в этой области были сосредоточены на методах и алгоритмах создания онтологий. В настоящее время, темой исследований становится использование онтологий в практических задачах. Наша работа принадлежит ко второму направлению.

Она вызвана необходимостью автоматизации аннотирования текстового документа ключевыми словами. Мы предлагаем в качестве источника ключевых слов для аннотирования документа использовать таксономию – иерархическую составляющую онтологии.

Метод, который мы разрабатываем, представляет собой комбинацию нескольких алгоритмов (алгоритм сличения ключевой фразы и текста, алгоритм кластер анализа, алгоритм оптимального подъема в таксономии). Он позволяет оценить и выбрать таксономические темы, наиболее полно соответствующие содержанию текста.

Данная работа выполнялась при частичной финансовой поддержке Научного Фонда НИУ-ВШЭ через Международную лабораторию анализа и выбора решений и коллективный исследовательский проект «Учитель-Ученики» 11-04-0019.

Метод «Отображение-Аннотация» (ОТАН)

Входные данные

Входная информация к методу состоит из двух частей: 1) одного или нескольких (коллекции) текстовых документов; 2) таксономии предметной области.

В нашем представлении, таксономия – это древовидная структура понятий, в которой родительские узлы соответствуют более общим понятиям, чем их потомки. Допустимы и другие отношения между понятиями, так, между узлами на различных уровнях могут быть ссылки.

Для экспериментального анализа мы рассматривали следующую совокупность входных данных: 1) коллекция аннотаций к статьям из журналов Ассоциации вычислительной техники ACM; 2) классификационная система Ассоциации вычислительной техники ACM-CCS[1].

Из журналов ACM мы выбрали несколько, находящихся в свободном доступе, в частности, журнал JETC (Journal of Emerging

Technologies in Computing Systems). Для каждой публикации были извлечены: 1) текст аннотации; 2) множество ключевых слов, предоставленных авторами публикаций; 3) множество индексационных слов, т.е. тем классификационной системы ACM, используемых для аннотирования публикаций на сайте журнала. Индексационные слова были также назначены авторами публикаций. И тексты аннотаций, и ключевые слова были использованы для представления/отражения содержания статьи.

Классификационная система ACM – это четырехуровневое дерево, в котором каждый узел представляет одну тему из области вычислительной техники. Три верхних уровня в этом дереве закодированы, т.е. каждому узлу на этих уровнях присвоен уникальный код, а на нижнем, четвертом уровне находятся не кодированное описание тем третьего уровня, иначе листов таксономического дерева. На первом уровне таксономии располагаются 11 основных узлов, таких как B. Hardware, C. Computer System Organization, и т.д. На втором уровне таксономии находится 81 узел. Глубина дерева не постоянна, некоторые листья находятся на втором уровне, некоторые на третьем. Кроме того, между узлами из разных разделов имеются перекрестные ссылки.

Описание метода ОТАН

Метод получает на вход текст и таксономию, строит профиль текста, и переходит к анализу профиля, описанному ниже. Профилем текста мы называем список тем, приписанных к висячим вершинам (листовым темам) таксономического дерева, и количественных оценок степени их принадлежности этому тексту. Чем степень принадлежности выше, тем больше сходство темы и текста. Оценки рассчитываются по методу аннотированного суффиксного дерева (АСД) [2].

Предлагаемый метод может быть применен к одному тексту или к большому числу текстов, т.е. к их коллекции. В последнем случае, может потребоваться дополнительный этап – нахождение кластеров таксономических тем и оценка вклада каждой темы в полученный кластер.

На основе профилей текстов, состоящих из таксономических тем, или кластеров тем, строятся так называемые множества запроса к таксономии. На последнем этапе метода, ищутся узлы на высших уровнях таксономии, достаточно хорошо покрывающие множества запроса – они и образуют искомому аннотацию.

Более формально, общая схема состоит из следующих шагов:

1. Предварительная подготовка текстов и таксономии, представление их в удобном для машинной обработки виде;
2. Представление текстов как АСД;
3. Вычисление оценок сходства между листовыми темами таксономии и текстами по методу АСД;

4. Построение профилей текстов и построение матрицы сходства листовых тем на основе профилей текстов;
5. Кластеризация таксономических тем;
6. Агрегирование кластеров таксономических тем в таксономии.

Если метод применяется к одному тексту, то шаги 4 и 5 следует опустить и в качестве кластера тем на шаге 6 использовать непосредственно профиль текста.

Предварительная подготовка текста

Каждая таксономическая тема, как правила, представлена одним словосочетанием или предложением. Следовательно, разумно в качестве единицы текста, сравниваемой с темой, выбрать предложение. Предварительная подготовка текста заключается в 1) удалении разметки; 2) разбиении текста на предложения. Таким образом, мы рассматриваем текст как неупорядоченный набор предложений, или строк. Если текст сопровожден ключевыми словами или словосочетаниями статьи, они включаются в этот набор без изменений.

Аннотированное суффиксное дерево как представление текста

Аннотированное суффиксное дерево (АСД) является структурой данных, используемой для хранения и вычисления частот всех фрагментов текста. АСД для строки – это корневое дерево, в котором каждый узел помечен одним символом и одним числом. Говорят, что путь от корня до узла кодирует/прочитывает один из суффиксов строки. Частота, приписанная к узлу, – это частота в строке фрагмента, который кодирует/прочитывает путь от корня до узла. АСД для множества строк кодирует/прочитывает каждый фрагмент каждой строки и его частоту во всем множестве.

Алгоритмы построения АСД и вычисления оценок таксономических тем описаны подробно в [2]. В общих чертах, алгоритм вычисления оценок заключается в следующем. Оценка темы может быть получена как средняя оценка каждого суффикса темы. Каждый суффикс посимвольно накладывается на АСД, т.е. находится такой путь из корня, что кодирует/прочитывает префикс суффикса. В найденном пути рассчитывается условная вероятность каждого узла (отношение частоты узла к частоте его родителя). Сумма условных вероятностей в пути составляет оценку суффикса.

При необходимости, модель текста может быть усложнена. Так, например, ключевым словам может быть придан больший вес, чем обычным предложениям.

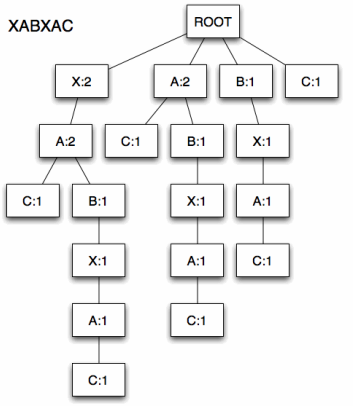


Рис. 1. АСД для строки XABXAC.

Результатом применения метода АСД к таксономическим темам и тексту является профиль текста. С одной стороны, он может быть интерпретирован как нечеткое множество тем, с другой стороны, темы с наибольшими оценками могут считаться четким представлением текста в таксономии.

Нахождение сходства таксономических тем на основе профилей текстов

Каждая листовая тема может быть представлена вектором оценок степени принадлежности этой темы статьям рассматриваемой коллекции статей. Эти векторы подвергаются кластер-анализу с использованием специально разработанного метода FADDIS3 [3] .

Подъем множества запроса на таксономии

Мы рассматриваем подъем множества как способ аннотирования данных в таксономии. Как уже было сказано выше, в качестве множества запроса к таксономии может выступать профиль текста или кластер тем. Таким образом, идея состоит в том, чтобы представить множество запроса, представленное листьями таксономии более общей темой (вершиной таксономии), называемой головная тема. Эта головная тема должна покрывать все множество запроса с минимальным количеством пробелов, то есть, вершин, покрываемых головной темой, но не входящих в множество запроса. В случае, если количество пробелом, не может быть уменьшено, следует спуститься «по дереву таксономии» и взять две головные темы, покрывающие множество запроса с минимальным числом пробелов. Если и это невозможно, число головных тем может быть увеличено. Критерием «подъема» является суммарное чис-

ло головных тем и пробелов, взвешенных соответствующими весами. Рекурсивный алгоритм, минимизирующий этот критерий, описан в[4].

Пример применения метода

Рассмотрим в качестве примера нечеткий кластер тем таксономии ACM-CCS, полученный на выходе алгоритма АСД (таблица 1).

Таблица 1. Нечеткий иллюстративный кластер листовых тем ACM-CCS

Код темы	Степень принадлежности	ACM-CCS Тема
F.1.3	0.597	Complexity Measures and Classes
H.2.3	0.475	Languages
F.2.3	0.401	Tradeoffs between Complexity Measures
H.2.1	0.351	Logical Design
F.1.1	0.322	Models of Computation
H.2.4	0.298	Systems
D.2.8	0.240	Metrics
H.2.8	0.220	Database Applications
J.4	0.211	SOCIAL AND BEHAVIORAL SCIENCES
K.8.0	0.203	General
H.2.6	0.184	Database Machines
F.2.2	0.174	Nonnumerical Algorithms and Problems
I.1.2	0.018	Algorithms
...		

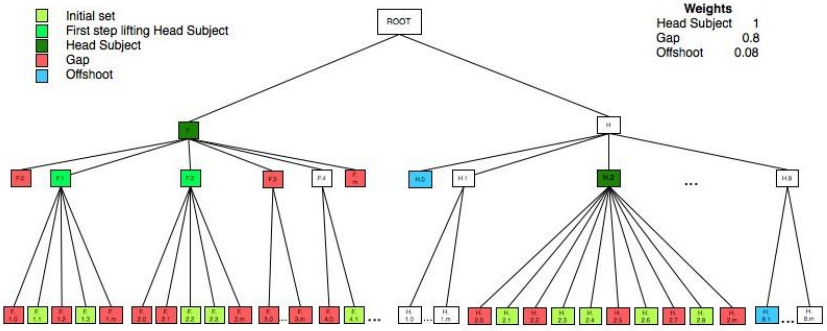


Рис. 2. Результат поднятия множества запроса из таблицы 1.

Рассмотрим пример подъема нечеткого множества тем в таксономии ACM-CCS. В качестве искомая аннотации выступают темы представленные на рисунке темными узлами H.2 Database Management и F. Theory of Computation.

Выводы

В данной статье представлены первые шаги в разработке метода автоматизации аннотирования текстового документа или коллекции текстовых документов ключевыми словами. Он основан на использовании онтологий как средств хранения знания. Метод ОТАН включает в себя два обязательных шага, выполняемых при анализе одного текстового документа, – процедуру отображения и процедуру аннотации. Для анализа коллекции текстовых документа дополнительно используется кластерный анализ.

Дальнейшие направления развития метода ОТАН включают в себя как повышение вычислительной эффективности используемых алгоритмов (алгоритма АСД), так и использование веб-энциклопедий как средства уточнения и детализации таксономии.

Список источников

1 The Association for Computing Machinery Computing Classification System (ACM-CCS), 1998. http://www.acm.org/about/class/ccs98-html

2 Rajesh Pampapathi , Boris Mirkin , Mark Levene, A suffix tree approach to anti-spam email filtering, Machine Learning, v.65 n.1, p.309-338, October 2006

3 Boris Mirkin, Susana Nascimento, Additive Spectral Method for Fuzzy Cluster Analysis of Similarity Data Including Community Structure and Affinity Matrices, Information Sciences, 183(1), pp. 16-34, Elsevier, January 2012

4 Mirkin B., Nascimento S., Fenner T., Fenner T., and Pereira L.M., 2010. Fuzzy Thematic Clusters Mapped to Higher Ranks in a Taxonomy. International Journal of Software and Informatics, 4, 257-275

Влияние разрешения изображений на качество детектирования лиц

Дегтярёв Н.А.[1], Кушнир О.А., Середин О.С.

[1] n.a.degtyarev@gmail.com

ТулГУ, Тула, Россия

Аннотация. Доступность фототехники, способной делать снимки в высоком разрешении, очень сильно повысила качество фотографий, размещаемых в социальных сетях и других ресурсах Интернета. Однако современные алгоритмы поиска лиц на изображениях показывают ухудшение качества детектирования и локализации лиц на фотографиях высокого разращения. В работе представлен анализ влияния разрешения изображений и различных преобразований над ними, таких как размытие, масштабирование в сторону уменьшения, на качество детектирования и локализации лиц на них. Приводятся рекомендации для обработки большого числа изображений высокого разрешения.

Ключевые слова: Изображения высокого разрешения, поиск лиц, оценка качества детектирования.

Введение

В настоящее время качество цифровых фотографий повышается за счёт увеличения их разрешения. С точки зрения обычного человека, увеличение количества точек на фотографии приводит к её визуальному улучшению, т.е. увеличению количества деталей, глубины изображения и т.д. Данный факт достаточно часто используется в рекламных компаниях фото/видео камер.

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16-18 марта, 2012.

В то же время, алгоритмы поиска объектов (в частности лиц) на изображениях, нашедшие широкое применение в социальных сетях, в кодировании и сжатии видео, системах автоматического наблюдения, работают существенно дольше на изображениях высокого разрешения. Это обусловлено увеличением числа областей, подлежащих анализу, числа особых точек и т.п. Более того, в рамках сравнительного тестирования алгоритмов поиска лиц на изображениях [1], проведённого авторами, было установлено, что с увеличением разрешения изображений в тестовой базе ухудшаются показатели качества детектирования и локализации лиц на фотографиях, как содержащих, так и не содержащих лица. Для объяснения этого факта мы выдвинули две гипотезы: **первая -** увеличение числа деталей на изображениях высокого разрешения приводит к увеличению числа ложных срабатываний детекторов; **вторая -** наличие «шумовых лиц» на изображениях, например, отражений лица фотографа в предметах интерьера, в зрачках фотографируемого человека, на элементах одежды и т.п. значительно влияет на показатели качества детектирования.

Целью данной работы является экспериментальная проверка изложенных выше гипотез и изучение влияния размера изображений и числа деталей, находящихся на них, на точность детектирования и локализации объектов. Мы предполагаем, что нет необходимости обрабатывать исходные фотографии высокого разрешения, а можно отмасштабировать их в сторону уменьшения размера с приемлемой потерей точности детектирования и локализации объектов.

Описание экспериментов

Эксперименты заключаются в рутинном поиске лиц на наборах изображений при помощи алгоритма, предложенного П. Виолой и М. Джонсом [2] и реализованного в OpenCV 2.1. Этот алгоритм был выбран, потому, что он имеет наилучшие показатели качества детектирования лиц на изображениях среди некоммерческих/широкодоступных алгоритмов [1,3,4]. Наборы тестовых изображений получены в результате описанных ниже преобразований над тестовой базой фотографий.

Для проверки **первой гипотезы,** т.е. отрицательного влияния мелких деталей на качество и скорость детектирования лиц на изображениях, производилось предварительное сглаживание (т.е. находилась свёртка изображений с функцией Гаусса при различных значениях среднеквадратичного отклонения). Для проверки **второй гипотезы**, т.е. влияния «шумовых лиц» на качество и скорость детектирования лиц на изображениях, изменялся параметр настройки алгоритма, ответственный за предполагаемый размер искомого лица. Для исследования влияния размера изображений на качество и скорость поиска лиц, исходные изо-

бражения масштабировались в сторону уменьшения в кратное число раз. Следует заметить, что такое масштабирование соответствует одновременному размытию (так как происходит потеря деталей) и увеличению значения параметра алгоритма, ответственного за минимальный размер искомого лица.

Экспериментальный материал

Экспериментальные материалы состоят из 36643 фотографий, не содержащих лиц, и 1408 фотографий, на которых отчётливо видны лица людей, разрешением не менее 2560x1920. Максимальный размер изображений, представленных в базе, составлял 3264x2465. Пять наиболее часто встречающихся размеров изображений в базе и количество таких изображений приводится в табл. 1.

Табл. 1. Пять наиболее часто встречающихся размеров изображений в базе

Размер изображения	Количество в базе	% в базе
3264x2448	13268	34,87
3008x2000	4315	11,34
3072x2304	3754	9,87
4000x3000	3496	9,19
3648x2736	2152	5,66
Всего	26985	70,92

Критерии оценки результатов экспериментов

Результаты экспериментов оценивались по стандартным критериям, принятым в биометрии [3]: ошибка первого рода (FRR), ошибка второго рода (FAR), среднее время обработки одного изображения, евклидово расстояние до «идеального алгоритма» (FAR=FRR=0) – $\sqrt{FAR^2 + FRR^2}$. Для масштабирования и размытия изображений применялись соответствующие методы, реализованные в библиотеке OpenCV.

Результаты

Зависимость ошибок первого и второго рода, а так же времени выполнения, от коэффициента СКО при Гауссовском размытии изображений, приводится в табл. 2 Зависимость ошибок детектирования/локализации от коэффициента размытия изображений; от минимального размера искомого лица - в табл. 3; от коэффициента масшта-

бирования исходных изображений - в табл. 4. Графическое представление зависимости FRR от FAR приводится на рис. 1.

Как можно видеть из табл. 2-4, две выдвинутые гипотезы полностью подтвердились: с уменьшением числа деталей (увеличением коэффициента размытия) на изображениях в выборке постепенно увеличивалось качество детектирования до некоторого оптимального значения (СКО = 8), а затем происходила потеря деталей, необходимых для правильного детектирования; (см. табл. 2); с увеличением минимального размера искомого лица на изображениях в выборке постепенно улучшалось качество детектирования лиц, вплоть до достижения наилучшего значения: минимального размера искомого лица, равного 240 пикселей, что примерно соответствует 8-9% от максимальной стороны изображения (см. табл. 3).

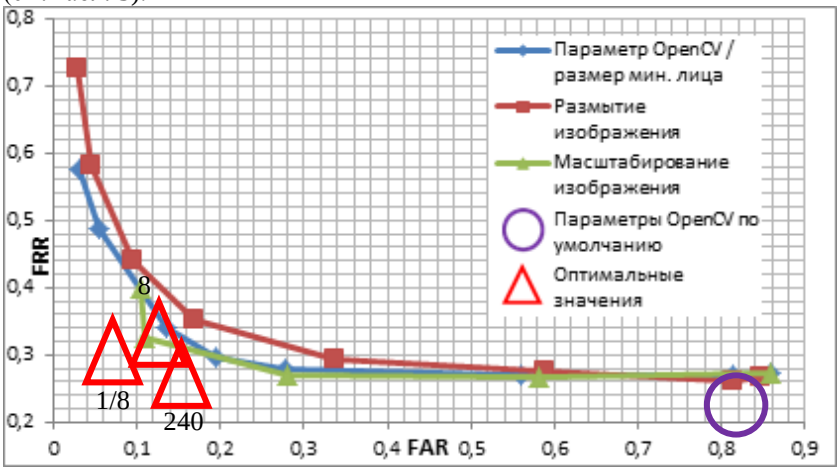


Рис. 1 Зависимость FRR от FAR при различных условиях поиска; значения параметров, при которых достигается минимальная обобщённая ошибка, выделены треугольниками

В свою очередь, масштабирование изображений в базе в 1/8 раз (до размера, сопоставимого с 320x240 – 640x480) позволяет достичь минимальной обобщённой ошибки детектирования, так как масштабирование в сторону уменьшения размера соответствует одновременному размытию (происходит потеря деталей) и увеличению значения параметра алгоритма, ответственного за минимальный размер искомого лица. Пример работы алгоритма поиска лиц с различными параметрами на предобработанных изображениях приводится на рис. 2.

Также из табл. 2-4, можно увидеть, что с уменьшением количества деталей на изображениях и с увеличением минимального размера иско-

мого объекта, уменьшается время средней обработки одного изображения. Наибольший прирост производительности с приемлемой потерей качества детектирования изображения (см. табл. 4) может быть достигнут изменением размера исходного изображения, так как для его обработки требуется гораздо меньший объём памяти, а также такое преобразование уменьшает число мелких деталей на изображениях и эквивалентно увеличению значения параметра алгоритма, ответственного за минимальный размер искомого лица.

Рис. 2. Пример работы алгоритма поиска лиц (Viola-Jones, реализация: OpenCV 2.1) с различными параметрами на предобработанных изображениях размера 3264x2448 пикселей: **a** – исходное изображение, настройки по умолчанию; **b** – минимальный размер искомого лица (параметр алгоритма) установлен равным 240; **c** - размытое изображение (СКО=8), настройки алгоритма поиска по умолчанию; **d** - размер изображения уменьшен в 4 раза, параметры по умолчанию

Такие результаты экспериментов являются достаточно ожидаемыми, так как алгоритм детектирования, использующий скользящие окна, не должен применяться к фрагментам изображения, где вероятность найти объект несущественна, поскольку это может лишь привести к увеличению ошибок детектирования.

В данной работе, ошибки первого (FRR) и второго (FAR) рода рассматриваются равнозначными. Однако в большинстве социально ориентированных приложений алгоритмов поиска лиц на изображениях (например, автофокусировка на групповых снимках, поиск лиц на фотографиях в социальных сетях, интерактивная бытовая техника и т.п.) FRR имеет большее значение, чем FAR. Для таких задач, авторы считают более целесообразным масштабировать изображения до размера, сопоставимого с 640x480.

Табл. 2 Зависимость ошибок детектирования/локализации от коэффициента размытия изображений

СКО	FRR	FAR	$\sqrt{FAR^2 + FRR^2}$	Среднее время детектирования., мс	Среднее время пред. обработки, мс	Среднее суммарное время, мс
0	0,272	0,858	0,900	16512	0	16512
1	0,268	0,846	0,887	14289	475	14764
2	0,262	0,814	0,8554	10951	634	11585
4	0,276	0,588	0,650	6292	1243	7535
6	0,293	0,337	0,446	3888	2136	6024
8	0,352	0,168	0,389	2646	2845	5491
10	0,440	0,095	0,450	2040	3284	5324
12	0,582	0,045	0,583	1621	3956	5577
14	0,727	0,030	0,728	1543	4232	5775

Табл. 3 Зависимость ошибок детектирования/локализации от параметра алгоритма

Парам. алгор. Viola-Jones Мин. размер искомого лиц	FRR	FAR	$\sqrt{FAR^2 + FRR^2}$	Среднее время детектирования., мс	Среднее время пред. обработки, мс	Среднее суммарное время, мс
20 (значение по умолчанию)	0,272	0,858	0,900	16512	ОТСУТВУЕТ	16512
40	0,269	0,813	0,856	8952		8952
80	0,269	0,559	0,620	3003		3003
160	0,279	0,277	0,393	1490		1490
240	0,296	0,195	0,353	1266		1266
320	0,341	0,135	0,366	1022		1022
480	0,489	0,054	0,492	850		850
640	0,576	0,032	0,577	800		800

Табл. 4 Зависимость ошибок детектирования/локализации от коэффициента масштабирования исходного изображения

Масштаб. Коэффициент	FRR	FAR	$\sqrt{FAR^2 + FRR}$	Среднее время детектирования., мс	Среднее время пред. обработки, мс	Среднее суммарное время, мс
1/16	0,399	0,104	0,412	86	100	186
1/8	0,325	0,110	0,343	422	109	531
1/4	0,271	0,280	0,389	1213	110	1323
1/2	0,266	0,580	0,638	4319	121	4440
1	0,272	0,858	0,900	16512	0	16512

Заключение

В работе было показано, что главными помехами в поиске объектов на изображениях высокого разрешения является переизбыток деталей и наличие «шумовых» лиц на них. Наилучшим способом для устранения этих помех является масштабирование изображений примерно до 320x240-640x480 пикселей. При таком разрешении удаётся заметно увеличить скорость поиска объектов на изображениях, уменьшить объём памяти, занимаемой базой изображений, снизить ошибки первого и второго рода. Результаты были вполне прогнозируемы, и основную ценность работы мы видим именно в экспериментальном подтверждении, кажущихся очевидными, результатов, и полученных сравнительных оценках ошибок и времени обработки при детектировании лиц. Впрочем, зная тенденцию современных разработчиков **не** утруждать себя подбором параметров настройки, а использовать их значения, заданные по умолчанию, ещё раз заострим внимание на результатах, демонстрирующих колоссальное снижение ошибки второго рода (рис. 2) и предостержем их от подобной небрежности. В дальнейшем, при продолжении цикла работ, связанных с экспериментальными сравнениями различных алгоритмов поиска лиц на изображениях, мы планируем использовать процедуру масштабирования для приведения размеров фотографий к рекомендованной в этой работе величине.

Стоит заметить, что для уменьшения ошибок детектирования мы считаем возможным комбинирование совокупности результатов работы алгоритма, полученных на наборе изображений, сгенерированных путем

различных преобразований (масштабирование, сглаживание и т.п.). Такая методика предложена в [5].

Список источников

1 Degtyarev, N., Seredin, O.: Comparative Testing of Face Detection Algorithms. In: Elmoataz,A., Lezoray, O., Nouboud, F., Mammazz, D., Meunier, J. (eds.) ICISP 2010. LNCS, vol. 6134, pp. 200 209, Springer, Heidelberg (2010).

2 Viola, P., Jones, M.J.: Robust Real-Time Face Detection. International Journal of Computer Vision 57, pp. 137 154 (2004).

3 Wechsler H.: Reliable face recognition methods: system design, implementation and evaluation, Springer, 329 p. (2007).

4 Zhang, C., Zhang, Z.: A Survey of Recent Advances in Face Detection, Microsoft Research Technical Report #MSR-TR-2010-66, pp.17, (2010).

5 Degtyarev, N., Seredin, O.: A Geometric Approach to Face Detector Combining. In: C. Sansone, J. Kittler, F. Roli (Eds.), MCS 2011, LNCS 6713, pp. 299 308, Springer, (2011).

Визуализация данных социосемантической сети

А. Друца[1], К. Яворский[2]

[1] adrutsa@yandex.ru, [2] konstantin.yavorsky@witology.com

Московский государственный университет имени М. В. Ломоносова, механико-математический факультет, Москва, Россия
Witology, Москва, Россия

Аннотация. В работе рассматривается пакет программ социально-сетевого анализа WitoAnalytics, позволяющий анализировать и визуализировать данные социосемантической сети платформы Witology. Данный пакет является незавершенным проектом (work in progress) и в статье приведены первые его возможности по визуализации нескольких типов подграфов социосемантической сети.

Ключевые слова: визуализация данных; социосемантическая сеть; графы.

Введение

Компания Witology занимается решением конкретных практических задач при помощи формирования деятельного сообщества людей, развивая при этом коллективный разум участников. Для достижения этой цели в компании разработана и используется коллаборативная программная платформа. Существенным её отличием от других подобных систем является наличие прямого участия в деятельности сообщества специально обученных фасилитаторов. В связи с возможным большим количеством

Игнатов Д. И., Яворский Р. Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16–18 марта, 2012

участников сообщества, возникает необходимость визуального представления данных об их деятельности на платформе. Эти данные могут быть использованы фасилитаторами как аналитический материал, позволяющий быстро принимать правильные решения.

В настоящее время существует большое количество программного обеспечения, предназначенного для анализа и визуализации данных социальных сетей (Social Network Analysis, SNA). К ним относятся как широкопрофильные программы по анализу графов всевозможного типа, например, UCINet [2], Pajek [3] и Cytoscape [4], так и программы по анализу текстов, например, Discourse Network Analyzer [5] и AutoMap [6]. Кроме этого к SNA программам относятся специализированные программы по анализу социальных сетей, например, NodeXL [7], которая позволяет извлекать, анализировать и визуализировать данные из таких сетей, как Twitter и Facebook. Поскольку платформа Witology представляет собой социосемантическую сеть [1], то для её анализа необходим специальный пакет программ, заточенный под анализ и визуализацию данного типа сети.

Постановка задачи

В работе [1] социосемантическая сеть определяется как тройка $\mathbb{G} = (G, C, A)$, в которой

- $G = \{V, E_1, \ldots, E_k; \pi, \delta_1, \ldots, \delta_k\}$ — социальный граф — взвешенный ориентированный мультиграф, где V — участники сети, $E_1, \ldots, E_k \subset V \times V$ — различные отношения между участниками, $\pi : V \to \Pi$ — функция профиля пользователя, $\delta_i : E_i \to \Delta_i$ ($i \in \{1, \ldots, k\}$) — параметры соответствующих отношений;
- $C = \{T, R_1, \ldots, R_m; \theta, \gamma_1, \ldots, \gamma_m\}$ — мультиграф контента, где T — множество всех элементов контента, $R_1, \ldots, R_m \subset T \times T$ — отношения между контентом, $\theta : T \to \Theta$ — параметры контента, $\gamma_i : R_i \to \Gamma_i$ ($i \in \{1, \ldots, k\}$) — параметры соответствующих отношений;
- $A \subset V \times T$ — отношение между социальным графом и контентом.

Для анализа такого графа поставлена задача: написать специализированный пакет программ, далее именуемый WitoAnalytics. Данный пакет должен уметь в максимально информативной форме визуализировать наиболее значимую деятельность участников на платформе, в частности, оценки пользователей, формирование текстов и др., причем такие визуализации должны демонстрировать как некоторые временные срезы базы данных, так и изменение данных во времени.

Полученные результаты

Как упоминалось ранее, разрабатываемый авторами пакет можно рассматривать как одно из множества ПО социально-сетевого анализа, но заточенного под анализ и визуализацию графа конкретного типа — социосемантической сети платформы Witology. Сеть, приведенная для иллюстраций в настоящей статье, имеет более 500 участников, однако на визуализациях отображено более 200 существенных участников сети. К настоящему моменту пакет WitoAnalytics позволяет строить несколько визуализаций однодольного графа и визуализацию двудольного графа.

Граф межпользовательских оценок. Рассмотрим следующий взвешенный ориентированный подграф социосемантической сети:

$$G_e = \{V_e, E_e, \delta_e\},$$

где $\delta_e : E_e \to [-k, k] \times \mathbb{N}$ — двумерный вес ребра, где первая компонента отвечает среднему значению оценок узла в некоторой шкале $[-k, k]$, а вторая — количеству оценок. Тогда такой подграф назовем графом межпользовательских оценок. Такой граф может быть получен, например, из данных об оценках пользователями текстов. Граф межпользовательских оценок может быть визуализирован в WitoAnalytics двумя способами:

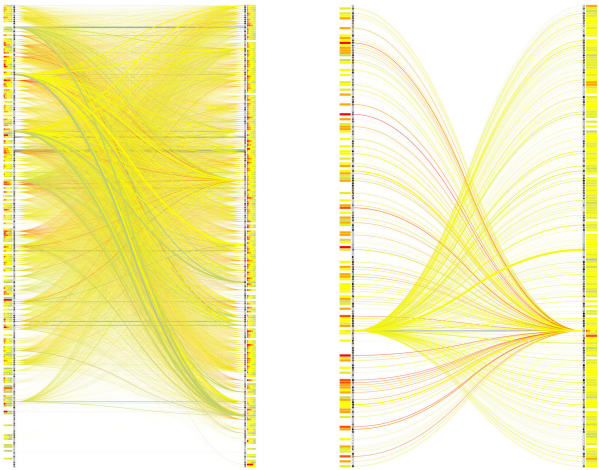


Рис. 1. (A) — визуализация «елка», (B) — визуализация локальной окрестности пользователя

— в виде двудольного представления, где каждому элементу из V_e соответствует два узла, расположенных на плоскости, причем

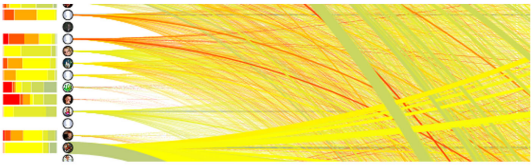


Рис. 2. Увеличенный фрагмент визуализации «елка»

вертикальные координаты этих узлов совпадают. В этом случае направление ребра всегда совпадает с направлением горизонтальной оси. Такая визуализация условно называется «елкой» и ее пример приведен на рис. 1. Здесь толщина ребра соответствует количеству оценок между узлами, а цвет — среднему значению оценок, при этом диагональные ребра выделены отдельным цветом. Рядом с узлами приводятся гистограммы распределения выходящих оценок (слева) и входящих оценок (справа). Справа на рис. 1 приводится локальная окрестность рассматриваемого графа, то есть отображены только те ребра, которые соединены с отдельно выбранным узлом, узлы без видимых ребер удалены. На рис. 2 приведен увеличенный фрагмент визуализации «елка».

— в виде однодольного представления, где элементу из V_e соответствует один узел, размещенный на окружности. Все входящие ребра примыкают к узлу под одним и тем же углом, равно как и выходящие ребра, причем углы входящих и выходящих ребер отличаются и симметричны относительно радиуса данного узла. Такая визуализация условно называется «солнцем» и ее пример приведен на рис. 3.

Визуализация «елка» позволяет быстро и качественно представить общую картину оценок между пользователями, а также выявить характер оценок отдельных пользователей, выделяющихся на фоне остальных. Так, например, из рис. 1 видно, что в среднем все пользователи давали нейтральные оценки друг другу. В то же время, среди них выделяются отдельные узлы, оценки которых практически полностью отрицательные, или, наоборот, положительные. Такие пользователи, например, могут быть взяты на особый контроль фасилитаторами. Кроме того, на такой визуализации можно мгновенно обнаружить сговор некоторой группы пользователей по отрицательному оцениванию отдельного узла. Это выражалось бы в виде нескольких широких красных линий, ведущих к одному из узлов в правой колонке. И при этом остальные ребра, входящие в него, в среднем не имели бы красного цвета.

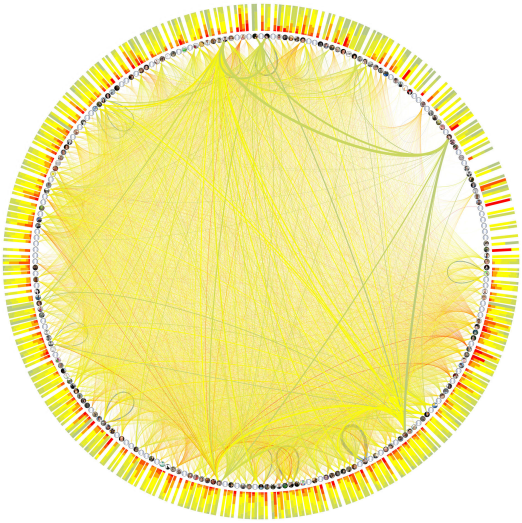


Рис. 3. Визуализация «солнце»

К сожалению, визуализация «елка» не позволяет выявить так называемые «группы накруток», между пользователями которых существует договоренность о взаимном положительном оценивании. Так, уже группа из двух участников на данной визуализации должна иметь вид толстых зеленых пересекающихся ребер, проверить симметричность которых является достаточно трудоемким процессом при большом количестве улов. Для решения подобной задачи очень хорошо подходит визуализация «солнце», где входящие и выходящие концы ребер узла совпадают.

Граф поддержки идей. Рассмотрим некоторое сужение социсемантического графа $\bar{\mathbb{G}} = (\bar{G}, \bar{C}, A)$, где контент $\bar{C}$ содержит лишь одно отношение $\bar{R}$, являющееся строгим частичным отношением порядка на множестве $\bar{T}$, а $\bar{G}$ тоже содержит одно отношение $\bar{E}$, индуцируемое отношением A следующим образом:

$$v\bar{E}w \iff \exists t, \tau \in \bar{T} \mid vA\tau \wedge wAt \wedge t\bar{R}\tau \wedge \tau \in \bar{T}',$$

где $\bar{T}'$ — множество максимальных элементов из $\bar{T}$ относительно $\bar{R}$. Тогда такой подграф назовем графом поддержки идей. Граф поддержки идей визуализируется WitoAnalytics следующим образом. Узлы $\bar{V}$ размещаются на внешней концентрической окружности, а узлы $\bar{T}'$ — на внутренней. Размер узлов и их отклонение от линии окружности соответствует количеству ребер. Такая визуализация условно называется «глазом» и ее пример приведен на рис. 5.

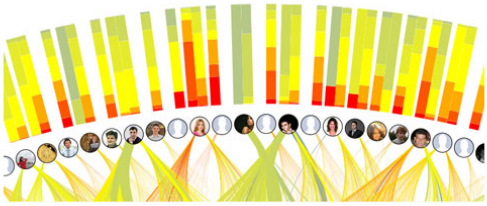


Рис. 4. Увеличенный фрагмент визуализации «солнце»

«Планы на будущее»

Поскольку Witology является относительно молодой компанией, первый этап инвестиций (Serie A Funding) был произведен в феврале 2011 года, то данная работа по анализу и визуализации социосемантической сети платформы является незавершенным проектом, в котором предстоит решить множество аналитических задач и задач визуализации, среди которых можно выделить следующие открытые вопросы:

— какие данные визуализировать, чтобы, например, выявить сговор и «группы накруток» среди участников;
— как размещать узлы и проводить ребра;
— какие выставлять пороговые значения и для каких параметров узлов и ребер;

Список источников

1. Yavorskiy R., *Research Challenges of Dynamic Socio-Semantic Networks*, `http://www.witology.com`.
2. Borgatti S., Everett M. and Freeman L., UCINET, Analytic Technologies, `http://www.analytictech.com/ucinet/`.
3. Pajek, `http://vlado.fmf.uni-lj.si/pub/networks/pajek/`.
4. Cytoscape, `http://www.cytoscape.org/`.
5. Leifeld P. Discourse Network Analyzer, `www.philipleifeld.de/discourse-network-analyzer/`.
6. Auto Map, Casos, `http://www.casos.cs.cmu.edu/projects/automap/`.
7. NodeXL, CodePlex, `http://nodexl.codeplex.com/`.

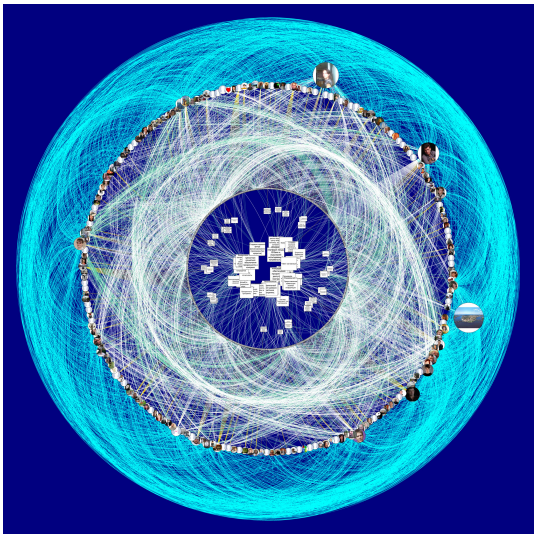


Рис. 5. Визуализация «глаз»

8. Апанович З. В., *От рисования графов к визуализации информации*, (препринт) Новосибирск, 27 с, 2007 (`http://www.iis.nsk.su/files/preprints/148.pdf`).

Система автоматического квазиреферирования WEXSY

Л.М. Ермакова

liana87@mail.ru

Пермский государственный национальный исследовательский университет

Аннотация. В данной статье рассматривается система автоматического квазиреферирования Википедии WEXSY. WEXSY генерирует квазиреферат заданного объема по заданной теме. Для извлечения релевантных предложений предлагается метод сглаживания по локальному контексту. Для формирования итогового реферата решается задача о рюкзаке с использованием интегральной меры, объединяющей релевантность и читаемость. Текст представляется в виде графа. Упорядочивание предложений осуществляется с помощью жадного алгоритма решения задачи коммивояжера.

Ключевые слова: квазиреферирование, сглаживание, локальный контекст, именованные сущности, задача о рюкзаке, задача коммивояжера, разрешение анафоры, Википедия

Введение

Многие пользователи следят за новостными сообщениями в Твиттере. В такой системе текст ограничен сравнительно небольшим количеством символов (140), что примерно соответствует заголовку статьи и первому предложению и этого далеко не всегда достаточно для полного понимания. Важной компонентой содержания новостного сообщения являются именованные сущности (персоналии, организации, топони-

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16-18 марта, 2012.

мы). Именованные сущности могут быть не известны пользователю, а их поиск в Интернете на мобильном устройстве (например, телефоне) является достаточно медленным, а зачастую дорогостоящим. В связи с этим целесообразна разработка системы реферирования, предоставляющей контекст для новостного твита на основе локальных ресурсов (например, дампа Википедии).

Материал: 132 твита, состоящих из заголовка статьи New York Times и первого предложения новости, а также англоязычный дамп Википедии (апрель 2011г.). Статьи Википедии представляли собой отдельные документы (3 217 015 непустые страницы), состоящие из аннотации, секций и подзаголовков.

Задача: Разработка системы, составляющей реферат по твиту на основе Википедии объемом не более 500 слов [1].

Предлагаемый метод состоит из предварительной обработки текстов, поиска релевантных предложений и их расположения в определенном порядке. На первом этапе проводится поиск документов, релевантных новостному твиту. Для этого используется открытая поисковая система Terrier [2]. Твиты и найденные документы анализируются при помощи Stanford CoreNLP [3]. На основе размеченных текстов строится индекс для отдельных предложений, включающий не только леммы, но и именованные сущности. Для поиска релевантных предложений применяется TF-IDF метрика. При извлечении предложений мы исходили из *гипотезы*, что релевантность предложения зависит от контекста, как левого, так и правого; и значимость этого контекста убывает по мере удаления от целевого предложения. Наиболее релевантные предложения, общим объемом 1000 слов, включаются во множество кандидатов (множество предложений, которые могут быть включены в результирующий реферат).

Важную роль в читаемости реферата играет степень разрешения анафоры. В WEXSY анафора разрешается при помощи Stanford CoreNLP, а в результирующем реферате личные местоимения уточняются ссылкой на референт, указанный в скобках. Поскольку в системах сводного реферирования невозможно использование исходного порядка следования предложений (порядок предложений обычно зависит от расставляемых акцентов), мы предлагаем метод, основанный на графовом представлении текста. Несмотря на то, что в литературе широко применяется представление текста в виде графа [4][5][6][7], классические задачи теории графов практически не используются. Мы определили читаемость как суммарное расстояние между соседними предложениями текста. Для предложений-кандидатов вычисляется F-мера, объединяющая релевантность предложения и общую читаемость реферата, а в результирующий реферат включаются предложения с максимальным весом (задача о рюкзаке). При этом порядок вычисляется при помощи

жадного алгоритма решения задачи коммивояжера. Поскольку существительные несут наибольшее количество информации, при значительном совпадении существительных предложения считаются идентичными и в реферат включается только одно из них.

Обзор существующих методов

Работы по контекстуализации небольших текстов появились относительно недавно. Основная идея подобных систем заключается в поиске документов, релевантных тексту небольшого объема (например, микроблогу) [8]. В отличие от других методов, отображающих тексты на страницы Википедии, мы предлагаем формировать сводный квазиреферат, содержащий максимально релевантную и полную информацию в сжатом виде, т.е. первоочередной задачей становится не поиск документов, а наиболее подходящих фрагментов.

Реферат - сжатая версия исходного документа или множества документов, выполненная в определенном жанре и предоставляющая точную, кратко выраженную идею оригинального текста [9]. Системы реферирования могут формировать как квазирефераты, состоящие из наиболее важных предложений исходного текста, так и настоящие рефераты, в которых предложения перефразируются [5][7][10].

Статистические методы реферирования

Одним из наиболее распространенных подходов к автоматическому реферированию документов является выделение значимой информации на основе частоты ключевых слов, позиции предложения в тексте, ключевых оборотов и т.д. [7][11][12]. При генерации тематических рефератов возможно расширение запроса, например, синонимами [13]. Традиционно для поиска релевантных фрагментов используется статистических языковых моделей [8][14]. В отечественной литературе также встречаются методы выделения кластеров, основанные на сканирующих статистиках $n(d)$, которые фиксируют максимальное число точек n, находящихся в интервале длины d при всевозможных расположениях интервала внутри единичного отрезка. Найденные таким образом кластеры интерпретируются как сверхфразовые единства, отражающие семантику фрагмента текста. Каждому предложению приписывается вес, исходя из количества кластерообразующих лексических единиц [15]. Алгоритм Manifold-Ranking предполагает итеративный процесс ранжирования документов. На первом шаге вычисляется нормализованная матрица сходства предложений. Выбирается предложение с максимальным весом. Рейтинг остальных предложений пересчитывается с учетом штрафа за сходство с выбранным предложением. Процесс повторяется пока не будет составлен реферат требуемой длины [4].

В случае применения машинного обучения система использует корпус исходных текстов и соответствующих им рефератов. Признаки, которые должны быть представлены в результате, определяются статистически [**Ошибка! Закладка не определена.**]. Корпус также дает информацию о том, какой объем исходного текста должен быть сохранен, какова средняя длина предложения в реферате, количество абзацев и количество предложений в каждом абзаце [16].

Лингвистические методы

Исторически первыми были системы автоматического реферирования, основанные на правилах. Система SUMMONS отбирает информацию из базы знаний и ранжирует ее в зависимости от частоты встречаемости. Лингвистическая компонента проверяет правильность составления предложений [12]. Система SUMMARIST синтезирует статистический подход с применением правил для генерации квазирефератов [16][17].

В системах реферирования, работающих с определенными видами текстов, возможно использование жанровых особенностей (вид, структура текста [11]). Например, в новостных текстах наиболее значимая информация содержится в начале статьи [16], специфично употребление глагольных словоформ [18], а описания одних и тех же событий повторяются в течение короткого промежутка времени [19][20]. При этом информация может быть аналогичной или противоречивой. Одним из первых подходов было сохранение только повторяющейся информации. Все предложения кластеризуются путем рекурсивного сравнения компонент деревьев составляющих вне зависимости от порядка их следования, а затем преобразуются к структуре предикат-аргумент. Похожие предложения также могут быть определены при помощи шаблонов перефразирования, полученных корпусным исследованием. Структура предикат-аргумент позволяет выявить общую информацию. Выявленные фразы дополняются необходимой информацией (описанием сущностей, временными ссылками, ссылками на источник) [20]. В новостных текстах может применяться хронологический порядок следования. В этом случае каждому событию приписывается временной штамп, после чего проводится сортировка. Временные ссылки (например, *сегодня, в пятницу*) должны быть заменены на явную дату, т.к. могут быть неоднозначными в реферате [20]. Структура текста может быть вычленена с помощью фраз-маркеров. На первом этапе выделяются наиболее релевантные предложения, после чего они классифицируются в зависимости от риторической функции [19]. Идея риторической структуры также нашла отражение в CST (cross-document structure theory). В отличие от теории риторической структуры, CST пытается описать риторическую структуру нескольких связанных текстов [21].

Описание метода

Поиск релевантных предложений

По текстам, размеченным с помощью Stanford CoreNLP, строится индекс для отдельных предложений, включающий не только леммы, но и именованные сущности. Инвертированный индекс – отсортированный список троек вида (слово, документ, вес) [22]. Для поиска релевантных предложений применяется TF-IDF метрика [23]:

$$TF(t_k, d) \times IDF(t_k) = \frac{n_d(t_k)}{\sum_i n_d(t_i)} \times \log \frac{|D|}{|d_i \supset t_k|}, \quad (1)$$

где t_k – искомый термин, $n_d(t_k)$ – количество вхождений термина t_k в документ d, $|D|$ - размер коллекции, а $|d_i \supset t_k|$ - количество документов, содержащих термин t_k. Возможно использование трех мер сходства:

1. косинусного коэффициента (cosine similarity) [22]:

$$similarity(Q, S) = \frac{\sum_{i=1}^{n} Q_i \times S_i}{\sqrt{\sum_{i=1}^{n}(Q_i)^2} \times \sqrt{\sum_{i=1}^{n}(S_i)^2}}, \quad (2)$$

2. коэффициента Дайса (Dice similarity) [24]:

$$similarity(Q, S) = \frac{2\sum_{i=1}^{n} Q_i \times S_i}{\sum_{i=1}^{n}(Q_i)^2 + \sum_{i=1}^{n}(S_i)^2}, \quad (3)$$

3. коэффициента Жаккарда (Jaccard similarity) [23]:

$$similarity(Q, S) = \frac{\sum_{i=1}^{n} Q_i \times S_i}{\sum_{i=1}^{n}(Q_i)^2 + \sum_{i=1}^{n}(S_i)^2 - \sum_{i=1}^{n} Q_i \times S_i}, \quad (4)$$

где Q - твит, S - предложение, Q_i - i-ый токен в твите, а S_i - i-ый токен в предложении. Если i-ый токен отсутствует в твите или предложении, то Q_i или S_i равно 0 соответственно.

Дополнительно вводятся коэффициенты для частей речи, именованных сущностей, предложений без глаголов в личной форме, предложений-определений, а также для аннотаций в статьях Википедии. Для поиска определений используется лингвистический шаблон $< NE >< Be_{pers} >< NounPhrase >$*, где* NE *– именованная сущность,* Be_{pers} *– личная форма глагола to be, а* $NounPhrase$ *– именная группа.*

Поскольку предложения обычно намного меньше документа, традиционные системы информационного поиска дают худшие результаты в задаче поиска релевантных предложений. Системы поиска документов исходят из предположения, что документ «о запросе», в то же время этого не достаточно для извлечения предложений [14]. Чаще всего сглаживание предложений производится по всей коллекции документов [14]. Однако смысл предложения за-

висит от контекста, поэтому целесообразней рассматривать не весь корпус, а только локальный контекст [14].

Мы считаем, что релевантность предложения зависит от контекста, как левого, так и правого; и значимость этого контекста убывает по мере удаления от целевого предложения. Таким образом, вес целевого предложения R_t представляет собой взвешенную сумму соседних предложений r_i и его самого r_0:

$$R_t = \sum_{i=-k}^{k} w_i \times r_i \tag{5}$$

$$w_i = \begin{cases} \frac{1 - w_t}{k + 1} \times \frac{k - |i|}{k}, 0 < |i| \leq k \\ w_t, i = 0 \\ 0, |i| > k \end{cases} \tag{6}$$

$$\sum_{i=-k}^{k} w_i = 1, \tag{7}$$

где w_t – вес целевого предложения, задаваемый пользователем, а w_i – веса предложений из k контекста. Эти коэффициенты линейно уменьшаются с ростом $|i|$ (i отрицательно для левого контекста и положительно для правого). Если количество предложений в левом или правом контексте меньше k, остаток добавляется к весу исходного предложения. Таким образом, сумма всегда остается равной 1.

Читаемость реферата в значительной степени зависит от уровня разрешения местоименной анафоры. Предложения, содержащие неразрешенные анафоры, могут не приниматься во внимание [25]. Другой подход заключается в добавлении предложений, в которых встречается референт. В обоих случаях снижается общая релевантность реферата, поэтому в данной работе личные местоимения дополняются референтом, если он находится в другом предложении: *it [the Security Council] noted…* Анафора разрешалась с помощью Stanford CoreNLP.

Все предложения сортируются по рейтингу. Во множество кандидатов на включение в итоговый реферат отбираются предложения с максимальным весом так, чтобы их суммарный объем не превышал $2n$, n – максимальное количество слов в результирующем тексте.

Определение порядка следования предложений

При определении порядка следования предложений применяется следующая гипотеза: соседние предложения должны быть похожими, а суммарное расстояние между ними минимальным. На первом этапе строится граф. Если кандидаты являются соседями в исходном тексте, они объединяются в одну вершину. Далее используется алгоритм ближ-

него соседа [26]. Основной недостаток метода в том, что в «идеальном» реферате все предложения будут одинаковыми, поэтому целесообразно отсекать избыточную информацию. Самый простой способ – введение порогового значения, однако метод малопригоден для предложений разной длины (например, отличающихся количеством прилагательных и наречий). Мы исходим из предположения, что наиболее значимая информация содержится в существительных, поэтому при их значительном совпадении, предложения считаются идентичными и в реферат включается только одно из них.

Предложение может быть релевантным, но не иметь смысла в данном контексте. В этом случае имеет смысл заменить его другим. Если принять количество слов в предложении за вес, а интегральную оценку релевантности и читаемости за ценность, то отбор предложений можно смоделировать задачей о рюкзаке. Многие методы решения задачи о рюкзаке требуют выполнения неравенства треугольника, что неприемлемо к реферированию. Поэтому был использован метод ветвей и границ [27]. В качестве интегральной оценки релевантности и читаемости мы использовали F-меру:

$$F = \frac{Relevance \times Readability}{\alpha \times Relevance + (1 - \alpha) \times Readability} \quad (8)$$

$$Readability = 1 - Length(Path)\,, \quad (9)$$

где $Length(Path)$ – длина наилучшего пути, вычисленная методом ближнего соседа, $Relevance$ – суммарная релевантность предложений, α – параметр, задаваемый пользователем.

Оценка результатов

На форуме INEX 2011 было произведено сравнение базовой системы, не включавшей разрешение анафоры и без упорядочивания, с 11 другими системами, 7 из которых использовали индекс Indri (http://qa.termwatch.es/). Была произведена оценка системы, работающей с тремя наборами параметров (с различными коэффициентами сходства, количеством фраз в контексте, используемом для сглаживания, весами целевого предложения и коэффициентам для секций, которые не являются аннотациями): *ID12RIRIT_default, ID12RIRIT_07_2_07_1_dice и ID12RIRIT_05_2_07_1_jac*.

Предлагаемая система показала наилучшие результаты с точки зрения релевантности [28][29]. Рефераты сравнивались с исходными статьями из New York Times (Таблица 1), а также с множеством релевантных предложений (Таблица 2), полученных экспертным методом, по формуле:

$$\sum \log\left(\frac{max(P(t|reference), P(t|summary))}{min(P(t|reference), P(t|summary))}\right) \quad (10)$$

RUN является идентификатором системы с определенными параметрами. RANKING соответствует рангу системы. UNIGRAM показывает расстояние измеренное при помощи униграмм, BIGRAM – биграмм, а WITH 2-GAP – биграмм, расположенных в окне из 4 слов. AVERAGE соответствует среднему значению.

Таблица 1. Расстояние до исходных статей New York Times

№	RUN	RANKING	UNIGRAM	BIGRAM	WITH 2-GAP	AVERAGE
1.	***ID12RIRIT_05_2_07_1_jac***	***0.104925***	***0.0447***	***0.076644***	***0.104925***	***0.076629***
2.	***ID12RIRIT_07_2_07_1_dice***	***0.104933***	***0.044728***	***0.076659***	***0.104933***	***0.076646***
3.	***ID12RIRIT_default***	***0.104937***	***0.044739***	***0.076668***	***0.104937***	***0.076653***
4.	ID129RRun1	0.10604	0.045626	0.077687	0.10604	0.077664
5.	ID132RRun1	0.106118	0.046187	0.077947	0.106118	0.077946
6.	Baselinesum	0.10646	0.046049	0.078101	0.10646	0.078084
7.	ID126RRun1	0.106536	0.045998	0.078113	0.106536	0.078101
8.	ID128RRun2	0.106601	0.046065	0.078179	0.106601	0.078167
9.	ID138RRun1	0.106605	0.046122	0.078225	0.106605	0.078201
10.	ID129RRun2	0.10708	0.046751	0.078775	0.10708	0.078746
11.	ID129RRun3	0.107209	0.046798	0.078864	0.107209	0.078837
12.	ID126RRun2	0.10728	0.046852	0.078916	0.10728	0.078897
13.	ID128RRun3	0.107341	0.046872	0.07895	0.107341	0.078937
14.	ID123RI10UniXRun1	0.107491	0.047084	0.079149	0.107491	0.079121
15.	Baselinemwt	0.10766	0.047508	0.079385	0.10766	0.079387
16.	ID62RRun1	0.10769	0.047283	0.079344	0.10769	0.079319
17.	ID128RRun1	0.107911	0.047482	0.07955	0.107911	0.079529
18.	ID62RRun3	0.107969	0.047598	0.079638	0.107969	0.079614
19.	ID62RRun2	0.107993	0.047674	0.079689	0.107993	0.079662
20.	ID123RI10UniXRun2	0.108036	0.047735	0.07973	0.108036	0.07971
21.	ID123RI10UniXRun3	0.108681	0.048326	0.080369	0.108681	0.080337
22.	ID46RJU_CSE_run1	0.108948	0.0487	0.080679	0.108948	0.08065
23.	ID46RJU_CSE_run2	0.10895	0.048702	0.08068	0.10895	0.080651

№	RUN	RANKING	UNIGRAM	BIGRAM	WITH 2-GAP	AVERAGE
24.	ID124RUNAMiiR12	0.109389	0.04931	0.081181	0.109389	0.081161
25.	ID124RUNAMiiR3	0.110429	0.050541	0.082313	0.110429	0.082288

Таблица 2. Расстояние до множества релевантных предложений

№	RUN	RANKING	UNIGRAM	BIGRAM	WITH 2-GAP	AVERAGE
1.	***ID12RIRIT_default***	***0.105506***	***0.048639***	***0.07867***	***0.105506***	***0.078697***
2.	***ID12RIRIT_07_2_07_1_dice***	***0.105747***	***0.048781***	***0.078857***	***0.105747***	***0.07889***
3.	***ID12RIRIT_05_2_07_1_jac***	***0.106195***	***0.049083***	***0.079249***	***0.106195***	***0.079277***
4.	ID129RRun1	0.107806	0.050253	0.080676	0.107806	0.080689
5.	ID129RRun2	0.110616	0.05178	0.082987	0.110616	0.082954
6.	ID128RRun2	0.111033	0.052372	0.08345	0.111033	0.083438
7.	ID138RRun1	0.111516	0.052383	0.08374	0.111516	0.083716
8.	ID132RRun1	0.111666	0.052567	0.083836	0.111666	0.083857
9.	ID126RRun1	0.112529	0.053464	0.084754	0.112529	0.084752
10.	Baselinesum	0.114346	0.053691	0.085915	0.114346	0.085881
11.	ID126RRun2	0.114404	0.054608	0.086328	0.114404	0.086311
12.	ID128RRun3	0.11512	0.054904	0.086875	0.11512	0.086846
13.	ID129RRun3	0.115219	0.054883	0.086928	0.115219	0.086896
14.	ID46RJU_CSE_run1	0.115557	0.056092	0.087656	0.115557	0.087617
15.	ID46RJU_CSE_run2	0.11558	0.056122	0.087682	0.11558	0.087643
16.	ID62RRun3	0.117158	0.056456	0.088684	0.117158	0.088667
17.	ID123RI10UniXRun2	0.117196	0.056143	0.088538	0.117196	0.088537
18.	ID128RRun1	0.117406	0.05655	0.088886	0.117406	0.088852
19.	Baselinemwt	0.117854	0.055786	0.088604	0.117854	0.088701
20.	ID62RRun1	0.118016	0.05661	0.089207	0.118016	0.089203
21.	ID123RI10UniXRun1	0.118346	0.056717	0.08948	0.118346	0.08945
22.	ID62RRun2	0.118805	0.057196	0.089971	0.118805	0.089925
23.	ID124RUNAMiiR12	0.122111	0.060737	0.09335	0.122111	0.093325
24.	ID123RI10UniXRun3	0.123938	0.061052	0.094556	0.123938	0.094502
25.	ID124RUNAMiiR3	0.124792	0.062794	0.095747	0.124792	0.095726

Кроме того, базовая система была оценена с точки зрения читаемости (Таблица 3). Асессоры должны были определить, присутствуют ли в рефератах синтаксические ошибки, неразрешенные анафоры и избыточная информация. Оценка представляет собой среднее нормализованное число слов в верных фрагментах. Фрагмент считается верным, если он не содержит ошибок и понятен в данном контексте [30]. Основным недостатком системы была неразрешенная анафора, что было устранено в новой версии. Помимо этого, уровень релевантности был повышен, благодаря включению референта.

Таблица 3. Результаты сравнения читаемости рефератов

№	RUN	SCORE	№	RUN	SCORE
1.	ID129R_Run1	359.0769	13.	ID126R_Run2	296.3922
2.	ID129R_Run2	351.8113	14.	ID62R_Run2	288.6154
3.	ID126R_Run1	350.6981	15.	ID128R_Run1	284.4286
4.	ID46R_JU_CSE_run1	347.92	16.	ID62R_Run3	277.9792
5.	***ID12R_IRIT_05_2_07_1_jac***	***344.1154***	17.	ID62R_Run1	266.1633
6.	***ID12R_IRIT_default***	***339.9231***	18.	ID18R_Run1	260.1837
7.	***ID12R_IRIT_07_2_07_1_dice***	***338.7547***	19.	ID123R_I10UniXRun1	246.9787
8.	ID128R_Run2	330.283	20.	ID123R_I10UniXRun2	246.5745
9.	ID46R_JU_CSE_run2	330.14	21.	ID123R_I10UniXRun3	232.6744
10.	ID129R_Run3	325.0943	22.	ID124R_UNAMiiR12	219.1875
11.	ID138R_Run1	306.2549	23.	Baseline_mwt	148.2222
12.	ID128R_Run3	297.4167	24.	ID124R_UNAMiiR3	128.3261

Выводы

В рамках исследования предлагается метод, состоящий из поиска релевантных предложений и их расположения в определенном порядке. На основе текстов, размеченных при помощи Stanford CoreNLP, строится индекс для отдельных предложений, включающий не только леммы, но и именованные сущности. Для поиска релевантных предложений применяется TF-IDF метрика. Дополнительно вводятся коэффициенты для частей речи, предложений без глаголов в личной форме, предложений-определений, а также для аннотаций в статьях Википедии. Кроме того предложения с личными местоимениями расширяются референтами, находящимися в других предложениях. При извлечении предложений мы исходили из

гипотезы, что релевантность предложения зависит от контекста, как левого, так и правого; и значимость этого контекста убывает по мере удаления от целевого предложения. Наиболее релевантные предложения, общим объемом 1000 слов, включаются во множество кандидатов (множество предложений, которые могут быть включены в результирующий реферат).

В WEXSY читаемость определяется как суммарное расстояние между соседними предложениями текста. Для предложений-кандидатов вычисляется F-мера, объединяющая релевантность предложения и общую читаемость реферата, а в результирующий реферат включаются предложения с максимальным весом (задача о рюкзаке). При этом порядок вычисляется при помощи алгоритма ближнего соседа в задаче коммивояжера. Поскольку существительные несут наибольшее количество информации, при значительном совпадении существительных предложения считаются идентичными и в реферат включается только одно из них.

В дальнейшем планируется улучшить результаты релевантности за счет расширения запроса при помощи синсетов из WordNet. Синонимы также позволят более адекватно оценить избыточную информацию и меру связности между предложениями.

Список источников

1. Bellot P., Mothe J., Moriceau V., SanJuan E., Tannier X., Question Answering Track. URL: https://inex.mmci.uni-saarland.de/tracks/qa/

2. http://terrier.org/

3. http://nlp.stanford.edu/software/corenlp.shtml

4. Wan X., Yang J., Xiao J., Manifold-Ranking Based Topic-Focused Multi-Document Summarization. Proceedings of the 20th international joint conference on Artifical intelligence. 2007. C.2903-2908.

5. Vivaldi J., Cunha I., Ramэrez J., The REG summarization system at QA@INEX track 2010. INEX 2010 Workshop Pre-proceedings. 2010.

6. Morris J., Hirst G., Lexical cohesion computed by thesaural relations as an indicator of the structure of text. Computational Linguistics. 1991.

7. Erkan G., Radev D.R., LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. Journal Of Artificial Intelligence Research. 2004. C.457-479.

8. Meij E., Weerkamp W., de Rijke M., «Adding Semantics to Microblog Posts,» in Proceedings of the fifth ACM international conference on Web search and data mining, New York, NY, USA, 2012.

9. Saggion H., Lapalme G., Generating Indicative-Informative Summaries with SumUM. Association for Computational Linguistics. 2002.

10. Gholamrezazadeh S., Salehi M.A., Gholamzadeh B., A Comprehensive Survey on Text Summarization Systems. Computer Science and its Applications. 2009. С.1-6.

11. Seki Y., Automatic Summarization Focusing on Document Genre and Text Structure. ACM SIGIR Forum. 2005. С.65-67.

12. Radev D.R., McKeown K.R., Generating natural language summaries from multiple on-line sources. Computational Linguistics - Special issue on natural language generation. 1998. С.469-500.

13. Soriano-Morales E.P., Medina-Urrea A., Sierra G., Mendez-Cruz C.F., The GIL-UNAM-3 summarizer: an experiment in the track QA@INEX'10. INEX 2010 Workshop Pre-proceedings. 2010.

14. Murdock V.G. Aspects of Sentence Retrieval. Dissertation. 2006.

15. Гусев В.Д., Мирошниченко Л.А., Саломатина Н.В., Тематический анализ и квазиреферирование текста с использованием сканирующих статистик. Труды междунар. конф. Диалог. 2005. С.121-125.

16. Lin C.Y., Hovy E., Identifying Topics by Position. Proceedings of the fifth conference on Applied natural language processing. 1997. С.283-290.

17. Lin C.Y., Assembly of Topic Extraction Modules in SUMMARIST. AAAI Spring Symposium on Intelligent Text Summarisation. 1998.

18. Овчинникова И.Г., Черепанова Л.Л., Ягунова Е.В., Вариативность новостных текстов в аспекте информационного анализа. Проблемы динамической лингвистики: матер. Международной научн.й конф.., посвященной 80-летию профессора Л.Н. Мурзина. 2010. С.401-406.

19. Teufel S., Moens M., Summarizing scientific articles: experiments with relevance and rhetorical status. Computational Linguistics. 2002. С.409-445.

20. Barzilay R., McKeown K.R., Elhadad M., Information fusion in the context of multi-document summarization. ACL '99 Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics. 1999. С.550-557.

21. Radev D.R., A common theory of information fusion from multiple text sources step one: cross-document structure. Proceedings of the 1st SIGdial workshop on Discourse and dialogue. 2000.

22. Агеев М.С., Добров Б.В., Метод эффективного расчета матрицы ближайших соседей для полнотекстовых документов. Вестник Санкт-Петербургского университета. Информатика. 2011.

23. Manning C.D., Raghavan P., Schütze H., Introduction to Information Retrieval. Cambridge University Press. 2008.

24. Ягунова Е.В., Пивоварова Л.М., Исследование структуры новостного текста как последовательности связных сегментов. Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Международной конференции «Диалог». 2011.

25. Тарасов С.Д., Автоматическое составление обзорных рефератов новостных сюжетов. Вестник Балтийского государственного технического университета. 2008. С.61–67.

26. Морозенко В.В., Дискретная математика: учеб. пособие. Пермь: ПГУ. 2008.

27. Вирт Н., Алгоритмы и структуры данных. СПб.: Невский Диалект. 2008.

28. SanJuan E., Moriceau V., Tannier X., Bellot P., Mothe J., 2011 QrAzy Track Overview. 2011. URL: http://www.limsi.fr/~xtannier/files/inex_11_QA_results.pdf.

29. SanJuan E., Moriceau V., Tannier X., Bellot P., Mothe J., Overview of the INEX 2011 Question Answering Track (QA@INEX). INEX 2011 Workshop Pre-proceedings. 2011.

30. Bellot P., Mothe J., Moriceau V., SanJuan E., Tannier X., Question Answeing Track. 2012. URL: http://perso.limsi.fr/Individu/xtannier/files/QAINEX-11-readability-results.pdf.

Применение марковской модели для анализа влиятельности участников интернет-сообществ[1]

сообществ[1]

Федянин Д.Н.

dfedyanin@inbox.ru

Институт проблем управления им. В. А. Трапезникова РАН

Аннотация. В работе исследуется взаимное влияние участников(агентов) социальной друг на друга в рамках так называемой марковской модели. Основной задачей проводимого исследования являлась проверка гипотез о зависимости вычисляемой влиятельности агентов от используемых методов идентификации модели, а также непротиворечивость результатов получаемых в результате использования различных методов. В исследовании используются данные только о связях между агентами, но игнорируются данные о написанных ими друг другу сообщениях.

Ключевые слова: социальная сеть, распространение информации, марковская модель, влиятельность, альфа-центральность.

Введение

В работе исследуется взаимное влияние участников социальной сети (будем называть их агентами в соответствии с терминологией, принятой в [1]) друг на друга. Под влиянием понимается процесс и результат изменения субъектом (субъектом влияния) поведения другого субъ-

[1] Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект 10-07-00104).

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16-18 марта, 2012.

екта (индивидуального или коллективного объекта влияния), его установок, намерений, представлений и оценок (а также основывающихся на них действий) в ходе взаимодействия с ним [2]. Как показывают наблюдения психологов [3] в социальной сети агенты часто не имеют достаточной для принятия решения информации или не могут самостоятельно обработать её, поэтому их решения могут основываться на наблюдаемых ими решениях и/или представлениях других агентов (социальное влияние) [1].

Анализ был проведен на данных трех сообществ Живого Журнала. Сайт Живого Журнала (http://www.livejournal.com) состоит из блогов – последовательностей сообщений, называемых постами. Википедия [http://ru.wikipedia.org] дает следующее определение «Блог (англ. blog, от web log — интернет-журнал событий, интернет-дневник, онлайн-дневник) — веб-сайт, основное содержимое которого — регулярно добавляемые записи (посты), содержащие текст, изображения или мультимедиа. Для блогов характерны недлинные записи временной значимости, отсортированные в обратном хронологическом порядке (последняя запись сверху). Отличия блога от традиционного дневника обусловливаются средой: блоги обычно публичны и предполагают сторонних читателей, которые могут вступить в публичную полемику с автором (в комментарии к блогозаписи или своих блогах)». Авторы таких постов называются блогерами. Большинство постов доступно для чтения и комментирования другим блогерам. Живой Журнал также предоставляет возможность блогерам объединяться в сообщества и подписываться на чтение блогов. В этом случае все новые посты в выбранных блогах отображаются в специальной новостной ленте блогера. Блогер может состоять в нескольких сообществах одновременно.

Информация о сообществах, подписках и самих записях в большинстве случаев является открытой и доступной любому интернет – пользователю. Для каждого из сообществ известен список участников и список друзей для каждого участника. Хранение данных осуществлялось в трех таблицах: список сообществ, список блогеров и список связей между блогерами. Количество записей в списке участников (членов сообществ Живого Журнала) 964, 2960, 6587, количество связей, соответственно, 6359, 49504 и 190427. Далее в работе термины «блогер» и «агент» будем употреблять как синонимы.

Мотивация

Из работ как российских, так и зарубежных авторов известен ряд свойств марковской модели [1], описывающей социальную сеть. Традиционно анализ ведется на исключительно теоретическом уровне [4], однако для успешного применения полученных теоретических резуль-

татов необходимо иметь какой-либо хорошо зарекомендовавший алгоритм идентификации модели по наблюдаемым данным. Известны работы, в которых подобные методы описываются, например [1].

Несмотря на высокую эффективность существующих вычислительных алгоритмов, основным недостатком марковской модели является необходимость возведения исходной матрицы влияния в бесконечную или достаточно высокую степень. Кроме того остается некоторая неопределенность в определении исходной матрицы взаимного доверия агентов и их связей с другими агентами.

Большие перспективы в этом вопросе открывает рассмотрение сообщений, которыми обмениваются агенты в социальной сети.

В предлагаемом исследовании было проведено предварительное сравнение различных методов определения влиятельности агентов в социальной сети не учитывающее данные о сообщениях, которыми обмениваются агенты. Основой для идентификации сети являлись данные указанные агентами о том, чьи блоги они читают. Формат имеющихся данных и пример данных представлен ниже

Таблица 1. Фрагмент используемых в статье данных.

Идентификатор связи между агентами	Идентификатор читающего агента	Идентификатор агента, блог которого читает	Идентификатор сообщества, к которому принадлежат оба агента
1	1	2	1
2	3	2	1
3	4	1	2

Обзор существующих математических моделей

В литературе существует несколько подходов к описанию взаимодействия участников в социальной сети: марковская модель (или модель Де Гроота) [5], модель с порогами (Liner Threshold Model) [6], модель независимых каскадов (Independent Cascade Model) [7], модель просачивания и заражения, модель Изинга, модель клеточных автоматов и другие [1]. Модели исследованы с точки зрения различных аспектов, в том числе учитывающих: условия сходимости мнений членов социальной сети (см. [8]), динамическое изменение влияния, скорость сходимости, условие единственности итогового мнения (см. [9]). В данной работе будет использоваться модель, подробно описанная в книге [1].

В некоторых моделях используется ранжирования агентов, например, индексы влияния: индекс Хёде-Баккера (Houde-Bakker) [10], индекс

Банцафа [11], вычисление импакт-фактора научных журналов, ранжирование страниц в интернете алгоритмами PageRank , а также упорядочение по параметрам «промежуточность» [12], «центральность» [13], «коэффициент кластеризации» [14] и другие.

Используемые обозначения и определения.

В силу своей широкой известности, описание марковской модели в данной работе для краткости не приводится. Подробное описание этой модели можно найти, например, в работе [1]. Суть модели состоит в представлении взаимодействия агентов в виде последовательности линейных преобразований в пространстве мнений этих агентов, где мнение агентов в сети представляется в виде вектора, в котором *i*-ая координата – мнение *i*-го агента.

Матрица линейного преобразования называется *матрицей прямого влияния*, а матрица линейного преобразования, являющегося результатом бесконечного применения таких линейных преобразований, называется *матрицей результирующего влияния*. Для того чтобы построенная модель являлась марковской необходимо, чтобы матрица прямого влияния была стохастической. Для такой модели известен ряд интересных результатов как российских, так и зарубежных ученых[1,5,15].

Отметим, что влиятельность агентов, определяемую как

$$w_j = \sum_i a_{ij}^{\infty}$$

где a_{ij}^{∞} - элемент транзитивного замыкания матрицы прямого влияния, можно вычислить также для исходной стохастической матрицы прямого влияния. Для того чтобы различать эти влиятельности, будем называть влиятельность определенную по матрице прямого влияния *прямой влиятельностью*, а определенную по матрице транзитивного замыкания *результирующей влиятельностью*.

Обычный метод идентификации агентов заключается в том, что матрица прямого влияния формируется из матрицы смежности по формуле

$$a_{ij} = \frac{b_{ij}}{\sum_i b_{ij}}$$

где a_{ij} – веса в матрице прямого влияния, b_{ij} – элементы матрицы смежности.

В некоторых случаях можно попытаться учесть зависимость влияния от авторитетности агента. Можно рассмотреть случаи, когда авторитетность линейно или нелинейно пропорциональна количеству друзей агента, то есть

$$a_{ij} = \frac{f_j(b_{ij})}{\sum_i f_j(b_{ij})}; f_j(x) = (\sum_i b_{ij})^{\beta}$$

где f_j имеет смысл авторитетности i-го агента.

Проверяемые гипотезы

Гипотеза 0. Сеть, построенная по имеющимся данным, является безмасштабной (Scale-Free).

Гипотеза 1. Прямая влиятельность зависит от количества друзей агента, и зависимость между ними близка к некоторой степенной функции.

Гипотеза 2. Количество друзей не коррелирует с результирующей влиятельностью оценки влиятельности.

Гипотеза 3. Есть корреляция между результирующими влиятельностями агентов, построенных методами с различным учетом авторитетности агентов.

Гипотеза 4. Прямая влиятельность агента не коррелирует с его результирующей влиятельностью.

Гипотеза 5. Выполнение гипотез не зависит от размера сети.

Результаты анализа данных

Традиционным способом проверки гипотезы безмасштабности сети является построение гистограммы зависимости количества агентов с заданным количеством друзей от количества друзей. Известно, что для безмасштабных сетей эти зависимости являются степенными. На рис. 1 приведена зависимость для сети, состоящей из 964 агентов. Под рангом понимается номер агента в списке всех агентов, отсортированном по количеству друзей.

По имеющимся данным определяется, что выделенное для анализа подмножество агентов само обладает свойствами безмасштабной сети, и соответственно гипотеза 0 выполнена. Этот результат сам по себе важен, так как, строго говоря, не всякое подмножество агентов безмасштабной сети образует безмасштабную сеть.

Для проверки гипотезы о том, что прямая влиятельность зависит от количества друзей агента, и зависимость между ними близка к некоторой степенной функции был построен рисунок 2. Численное значение линейной корреляции равно 0,75.

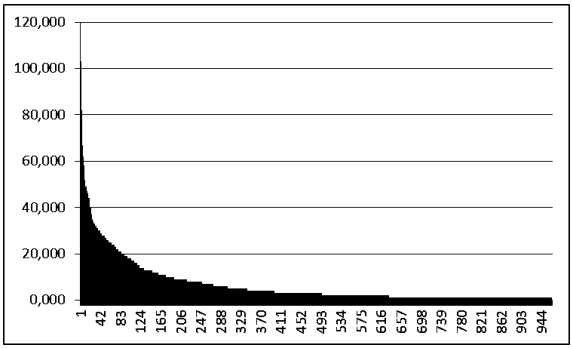


Рис. 1. Проверка гипотезы о безмасштабности подсети сети. Зависимость количества друзей у агента от его ранга.

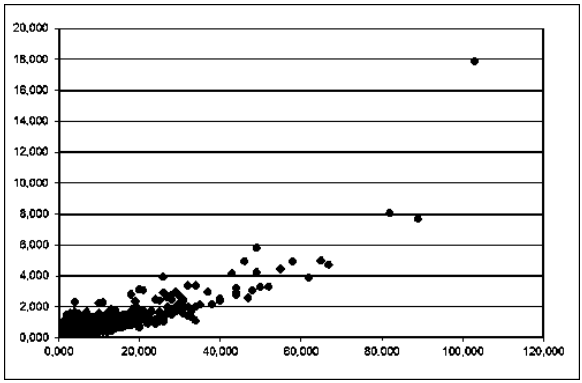


Рис. 2. Зависимость прямой влиятельности агентов (по вертикали) от количества у этих агентов друзей (по горизонтали).

По данным, показанным на рисунке видно, что зависимость существует и имеет степенной вид. Таким образом, гипотезу 1 также можно считать подтвержденной.

Для проверки гипотезы о том, что количество друзей не коррелирует с результирующей влиятельностью, был построен соответствующий график. Численное значение корреляции равно 0,45.

Помимо интуитивно понятной линейной зависимости обращает на себя внимание почти вертикальный «хвост», его наличие означает, что существует некоторое количество агентов, обладающих небольшим количеством друзей, однако оказывающим существенное. В частности существует три агента, результирующее влияние каждого из которых превосходит результирующее влияние агента с наибольшим количеством друзей (чье влияние достаточно понятно).

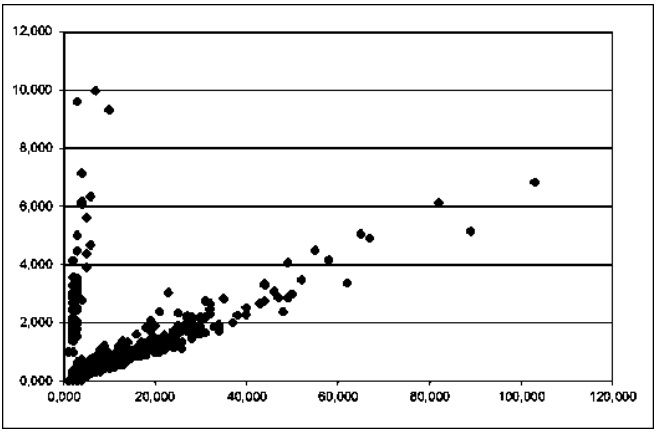


Рис. 3. График зависимости результирующей влиятельности агента (по вертикали) от количества у друзей у этого агента (по горизонтали).

Наличие этого явления теоретически было обосновано, однако проверки на экспериментальных данных его существования, если они существуют, остаются малоизвестными. Отметим, что наличие всего лишь двух явно выделенных линий остается неясным и его причины следует исследовать отдельно.

Таким образом, можно признать, гипотеза 2 выполняется, однако требует дополнительных оговорок.

Для проверки гипотезы 3, о наличии корреляции между результирующими влиятельностями агентов, построенных методами с различным учетом авторитетности агентов, были построены следующие зависимости. Численное значение линейной корреляции равно 0,64.

На рисунке 4 видно, что корреляция отсутствует. Это факт является важным в силу того, что делая предположения о значимости количества друзей агента на его авторитетности, можно получить, вообще говоря, заметно отличающиеся результаты. Если крайние несколько точек считать аномальными выбросами, то опять выделяется два «хвоста» - основного, для которого зависимости линейная и не равная константе, и второй – соответствующий случаю отсутствия увеличения результирующего влияния у агентов, несмотря на увеличение их транзитивного влияния в случае отсутствия учета авторитетности. То есть показывает, что существует некоторое количество влиятельных агентов с низким авторитетом. Это согласуется с выводом, полученным в процессе проверки гипотезы 1.

В силу вышесказанного можно утверждать, что корреляция между результирующими влиятельностями имеет сложный характер, и таким

образом, гипотеза 3, не может считаться выполненной без дополнительных пояснений и дополнений.

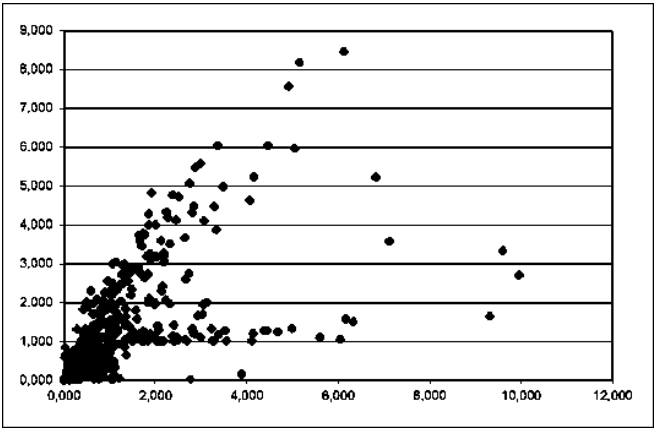


Рис. 4. Зависимость транзитивной влиятельности агента, вычисленной без учета его авторитетности (по горизонтали) от транзитивной влиятельности, вычисленной с учетом влиятельности (β=4) (по вертикали).

Для проверки гипотезы 4, о том, что прямая влиятельность агента не коррелирует с его результирующей влиятельностью, были построены следующие зависимости. Численное значение линейной корреляции равно 0,60.

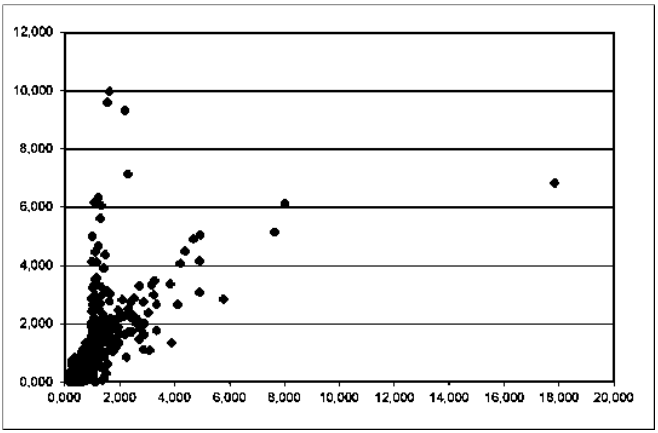


Рис. 5. Зависимость результирующей влиятельности агента (по вертикали) от его прямой влиятельности (по горизонтали), вычисленных без учета авторитетности агентов.

На рисунке 5 видно, что линейная корреляция между прямой и результирующей влиятельностями отсутствует. Это имеет также и содержательное применение, так как, если мы полагаем, что не все агенты могут принимать решения, основанные на вычислении результирующей влиятельности, и поэтому вынуждены использовать для этого прямую влиятельность. Тогда мы приходим к достаточно очевидному выводу, что в реально существующих сетях такие агенты будут находиться в заблуждении, однако, важно подчеркнуть, что они будут ошибаться лишь в отношении меньшего количества агентов. Более чем в 50% случаев они будут правы. Однако мы можем считать, что гипотеза 4 не выполняется.

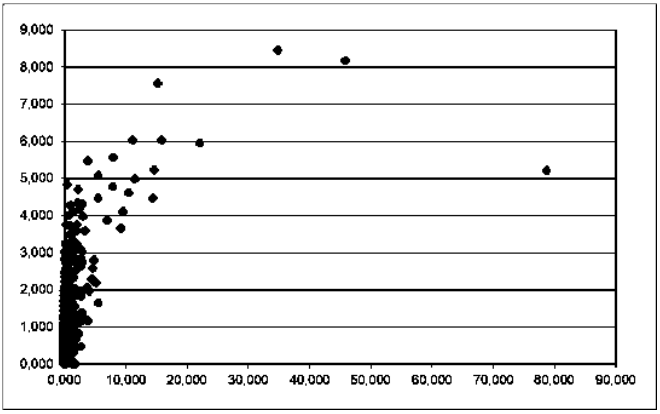


Рис. 6. Зависимость результирующей влиятельности агента (по вертикали) от его прямой влиятельности (по горизонтали), вычисленных с учетом авторитетности агентов.

В случае, изображенном на рисунке 6, линейная корреляция заметна лишь для более высоких значений, если проигнорировать существующий единичный выброс. Численное значение линейной корреляции равно 0,56. Таким образом, результат проверки выполнения гипотезы 4 в отношении в этих условиях имеет специфический вид. Гипотеза 6, заключающаяся в предположении, что выполнение гипотез не зависит от размера сети, пока не была проверена - такие исследования являются возможным направлением дальнейших исследований.

Выводы

Исследование показало наличие некоторого количества аномалий и эффектов, которые необходимо учитывать при планировании

идентификации марковских моделей по экспериментальным данным.

Было показано, что авторитетность агентов оказывает существенное влияние на результат вычисления влиятельности агентов.

Было продемонстрирована специфическая зависимость результирующей влиятельности от прямой влиятельности.

Было показано, что существует аномальный кластер агентов, обладающих небольшим количеством друзей, но имеющих большое значение результирующей влиятельности.

Перспективные направления дальнейших исследований

Интересным может быть продолжение исследования проверкой гипотезы 5, а также включением в анализ возможность учета обмена сообщениями между агентами.

Представляется также перспективным построить корреляционные зависимости между ранжированием агентов методом вычисления показателя альфа-центральность, ранжированием алгоритмом PageRank, а также другими популярными и широко используемыми методами и прямыми и результирующими влиятельностями агентов.

Список источников

1. Губанов Д.А., Новиков Д.А., Чхартишвили А.Г. Социальные сети: модели информационного влияния, управления и противоборства. - М.: Физматлит, 2010. - 228 с.

2. Glossary on Control Theory and its Applications. – URL: http://glossary.ru

3. Deutsch M., Gerard H. Study of Normative and Informational Social Influense upon Individual Judgement // Journal of Abnormal and Social Psychology. 1995. №51, P. 629-636.

4. Зуев А.С., Федянин Д.Н. Модели управления мнениями агентов в социальных сетях / Проблемы управления. № 1. М.: ИПУ РАН, 2011. С. 37-45.

5. DeGroot M.H. Reaching a Consensus // Journal of American Statistical Assotiation. 1974. №69, P.118-121.

6. Ganovetter M. Threshold Models of Collective Behavior // American Journal of Sociology. 1978. V. 83.№6, P.1420-1443.

7. Goldberg J., Libai B., Muller E. Talk of the Networks: A Complex Systems looks at the Underlying Process of Word-of-Mouth // Marketing Letters. 2001, №2, P.11-34.

8. Berger. R.L. Nessessary and Sufficient Conditions for Reaching a Consensus using DeGroot's method // Journal of American Statistical Association. 1981. V. 76. P. 415 – 419.

9. Golub. B., Jackson M. Naive Learning in Social Networks: Convergence, Influence and the Wisdom of Crowds. 2007. URL:http://www.stanford.edu/~jacksonm/naivelearnig.pdf.

10. Hoede. C., Bakker R. A Theory of Dicisional Power // Journal of Mathematical Sociology. 1982. №8, P.309-322.

11. Rusinowska A., Swart H. Generalizing and Modifying the Hoede-Bakker Index. Theory and Appications of Rational Structures as Knowledge Instruments. №2, Springer's Lecture Notes in Artificial INtellegence 4342. Springer, 2007. P.60-88.

12. Freeman L. A set of measures of centrality based upon betweenness. // Sociometry №40. 1977. P. 35–41.

13. Borgatti S, Everett M. A Graph-Theoretic Perspective on Centrality. // Social Networks (Elsevier) 28. 2005. P. 466–484.

14. Wasserman S., Faust K., Social Network Analysis: Methods and Applications. // Cambridge: Cambridge University Press. 1994.

15. Jackson M. Social and Economic Networks. — Princeton: Princeton University Press, 2008.

Методика совместной обработки разносезонных изображений Landsat-TM и создания на их основе карты наземных экосистем Московской области

Е. А. Гаврилюк[1], Д. В. Ершов[2]

[1]egor@ifi.rssi.ru, [2]ershov@ifi.rssi.ru

Центр по проблемам экологии и продуктивности лесов РАН,

117997 Москва, Профсоюзная ул., 84/32

Аннотация. Статья посвящена разработке и реализации методики моделирования поведения спектральных яркостей подстилающих поверхностей для весеннего и осеннего периодов года на основе летнего безоблачного композитного изображения. В качестве исходных данных используются разносезонных изображений Landsat-TM. Также рассматривается методика использования полученных в результате моделирования изображений для целей картографирования наземных экосистем на региональном уровне.

Ключевые слова: Landsat-TM, пространственное и временное моделирование, композитные изображения, наземные экосистемы, классификация многозональных изображений, тематическое картографирование.

Введение

Одним из важных направлений развития методов картографирования наземных экосистем и, прежде всего, лесов является адаптация существующих методов и алгоритмов обработки спутниковых изображе-

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16-18 марта, 2012.

ний для построения карт растительности на региональном уровне. Для этого наилучшим образом могут подходить данные высокого пространственного разрешения серии спутников LANDSAT, находящиеся в открытом доступе в мировых каталогах и архивах.

Преимуществом использования этих данных является пространственное разрешение, составляющее 30 метров на местности (0,09 га), и семь спектральных каналов в видимом, ближнем и среднем инфракрасном диапазонах электромагнитного спектра. В дополнение к этому огромный архив сцен, накопленный за два последних десятилетия.

Недостатком является низкая периодичность съемки одной и той же территории (один раз в 16 дней), что для бореальной зоны становится критичным при учете облачного покрова и теней от облаков.

Поэтому для картографирования растительности на региональном уровне необходимо разрабатывать и применять методы построения безоблачных композитных изображений [1-4]. При этом удается достичь желаемого результата только для летних периодов вегетационного сезона, так как наибольшее число измерений приходится на этот период года.

В свете этого, необходимы подходы, позволяющие моделировать поведения спектральных яркостей подстилающих поверхностей в весенний и осенний периоды года, используя летние безоблачные композитные изображения. Это дает возможность повысить информативность данных и улучшить распознавание различных видов лесной и нелесной растительности наземных экосистем региона. Далее на примере Московской области будет рассмотрена методика пространственно-временного моделирования изображения для тематической классификации.

Моделирование временных стеков

Исходные данные

При моделировании поведения спектральных яркостей подстилающих поверхностей для весеннего и осеннего периодов года и последующего создания карты наземных экосистем Московской области использовались следующие материалы (рис.1):

1). Разновременные космические снимки из коллекции Landsat 5 архива Landsat-TM пространственным разрешением 30 м на следующие гранулы в разметке WRS2: p177r021, p177r022, p178r020, p178r021, p178r022, p179r020, p179r021, p179r022, p180r020, p180r021. Данные сцены полностью покрывают территорию Московской области, а также значительную часть сопредельных субъектов. Временной охват снимков – с 6 мая по 24 сентября 2010, 2009 и 2007 годов.

2). Трехканальная (3, 4 и 5 каналы) безоблачная мозаика на Московскую область по состоянию на лето 2010 года (полученная по методике [1]).

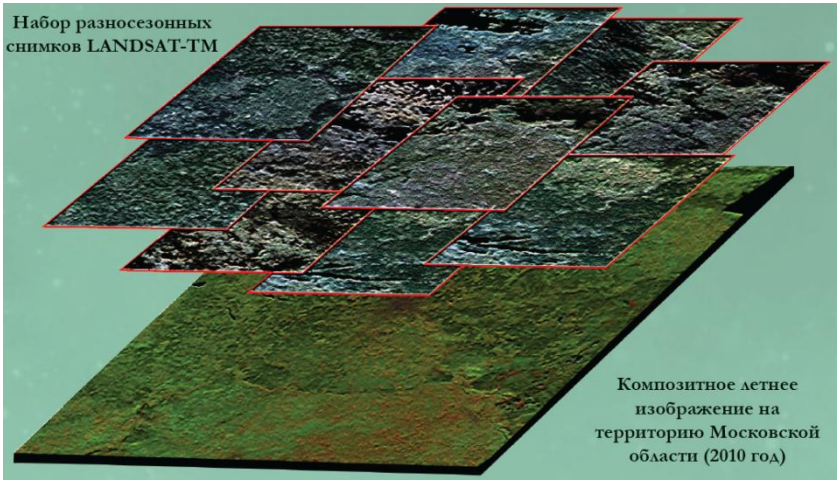


Рисунок 1 – Исходные материалы

Все изображения представлены в формате IMG ERDAS Imagine в проекции UTM/WGS84.

В пределах указанных выше гранул из архива Landsat-TM было отобрано 48 снимков оптимального качества на период с начала мая до конца сентября преимущественно 2010 и 2009 годов. Из них 9 снимков приходятся на май, 14 – на сентябрь и 25 – на летние месяцы. Часть майских снимков датируются 2007 годом по причине отсутствия качественных снимков более позднего периода. Это позволяет расширить временной диапазон наблюдения региона с начала мая по конец сентября. Фактически, подбираются изображения, в той или иной степени свободные от облачного покрова и охватывающие по времени весь вегетационный сезон. Все отобранные снимки подвергаются предварительной обработке – фильтрации облачности и атмосферной коррекции [5].

Безоблачное композитное изображение состоит из трех каналов, соответствующих красному (0,63-0,69 мкм) и двум интервалам инфракрасного диапазона (0,75-0,90 и 1,55-1,75 мкм). Данные каналы используются как наиболее информативная комбинация при распознавании основных типов наземных экосистем.

После предварительной обработки, из каждого отобранного изображения извлекаются эти каналы, которые затем упорядочиваются по времени – в соответствии с датой съемки, и объединяются отдельно в

многоканальные (многослойные) изображения – по три периода вегетационного сезона (весна, лето, осень). Важно отметить, что число слоев в каждом изображении соответствует числу отобранных сцен за период, при этом каждый слой соответствует определенной дате, соответственно покрытие территории на каждом слое различно. Обозначим эти изображения как «*исходные сезонные стеки*».

Кластеризация летней мозаики

Для летнего безоблачного композитного изображения выполняется неконтролируемая классификация методом ISODATA [6] с большим числом классов (15,000) в ERDAS Imagine. Результатом классификации является разбиение территории на множество зон (кластеров), однородных по спектральным яркостям в трехмерном пространстве красного и двух инфракрасных каналов. При этом предполагается, что с некой вероятностью пиксели этих зон могут встречаться по всему изображению. Это условие необходимо для последующего моделирования спектральных яркостей временных серий спутниковых изображений на всю территорию региона.

Моделирование многоканального изображения по разносезонным снимкам

Основная идея при моделировании поведения спектральных яркостей в зависимости от временного периода заключается в «переносе» значений спектральных яркостей наземных экосистем, взятых с каждого слоя исходного сезонного стека, на всю территорию региона, с использованием соотнесения этих показателей с кластерами композитного изображения. В качестве такого показателя рассматривается среднее значение яркости кластера (спектрально однородного участка земной поверхности) в канале спутникового изображения. На отдельно взятом слое исходного сезонного стека проводится вычисление среднего значения спектральной яркости пикселей в пределах каждого кластера. После чего, это среднее значение присваивается всем пикселям в пределах этого кластера на всем композитном изображении. То есть, достаточно, чтобы хотя бы один пиксель из каждого кластера приходился на территорию, покрытую значениями со снимка на данном слое. Данный алгоритм реализуется в виде графической модели ERDAS Imagine.

Полученные в результате этого многослойные изображения представляют собой сезонные стеки, где каждый слой соответствует состоянию территории на определенную дату в данном канале. Они соединяются поканально, и в результате получается три *временных стека* (по одному на канал), слои которых расположены в хронологическом порядке от начала мая до конца сентября, охватывают весь вегетационный сезон. Эти изображения, по сути, и являются *моделями поведения спек-*

тральных яркостей в зависимости от временного периода. Наглядно проиллюстрировать изменения значений спектральной яркости основных наземных экосистем в одном из каналов может спектральный профиль, который строится для любого пикселя изображения по значениям на каждом слое временного стека (рис.2).

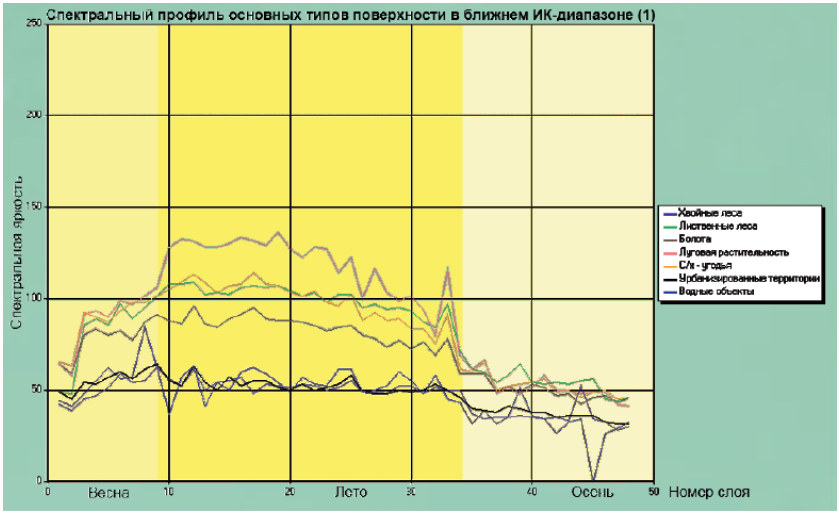


Рисунок 2 – Спектральный профиль основных типов поверхности в ближнем ИК-диапазоне

Анализ и обработка временных стеков

Выравнивание спектрального профиля

Неоднородность исходных данных, а также специфика моделирования временных стеков обуславливает наличие на спектральном профиле аномальных «выбросов» (резкое изменение яркости типа поверхности) и «провалов» (отсутствие значения яркости на слое). Причинами могут быть остаточная облачность или тени от облаков (неотфильтрованные участки изображения на краях облаков, или мелкая облачность меньше разрешения снимка), а также изменения в растительном покрове (пожар, вырубка и т.п.) в течение вегетационного сезона.

Эти флуктуации спектральной яркости могут вызывать в последствие ошибки при классификации, поэтому встает необходимость в процедуре сглаживания спектральных кривых. Она осуществляется методом скользящего окна с размерностью [1x7] для определения среднего значения и среднеквадратического отклонения спектральной яркости вдоль профиля спектральной кривой. Из анализа исключаются нулевые

значения, условно отнесенные к пропускам данных. Если значение анализируемого пикселя откланяется от среднего значения более чем на две величины СКО, то оно заменяется средним значением между двумя соседними слоями. Если значение анализируемого пикселя откланяется от среднего значения не более чем на две величины СКО, то оно остается неизменным:

$$R_i = \begin{cases} [\mu - 2\cdot\sigma] < R_i < [\mu + 2\cdot\sigma], R_i \\ \dfrac{(R_{i-1} + R_{i+1})}{2} \end{cases} \quad (1)$$

R_i – спектральная яркость пикселя в *i* слое стека;

μ_i – среднее значение пикселей в скользящем окне [1x7] профиля;

σ_i – среднеквадратическое отклонение яркостей пикселей в скользящем окне [1x7] профиля.

Данная процедура реализуется также в виде графической модели ERDAS Imagine. Она позволяет скорректировать явные флуктуации яркостей пиксела на смоделированных временных стеках (рис.3).

Осреднение спектрального профиля

Полученные временные профили подробно иллюстрируют динамику изменения спектральной яркости основных наземных экосистем региона. Данные изображения занимают относительно большие объемы памяти (десятки гигабайт), что делает проблематичным их автоматическую классификацию, в частности методом ISODATA. Кроме того, часть слоев может нести в себе зашумленную информацию, которая снижает точность классификации. Таким образом, перед классификацией, производится сокращение временного разрешение спектральных профилей за счет осреднения спектральных яркостей.

Процедура заключается в замене каждых шести последовательных яркостей в профиле на один, значения пикселей которого соответствуют среднему из этих шести измерений. Другими словами, выполняется сокращение пространства признаков (слоев) для каждого спектрального канала в течение вегетационного сезона (рис.4).

Окончательно все три спектральных диапазона (красный канал, БИК и средний ИК) объединяются в единое изображение. В случае с Московской областью было получено 18-тислойное изображение для тематической классификации – по 6 слоев на каждый из трех информативных каналов. Обозначим его как *«финальный стек»*. Примеры разносезонных финальных стеков, синтезированных по трем спектральным каналам, приводится на рисунке 5.

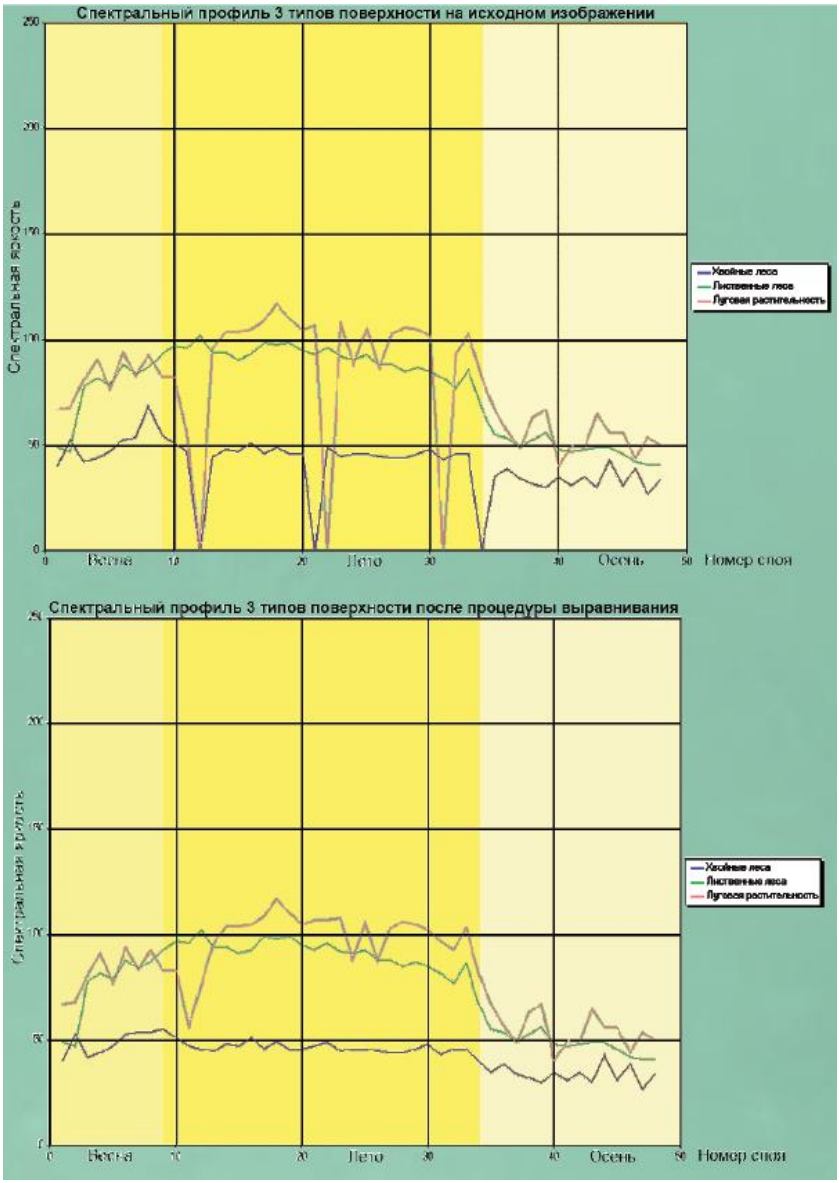


Рисунок 3 – Исходный и выровненный спектральные профили
для трех типов поверхности

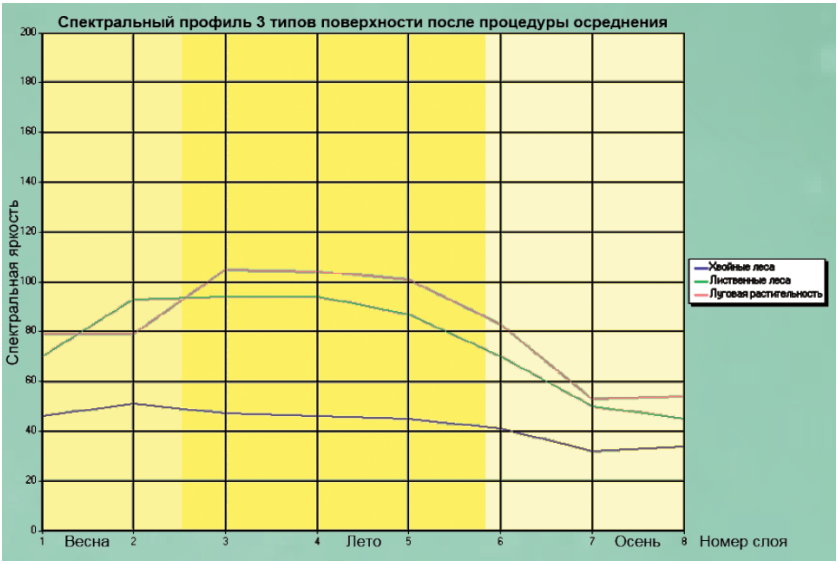


Рисунок 4 – Осредненный спектральный профиль для трех типов поверхности

Классификация изображения

На начальном этапе классификации финального стека выполняется стандартная неконтролируемая классификация алгоритмом ISODATA в ERDAS Imagine с последующим визуальным анализом и распределением кластеров по тематическим классам, соответствующим основным наземным экосистемам региона. Для Московской области рассматриваются следующие наземные экосистемы:

1). Темнохвойные леса
2). Светлохвойные леса
3). Лиственные леса
4). Смешанные леса с преобладанием хвойных пород
5). Смешанные леса
6). Смешанные леса с преобладанием лиственных пород
7). Болота
8). Луговая растительность
9). Сельскохозяйственные угодья
10). Открытая почва
11). Урбанизированные территории
12). Водные объекты

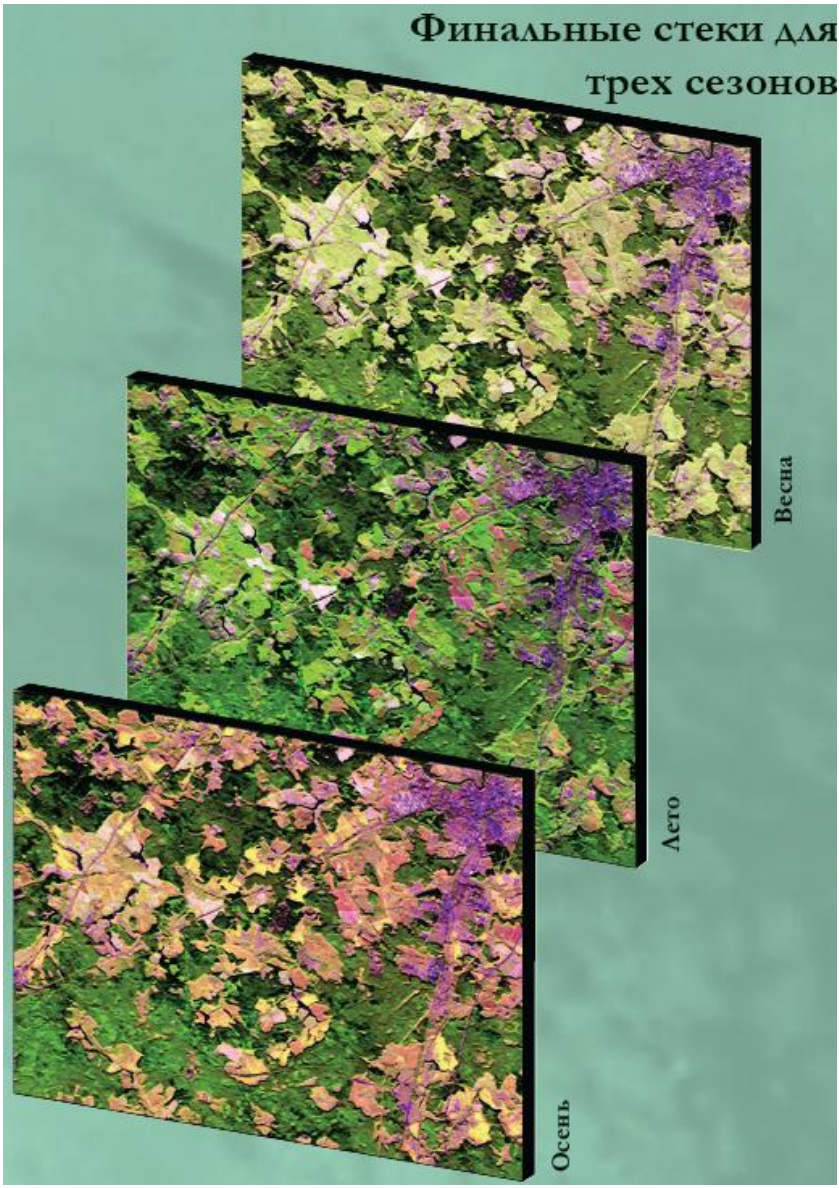


Рисунок 5 – Фрагменты псевдо-синтезированных изображений за весенний, летний и осенний периоды вегетационного сезона

Для определения состава классов использовалась легенда карты растительности России [7] , которая является продолжением научных работ, выполненных в рамках международного проекта GLC-2000 с целью создания карты наземных экосистем Северной Евразии [8].

Рисунок 6 – Блок-схема тематической обработки временных серий спутниковых изображений

С помощью карты растительности и другой дополнительной информации определяются тематически идентифицированные кластеры – *эталоны* – в качестве опорной информации для формирования обучающей выборки. На этом же этапе проводится количественная оценка в пикселе спутникового изображения доли площади, покрытой хвойными и лиственными породами. В качестве подхода используется метод декомпозиции спектральных смесей Mathieu [9], апробированный во многих исследованиях авторов по оценке различных характеристик лесов [10]. Эта информация используется для настройки порогов при определении соотношения хвойных и лиственных лесов в пространстве красного и ближнего инфракрасного каналов летних изображений.

Дополнительным источником информации при идентификации кластеров на этапе экспертного анализа также использовались межклассовые расстояния между средними значениями опорных спектров тематических классов и кластеров в многомерном пространстве. На основе этих расстояний определялись три ближайших тематических класса в

качестве кандидатов для идентификации кластеров на этапе экспертного анализа (реализовано в виде плагина к ERDAS Imagine).

В случае, когда результаты первичной кластеризации были неудовлетворительными, т.е. образовывались семантически смешанные кластеры (например, луга и с/х-угодья), то этап классификации повторялся с ограничением области кластерного анализа по отдельным участкам покрытых и непокрытых лесом территорий.

После того, как для всех классов наземных экосистем сформированы обучающие выборки на основе эталонных кластеров, проводится контролируемая классификация финального стека по методу наименьших расстояний (стандартная процедура ERDAS Imagine) и ее результат объединяется с результатами экспертного анализа «проблемных» семантически смешанных кластеров. Последовательность этапов классификации изображения для создания карты растительности по данным высокого разрешения приводится на рисунке 6.

Коррекция результатов классификации

Окончательно идентифицированное тематическое изображение растительности проходило этап дополнительных пространственных корректировок.

Поскольку часть объектов городской застройки, транспортной сети, а также гидрографии не могут быть корректно распознаны при кластерном анализе, то они используются в качестве принудительных масок при отнесении кластеров нелесным объектам. Для этого использовались картографические слои цифровых топографических карт крупного масштаба (1:100000 для Московской области). Кроме того, проводится ряд стандартных процедур по устранению ошибок первого и второго рода, возникающих при неконтролируемой классификации. В частности исключения из однородных областей тематического изображения одиночных пикселей (рис.7). На завершающем этапе выполняется преобразование полученной тематической карты в географическую систему координат относительно исходного композитного изображения. Все операции проводятся стандартными утилитами ERDAS Imagine.

Результатом проделанной работы является карта наземных экосистем Московской области по состоянию на 2010 год (рис.8).

Валидация карты наземных экосистем по картам лесоустройства

В качестве первичных данных для проверки точности классификации лесных классов при создании карты Московской области были использованы лесоустроительные планы на различные лесничества субъекта, объединенные в единую растровую карту, по состоянию на начало 2000-х годов. Планы представляют собой карты лесонасаждений, окрашенные по преобладающим породам в принятой для лесоустройства

цветовой палитре. Оценка точности проводится по методу случайных контрольных точек, с учетом различий в условных обозначениях на планах и на тематической карте растительности (рис.9).

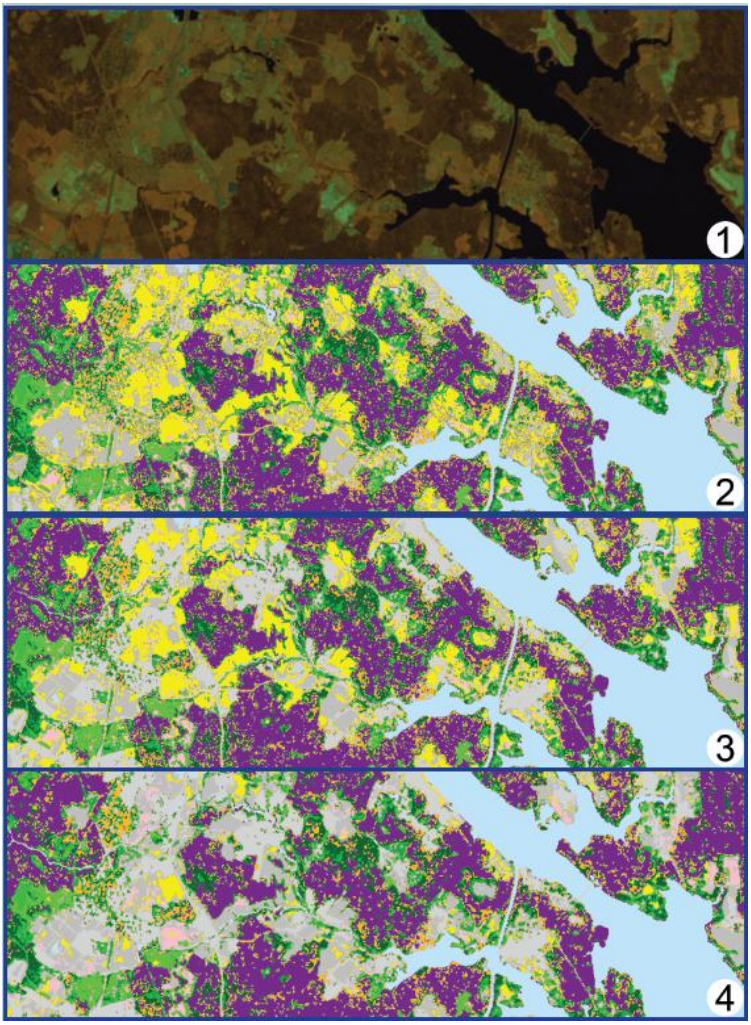


Рисунок 7 – Сравнение финального стека (1) и классифицированного тематического изображения (2) с результатами корректировок по топокартам (3) и устранения одиночных пикселей (4)

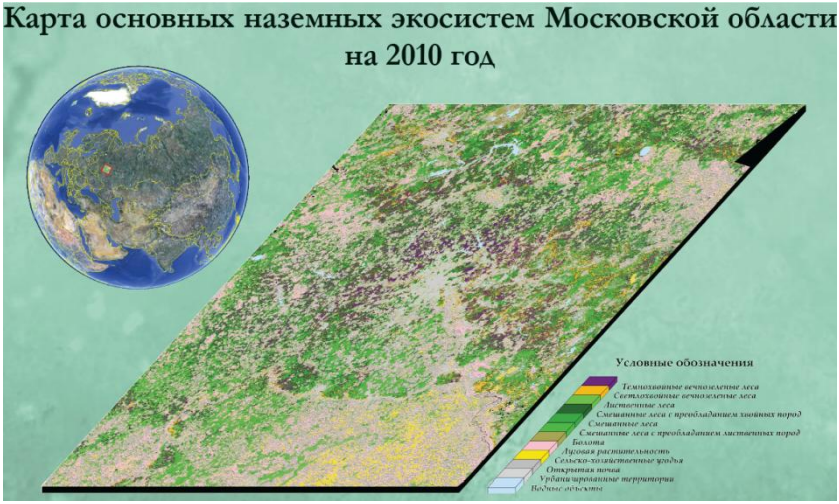


Рисунок 8 – Карта основных наземных экосистем Московской области по данным космической съемки Landsat-TM (30 м) 2010 года

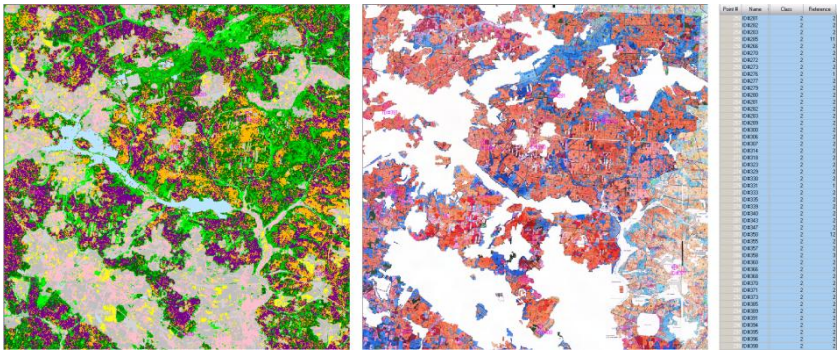


Рисунок 9 – Контроль точности классификации карты по лесоустроительным планам.

Поскольку структура легенды для этих планов несколько отлична от структуры классов, выделенных на карте наземных экосистем, проверка точности проводилась для следующих классов с определенными условиями:

1). Темнохвойные леса (1 класс) – соответствуют массивам ели на плане;

2). Светлохвойные леса (2) – соответствуют массивам сосны на плане;

3). Смешанные леса с преобладанием хвойных пород (4) – соответствуют массивам ели или сосны на плане;

4). Лиственные леса и смешанные леса с преобладанием лиственных пород (3 и 6) – соответствуют массивам любых лиственных пород на плане.

Проверку точности классификации для других классов по данным планам проводить не представляется возможным.

Анализ точности классификации проводится по средствам утилиты Accuracy Assessment в ERDAS Imagine путем набора случайных контрольных точек в пределах каждого проверяемого класса на карте и сравнения их с планом лесоустройства. Результаты анализа приведены в таблице 1.

Таблица 1 – Анализ точности классификации основных классов лесов на карте Московской области по сравнению с картой лесоустройства (%)

Класс на плане / Класс на карте	1	2	3	4	6	9	10	11	12	*Итого*
Темнохвойные (1)	**69,34**	24,66	3,33	0	0	0	1,33	1,33	0	*100*
Светлохвойные (2)	9,34	**78,00**	8,66	0	0	0	0,66	2,00	1,33	*100*
Лиственные (3)	8,06	1,61	**79,04**	0	0	1,61	9,68	0	0	*100*
С преобладанием хвойных (4)	0	0	0	**80,00**	18,00	0	1,34	0	0,66	*100*
С преобладанием лиственных (6)	7,96	2,27	0	0	**87,50**	0	2,27	0	0	*100*

Анализ значительного числа точек (600 площадок), дает от 70 до 87,5 % совпадений в зависимости от класса, и среднюю точность классификации 77,83% лесных классов. Полученный уровень точности является закономерным при сравнении карты наземных экосистем по спутниковым данным с лесоустроительными планами. Учитывая заметную разницу во времени (порядка 10 лет), некоторые несоответствия в классификации, а также определенную условность картографического изображения по сравнению с реальностью, данный результат можно признать удовлетворительным.

Выводы

Данные высокого пространственного разрешения Landsat являются на данный момент оптимальным и наиболее доступным решением для тематического картографирования наземных экосистем в региональном масштабе.

Предложенная в данной статье методика позволяет решать одну из главных проблем при использовании данных Landsat для тематической классификации – отсутствие однородного и непрерывного ряда разновременных данных в пределах вегетационного сезона на всю территорию картографирования.

Сочетание методов неконтролируемой и контролируемой классификации видится наиболее обоснованным и перспективным в плане автоматизации выбором при создании тематических карт наземных экосистем, который дает на выходе относительно высокие и стабильные показатели точности.

Список источников

1. Белова Е.С., Ершов Д.В. Методика создания безоблачных композитных изображений по спутниковым данным Landsat // Восьмая Всероссийская Открытая конференция «Современные проблемы дистанционного зондирования Земли из космоса» Москва, ИКИ РАН, 15-19 ноября 2010 г. Сборник тезисов конференции.

2. Liew, S.C.; Li, M.; Kwoh, L.K.; Chen, P.; Lim, H. 1998. Cloud-free multiscene mosaics of SPOT images. Proceedings of the 1998 International Geosciences and Remote Sensing Symposium. 2: 1083–1085

3. Helmer, E.H.; Ruefenacht, B. 2005. Cloud-free satellite image mosaics with regression trees and histogram matching. Photogrammetric Engineering and Remote Sensing. 71: 1079–1089

4. Martinuzzi, Sebastián; Gould, William A.; Ramos González, Olga M. 2006. Creating cloud-free Landsat ETM+ data sets in tropical landscapes: cloud and cloud-shadow removal. Gen. Tech. Rep. IITF-32. Rio Piedras, PR: U.S. Department of Agriculture, Forest Service, International Institute of Tropical Forestry. 12 p.

5. Белова Е.С., Ершов Д.В. Предварительная обработка временных серий изображений Landsat-TM/ETM+ при создании безоблачных изображений местности // Современные проблемы дистанционного зондирования Земли из космоса. Т8, №1, 2011, с. 73-82

6. Руководство пользователя: Erdas Imagine field guide, 2006, с.316

7. Барталев С.А., Егоров В.А., Ершов Д.В., Исаев А.С., Лупян Е.А., Плотников Д.Е., Уваров И.А. Спутниковое картографирование растительного покрова России по данным спектрорадиометра MODIS // Современные проблемы дистанционного зондирования Земли из космоса: Физические основы, методы и технологии мониторинга окружающей среды, потенциально опасных явлений и объектов. Сборник научных статей. 2011. Т.8. Том 8. Номер 4. – М.: ООО «ДоМира», 2011. - С. 285-302

8. S.A. BARTALEV, A.S. BELWARD, D.V. ERCHOV, and A.S. ISAEV, 2002, A new SPOT4-VEGETATION derived Land Cover Map of Northern Eurasia, International Journal of Remote Sensing, Reference No: RES 103841 and associated cover (reference RES 107158).

9. Mathie S., Berthod M., Leymarie P. Determination of proportions and entropy of land use mixing in pixels of a multispectral satellite image // IEEE Transactions on geoscience and remote sensing symposium (IGARSS 94), 1994. - p. 1154 - 1156

10. Барталев С.А., Ершов Д.В., Исаев А.С. Оценка дефолиации лесов по многоспектральным спутниковым изображениям методом декомпозиции спектральных смесей // «Исследование Земли из космоса», 1999, №4, С. 76-86

11. Landsat Glovis USGS archive [Электронный ресурс]: http://glovis.usgs.gov/, режим доступа: свободный.

12. Сервис ВЕГА: спутниковый сервис анализа вегетации [Электронный ресурс]: http://vega.smislab.ru/, режим доступа: регистрация.

Выделение гармонической информации из аудиозаписей

Н. Глазырин[1], А. Клепинин[2]

[1] nglazyrin@gmail.com, [2] alexandr.klepinin@usu.ru

Уральский федеральный университет имени первого Президента России Б. Н. Ельцина, Екатеринбург, Россия

Аннотация. Описывается подход для определения последовательности аккордов, содержащихся в музыкальной звукозаписи, и тональности этой музыкальной композиции.

Ключевые слова: музыкальный информационный поиск; определение аккордов.

Введение

Обычно гармонической называется информация о содержащихся в музыкальной композиции аккордах, включая указание основной ноты и типа аккорда, а также временны́х позиций начала и конца его звучания. Помимо этого к гармонической относится информация о тональности композиции в целом или отдельных её сегментов. Эти знания могут быть полезны при решении таких задач, как автоматическое разделение композиции на сегменты, нахождение похожих по содержимому композиций.

Начиная с 2008 года в рамках ежегодной кампании по оценке алгоритмов музыкального информационного поиска (Music Information Retrieval Evaluation eXchange – MIREX) [3] проводится сравнение алгоритмов, определяющих последовательность аккордов в звукозаписи. Для

Игнатов Д. И., Яворский Р. Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16–18 марта, 2012

оценки качества распознавания используются 12 альбомов The Beatles, два альбома Queen и один альбом Zweieck. Все песни с них были предварительно распознаны и размечены вручную и до недавнего времени других источников для сравнения качества алгоритмов не было [9].

В рамках MIREX для оценки качества распознавания используются метрики chord overlap ratio и chord weighted average overlap ratio, не описанные в публикациях. Однако исходный код для их вычисления свободно доступен на [2]. В литературе встречаются и другие метрики, например [10], [12], [13].

В данной работе представлен подход, показывающий неплохие результаты на тестах MIREX, но при этом не предполагающий использования каких-либо сложных техник, связанных с вероятностным анализом или машинным обучением. Хорошее качество распознавания обусловлено грамотным сочетанием уже известных техник обработки сигнала и музыкальной теории.

Как правило, переход от «низкоуровневой» информации о звуке (обычно представленной в звуковом файле как набор значений амплитуды звуковой волны в последовательные моменты времени) к «высокоуровневой» информации о соответствующей последовательности аккордов делается в несколько этапов. Общим для всех систем определения аккордов в музыке является наличие промежуточных представлений звукового файла сначала в виде спектрограммы (последовательности спектров звука на коротких последовательных участках), а затем в виде последовательности векторов спектральных характеристик. Каждый из этих векторов, по существу, является сжатым представлением спектра на соответствующем участке спектрограммы. Именно на основе последовательности векторов спектральных характеристик и производится определение последовательности звучащих аккордов. На этом этапе часто применяются различные алгоритмы машинного обучения, такие как скрытые марковские модели, нейронные сети и другие. Предварительная тренировка, характерная для таких подходов, является и достоинством (позволяет повысить качество распознавания), и недостатком (качество распознавания ставится в зависимость от качества обучения). Поэтому идея разработки подхода, обеспечивающего высокое качество распознавания без использования техник обучения, выглядит вполне разумной.

Описание предлагаемого подхода

Анализ звукового файла будет состоять из нескольких шагов.

Прежде всего, для получения спектра звукового файла будем использовать constant-Q преобразование [5], являющееся аналогом дискретного преобразования Фурье. Его основное отличие от преобразования Фурье

в том, что частотные компоненты расположены не равномерно, а логарифмически, в соответствии с частотами нот в традиционном звукоряде. Частота k-й компоненты задается формулой

$$f_k = 2^{k/b} f_{min}$$

где b — количество компонент в одной октаве и f_{min} — частота наименьшей из компонент. При $b = 12$ частоты компонент в точности соответствуют частотам нот традиционного звукоряда. А параметр f_{min} выбирается так, чтобы учесть информацию, находящуюся в низких частотах, но при этом сохранить время обработки одного звукового файла в разумных пределах (вычисление constant-Q преобразования для низкочастотных компонент спектра требует существенно больше ресурсов, чем для высокочастотных).

При обработке сигнала стоит учитывать, что музыкальные инструменты, участвующие в композиции, могут быть настроены на базовую частоту, отличную от 440 Гц (нота *ля* первой октавы). Поэтому до вычисления спектра делается попытка определить базовую частоту конкретной настройки инструментов (подстроиться под композицию). Это делается при помощи constant-Q преобразования с параметрами $f_{min} = 440$ Гц, $b = 120$, охватывающего 4 октавы. Звуковой файл при этом делится на последовательные фрагменты, на каждом из них делается преобразование, и сохраняется номер компоненты спектра, в которой достигается наибольшее значение (пик) на данном фрагменте. Известно, что частоты нот звукоряда получаются путем умножения базовой частоты (в данном случае ей является частота настройки) на $2^{k/12}$. А при наличии 120 частотных компонент на октаву можно уловить отклонения в частоте настройки, превышающие 1/10 полутона (1 полутон соответствует расстоянию между двумя соседними нотами звукоряда).

Можно сделать предположение, что выявленные спектральные пики должны соответствовать частотам нот, и при этом между соседними нотами должно быть расстояние в 10 компонент спектра constant-Q преобразования. Таким образом, можно построить гистограмму позиций обнаруженных спектральных пиков по всему файлу (в ней будет 120 компонент), разделить её на участки по 10 последовательных компонент и просуммировать результаты по всем таким участкам. В полученной гистограмме из 10 компонент позиция пика укажет на отклонение частоты настройки от стандартной (440 Гц) в пределах от $-1/2$ полутона до $+1/2$ полутона (что соответствует частотам от 427,5 Гц до 452,9 Гц) с шагом в 1/10 полутона. Заметим также, что отклонение частоты настройки от стандартной на 1 полутон соответствует простому сдвигу наименований нот звукоряда на 1 шаг, при этом их частоты остаются неизменными. Так

что ограничение диапазона подстройки от $-1/2$ полутона до $+1/2$ полутона выглядит вполне естественным и разумным. Полученная частота настройки затем используется для выбора f_{min} и частот компонентов для основного constant-Q преобразования.

При спектральном анализе звукового файла он разбивается на отдельные фрагменты. При этом имеет смысл позиции начала фрагментов выбрать в соответствии с темпом композиции, поскольку моменты начала звучания отдельных нот обычно соответствуют моментам начала метрических долей композиции. Поэтому при размещении анализируемых фрагментов в точках начала метрических долей композиции с большей вероятностью получится отразить в спектре точное звучание музыкальных инструментов. Для определения моментов начала метрических долей используется библиотека BeatRoot. Возвращаемый ей список корректируется для получения последовательности моментов времени, в которой разница между любыми двумя соседними элементами одинакова (поскольку обычно основная ритмическая сетка композиции остается равномерной во времени). После этого строится последовательность, в которой значения расставлены в 8 раз чаще. Тем самым, на каждую метрическую долю приходится 8 фрагментов, что дает возможность в дальнейшем усреднить значения по всем этим фрагментам для большей устойчивости к шумам.

Для получения спектрограммы используется constant-Q преобразование со следующими параметрами: f_{min} соответствует ноте до малой октавы с учетом базовой частоты настройки (130,8 Гц при частоте настройки 440 Гц), $b = 60$ (каждой ноте звукоряда соответствует 5 компонент получаемого спектра), охват — 4 октавы (частотный диапазон до 1975 Гц; в него попадает бóльшая часть значимой информации в спектре музыкальных композиций). Фрагменты располагаются в соответствии с полученной на предыдущем шаге последовательностью моментов времени. Спектрограмма, по сути, является двумерным массивом, столбцы которого соответствуют спектрам отдельных фрагментов.

Далее над спектрограммой производим несколько преобразований, нацеленных на уменьшение шума и на выделение участков, отражающих звучание музыкальных инструментов без учета тембра. На рис. 1 показаны последовательные состояния спектрограммы для первых 23 секунд композиции The Beatles — «Yellow Submarine» в процессе её обработки.

Сначала применяем фильтр скользящего среднего с шириной окна w_1, что соответствует усреднению по некоторому временному промежутку. Это позволяет за счет очень частого расположения позиций начала фрагментов сгладить спектр и снизить чувствительность к шумам.

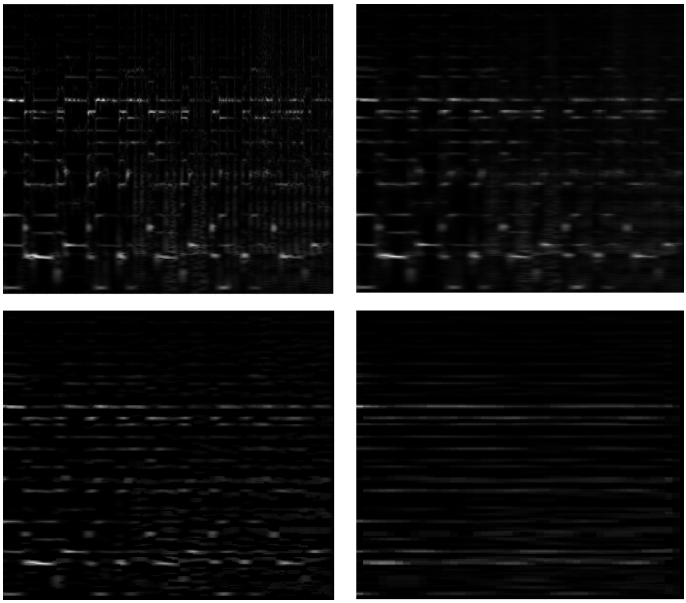


Рис. 1. Слева направо сверху вниз: исходный спектр; после первого сглаживания; после применения оператора Прюитт; после сжатия по горизонтали и последнего сглаживания.

К полученной сглаженной по горизонтали спектрограмме применяем «растянутый по вертикали» аналог оператора Прюитт [1] с одним ядром размера 9×3: первые 3 строки состоят из «-1», следующие 3 строки — из нулей, последние 3 строки — из «+1». Размерность матрицы подобрана таким образом, чтобы наилучшим образом выделить горизонтальные линии на спектрограмме при данных параметрах constant-Q преобразования ($b = 60$). Для каждого элемента спектрограммы вычисляется поэлементное произведение прямоугольника размером 9×3 с центром в данном элементе на описанное ядро. Если полученный результат больше 0, то значение элемента сохраняется, иначе — элемент обнуляется. После этого из спектрограммы удаляются все горизонтальные отрезки, состоящие из менее чем r идущих подряд ненулевых элементов. Всё это позволяет удалить ненужные вертикальные линии на спектрограмме, сохранив горизонтальные, которые и представляют наибольший интерес, поскольку соответствуют музыкальным инструментам, воспроизводящим определенные ноты, в то время как вертикальные линии обычно соответствуют шумовым и ударным инструментам.

Далее спектрограмма разбивается на последовательности столбцов длины 8 (каждый такой фрагмент соответствует одной метрической доле), и вместо каждого фрагмента сохраняется один столбец, равный среднему арифметическому всех 8 столбцов фрагмента (спектрограмма «сжимается» по горизонтали в 8 раз). Затем вновь применяем фильтр скользящего среднего с шириной окна w_2. И по полученной в итоге спектрограмме вычисляется последовательность векторов профилей тональных классов (pitch class profiles, PCP, [7]).

Для определения звучащего на каждом фрагменте звукового файла аккорда применяется техника сопоставления вектора профиля тональных классов на этом фрагменте с шаблонными векторами, каждый из которых соответствует одному из известных аккордов. В качестве результата при этом выбирается вектор, евклидово расстояние до которого является наименьшим. В предлагаемом методе используются шаблоны для 12 мажорных и 12 минорных аккордов, всего 24 шаблона, сгенерированных на основе сведений из теории музыки. Фактически, шаблон является 12-мерным вектором, в котором только компоненты, соответствующие нотам, входящим в аккорд, отличны от нуля. Фрагменты, на которых не звучит музыка, в отдельную категорию не выделяются, на каждом фрагменте делается попытка определить звучащий аккорд. Данное сопоставление проводится для всей последовательности векторов дважды: для определения тональности музыкальной композиции и для определения последовательности аккордов в ней с учетом тональности.

После первого сопоставления по полученной последовательности аккордов определяются 7 тональных классов, наиболее часто встречающихся в аккордах в этой последовательности. Из теории музыки известно, что любая из мажорных и минорных тональностей задается 7 ступенями, каждая из которых соответствует одному тональному классу. Если полученные 7 наиболее частых тональных классов задают тональность, то она используется в дальнейшем. Выявление тональности позволяет ограничить аккордный «алфавит» всего 6-ю основными аккордами, возможными в данной тональности. И при проведении второго сопоставления используются только эти 6 аккордов. В случае, если 7 наиболее частых тональных классов не соответствуют ни одной из мажорных/минорных тональностей, тональность считается неопределенной, и поэтому второе (уточняющее) сопоставление не производится. Результатом является последовательность аккордов с указанием моментов начала и конца звучания каждого из них.

Анализ результатов

Были проведены эксперименты на 12 альбомах The Beatles. Для каждой композиции была вычислена метрика overlap score [13]; она показывает, для какой доли фрагментов композиции были точно указаны аккорды, звучащие на них. При этом, в соответствии с правилами оценки MIREX 2011, 2 аккорда считались совпадающими, если они имели общее трезвучие (при этом, например, аккорды `C:maj` и `C:maj7`, а также `E:min` и `C:maj7` считаются совпадающими). По окончании были вычислены метрики chord overlap ratio и chord weighted average overlap ratio в соответствии с [2] для оценки качества работы метода на всём объеме данных. Наилучшие полученные значения: chord overlap ratio — 0.5222; chord weighted average overlap ratio — 0.5256.

Основными параметрами метода являются значения размеров окна для фильтров скользящего среднего w_1, w_2, а также минимально допустимая длина горизонтального отрезка спектрограммы, состоящего из ненулевых элементов, r. Также можно изменять поведение метода, меняя алгоритмы определения базовой частоты, тональности, предварительной очистки спектра. Зависимость качества распознавания от величин w_1, w_2 представлена в табл. 1.

Табл. 1. Зависимость качества распознавания от w_1 и w_2

	overlap ratio			weighted average OR		
$w_1 \setminus w_2$	1	5	9	1	5	9
1	0.4718	0.5098	0.4884	0.4738	0.5134	0.4919
5	0.4800	0.5209	0.4969	0.4812	0.5241	0.5003
9	0.4832	0.5218	0.4961	0.4847	0.5253	0.4997
17	0.4903	**0.5222**	0.4959	0.4918	**0.5256**	0.4993
33	0.5129	0.5177	0.4940	0.5155	0.5214	0.4977

Видно, что наилучшие результаты достигаются при $w_1 = 17, w_2 = = 5$. Это соответствует размеру окна в 2 метрические доли при первом сглаживании и в 5 метрических долей при втором сглаживании. При этом было выбрано значение $r = 20$, соответствующее 2,5 метрическим долям.

Эти значения были получены с отключенным модулем определения тональности. С его использованием оба значения имели наибольшую величину около 0.51. Данный результат свидетельствует о недостаточно хорошем методе определения тональности. Ошибочное её определение ограничивает «алфавит» аккордов неверными аккордами и ведет к очень низкому результату на данной композиции. Кроме того, встречаются

композиции, не принадлежащие целиком к одной тональности, для них ошибки также будут неизбежны. В свою очередь, без заданной тональности достаточно часты ошибки в выборе между мажорным или минорным вариантами аккорда с одной и той же основной нотой.

При отключенном модуле определения базовой частоты (она принималась равной 440 Гц для всех композиций) были получены следующие значения метрик: 0.4914 и 0.4896 соответственно. Таким образом, даже простой алгоритм определения базовой частоты настройки музыкальных инструментов повышает качество распознавания. Для записей *The Beatles* это вполне естественно, поскольку они были сделаны на аналоговом оборудовании, подвергались оцифровке с магнитных лент и дополнительной обработке. Для большей части современной музыки, изначально записываемой в цифровом виде, данный шаг будет практически бесполезен.

Альтернативой описанной выше процедуре фильтрации спектра может быть «обеление» спектра, описанное в [11]. Несмотря на то, что получаемый при этом спектр оказывается чище (имеет существенно меньше ненулевых элементов), его использование не дало улучшения значений метрик: 0.4997 и 0.5029 соответственно. Возможной причиной может быть то, что в процессе «обеления» сильнее подавляются менее выраженные гармоники и не учитывается наличие горизонтальных линий на спектрограмме, соответствующих звучанию музыкальных инструментов.

Существенного улучшения качества распознавания можно ожидать, если модифицировать этап сопоставления вектора спектральных характеристик с шаблонными векторами, соответствующими аккордам. В предлагаемом методе выбирается только один из шаблонных векторов, евклидово расстояние до которого является наименьшим. При этом отбрасываются остальные шаблонные векторы, среди которых может оказаться вектор, соответствующий аккорду, звучащему на самом деле в данный момент. Поэтому необходимо учитывать также векторы спектральных характеристик, соответствующие другим моментам времени.

Как раз с этой целью обычно применяются методы машинного обучения, такие как скрытые марковские модели. На каждом шаге они оценивают вероятность смены аккорда в данный момент, используя информацию в предыдущие моменты времени. Также здесь можно использовать теорию гармонии, структуру музыкальной композиции, как это делается, например, в [6] или [15]. В этих публикациях описываются 2 из алгоритмов, участвовавших в задаче *MIREX 2011: Audio Chord Description* [4]. Лучшие из участвовавших в данной задаче алгоритмов имеют значения метрик chord overlap ratio и chord weighted average overlap ratio выше

0.80. Таких результатов удается добиться за счет модификаций именно на данном шаге.

Как следствие, дальнейшая работа будет сосредоточена в основном в направлении улучшения сопоставления вектора тональных характеристик с шаблонами аккордов. В том числе требует улучшения алгоритм определения тональности композиции. Использование информации о тональности является первым шагом к применению теории гармонии. Также необходимо учитывать при анализе не отдельные короткие участки композиции, а их последовательности. Отметим, что, несмотря на наличие ошибок распознавания, с данным подходом удается достичь неплохих результатов распознавания без привлечения более сложных алгоритмов, таких как алгоритмы машинного обучения, а также сохранить время обработки одной композиции в разумных пределах (порядка 1 минуты).

В заключение также приведем несколько ссылок на важные работы в рассматриваемой области. Первой из них является диссертация Э. Гомез [8], где детально рассматривается система для определения последовательностей аккордов в музыкальных звукозаписях. Автор проводит анализ применяемых на каждом шаге методов и их сравнение с аналогами. Хорошая сравнительная статья [14], в которой рассматриваются несколько различных подходов, вышла в 2007 году. Из числа более современных можно выделить работу [13]. Описания наилучших на данный момент алгоритмов можно найти на сайте MIREX [3]. Все эти источники позволяют лучше сориентироваться в рассматриваемой задаче и применяемых методах.

Список источников

1. Фисенко Т.Ю. Фисенко В.Т., editor. *Компьютерная обработка и распознавание изображений: учебное пособие.* СПбГУ ИТМО, СПб, 2008.
2. ChordEvaluator. http://nemadiy.googlecode.com/svn-history/r2699/analytics/trunk/src/main/java/org/imirsel/nema/analytics/evaluation/chord/ChordEvaluator.java, 2012. [Online; дата обращения 16 февраля 2012 г.].
3. MIREX. http://www.music-ir.org/mirex/wiki/, 2012. [Online; дата обращения 16 февраля 2012 г.].
4. MIREX 2011: Audio Chord Detection. http://nema.lis.illinois.edu/nema_out/mirex2011/results/ace/summary.html, 2012. [Online; дата обращения 16 февраля 2012 г.].

5. J.C. Brown. Calculation of a constant q spectral transform. 89(1):425–434, 1991.
6. Taemin Cho and Juan P. Bello A feature smoothing method for chord recognition using recurrence plots (for MIREX2011 submissions). In *ISMIR* 2010.
7. Takuya Fujishima. Realtime chord recognition of musical sound: a system using common lisp music. *Proc. ICMC, 1999*, pages 464–467, 1999.
8. Emilia Gómez. *Tonal Description of Music Audio Signals*. PhD thesis, University Pompeu Fabra, Barcelona, Spain, July 2006.
9. C. Harte, M. Sandler, S. Abdallah, and E. Gómez. Symbolic representation of musical chords a proposed syntax for text annotations. 2005.
10. Maksim Khadkevich and Maurizio Omologo. Use of hidden markov models and factored language models for automatic chord recognition. In *ISMIR*, pages 561–566, 2009.
11. Anssi Klapuri. Multiple fundamental frequency estimation by summing harmonic amplitudes. In *ISMIR*, pages 216–221, 2006.
12. Matthias Mauch, Katy Noland, and Simon Dixon. Using musical structure to enhance automatic chord transcription. *Information Retrieval*, (Ismir):231–236, 2009.
13. L. Oudre, Y. Grenier, and C. Févotte. Chord recognition using measures of fit, chord templates and filtering methods. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 9–12, New York, USA, 2009.
14. Hélène Papadopoulos and Geoffroy Peeters. Large-scale study of chord estimation algorithms based on chroma representation and HMM. In *Proc. of the International Workshop on Content-Based Multimedia Indexing (CBMI)*, Bordeaux, 2007.
15. Thomas Rocher, Matthias Robine, Pierre Hanna and Darrell Conklin. Concurrent estimation of chords and keys from audio. In *ISMIR*, pages 141-146, 2010.

Кластеризация текстовых данных с помощью модифицированного генетического алгоритма

Д. Глушкова

daria_glushkova@mail.ru

Челябинский государственный университет, Челябинск, Россия

Аннотация. В данной работе для кластеризации текстовых документов предлагается использование модификации классического генетического алгоритма. Дополнительно к классическим операторам были разработаны новые операторы скрещивания и мутации, направленные на увеличение значений функции приспособленности хромосом новой популяции.

Ключевые слова: кластеризация текстовых документов; эволюционные вычисления; генетические алгоритмы.

Введение

В последние десятилетия наблюдается рост интереса к новому направлению в обработке информации — интеллектуальному анализу данных (Data Mining). В предлагаемой работе рассматривается частная задача интеллектуального анализа данных — задача кластеризации текстовых документов, известная также как задача автоматической группировки объектов или классификации без учителя. Кластерный анализ широко используется для улучшения точности и полноты в системах информационного поиска [1–3], для обнаружения закономерностей в данных и определения тематики документов [4], при решении задач классификации текстовых данных [5–8].

Игнатов Д. И., Яворский Р. Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16–18 марта, 2012

В данной работе задача кластеризации текстовых документов сводится к задаче целочисленного программирования, которая решается с помощью классического генетического алгоритма с модифицированными операторами скрещивания и мутации.

Постановка задачи кластеризации

Задача кластеризации состоит в выявлении групп семантически похожих документов среди заданного фиксированного множества документов.

При кластеризации текстовых документов применяются разные методы и алгоритмы [9–11]. В любом методе кластеризации данных можно выделить следующие этапы, первые три из которых являются общими для всех алгоритмов кластеризации текстовых данных.

Предварительная обработка данных.

1) Удаление «стоп-слов». Под «стоп-словами» понимаем слова, не оказывающие влияния на тематику документа, например, артикли, союзы и предлоги.
2) Стемминг. Данная процедура заключается в выделении значимой части слова с помощью отсечения суффиксов и окончаний.
3) Преобразование всех символов к верхнему или нижнему регистру.

Выбор модели представления данных. В качестве модели представления данных была выбрана модель TF IDF (Term Frequency Inverse Document Frequency). Согласно данной модели каждый документ D_i представляется в виде взвешенного вектора $D_i = (w_{ij}, \ldots, w_{im})$, $i = = 1, \ldots, n$, $j = 1, \ldots, m$, где n — число документов, m — число слов в наборе документов $D = (D_1, \ldots, D_n)$. Здесь w_{ij} — вес слова j в документе D_i, который определяется по формуле: $w_{ij} = f_{ij} \log_2(\frac{n}{n_j})$, где n_j — число документов, в которых появляется слово j; f_{ij} — функция частоты появления слова j в документе D_i, определенная формулой $f_{ij} = m_{ij}/m$; m_{ij} — число появлений слова j в документе D_i.

Выбор меры подобия. Меру подобия между документами можно определить, используя любую метрику. В работе [12] было произведено сравнение эффективности алгоритмов кластеризации, основанных на различных мерах подобия. Наиболее эффективным оказалось использование косинусной метрики и метрики Жаккара. В данной работе в качестве меры подобия используется косинусная метрика.

Алгоритм кластеризации. В качестве алгоритма кластеризации был выбран модифицированный классический генетический алгоритм. В дополнение к классическим операторам скрещивания и мутации [13,14] были введены новые. Генетические алгоритмы — это процедуры поиска, основанные на механизмах естественного отбора и наследования. Данные алгоритмы отличаются от других оптимизационных и поисковых процедур следующим:

1) работают в основном не с параметрами задачи, а с закодированным множеством параметров;
2) осуществляют поиск не путем улучшения одного решения, а путем использования сразу нескольких альтернатив на заданном множестве решений;
3) использует целевую функцию, а не ее различные приращения для оценки качества принятия решений;
4) применяют вероятностные, а не детерминированные правила выбора.

Благодаря перечисленным свойствам генетические алгоритмы широко применяются при решении различных задач оптимизации [13].

Классический генетический алгоритм состоит из следующих шагов.

1) Кодирование решений в виде хромосом. В качестве способа кодирования хромосом было выбрано кодирование по номеру кластера. Данное кодирование представляет вариант кластеризации n объектов в виде строки из n чисел, в которой i-ое число обозначает номер группы i-го объекта.
2) Инициализация, или выбор исходной популяции хромосом. Генерация начальной популяции осуществлялась случайным образом. При генерации популяции таким способом необходимо делать проверку хромосом на корректность и нечувствительность к контексту. Корректность означает невозможность образования потомков с меньшим числом кластеров. Нечувствительность к контексту проявляется, когда одно и то же группировочное решение кодируется разными последовательностями.
3) Оценка приспособленности хромосом в популяции. Оценивание приспособленности хромосомы состоит в расчете функции приспособленности. В качестве данной функции мы взяли функцию приспособленности, описанную в работе [15]. Чем больше значение этой функции, тем выше «качество» хромосомы.
4) Проверка условия остановки алгоритма. Число итераций, необходимых для выполнения генетическим алгоритмом, было найдено экспериментально.

5) Селекция хромосом. Селекция хромосом заключается в выборе (по рассчитанным на третьем шаге значениям функции приспособленности) тех хромосом, которые будут участвовать в создании потомков для следующей популяции. В качестве алгоритма селекции был выбран турнирный метод. Исследования подтверждают, что данный метод является наиболее эффективным и не требует производить масштабирование функции приспособленности [13,14].

6) Применение генетических операторов (скрещивание, мутация). Дополнительно к классическим операторам были разработаны новые операторы скрещивания и мутации, направленные на увеличение значений функции приспособленности хромосом новой популяции. Алгоритм работы модифицированного оператора скрещивания состоит из 5 шагов.

   a) Разбиваем родительский пул из m хромосом на 2 равные множества: $P_1 = \{p_{11}, \ldots, p_{1\frac{m}{2}}\}$ и $P_2 = \{p_{21}, \ldots, p_{2\frac{m}{2}}\}$

   b) Выбираем из каждого множества P_1 и P_2 по одному элементу и принимаем их в качестве родительских. Пусть выбрали элементы p_{11} и p_{21}. Первоначально дочерняя хромосома полностью совпадает с первой родительской хромосомой p_{11}.

   c) Во второй родительской хромосоме p_{21} выбираем группу G_p с наибольшим значением средней меры подобия.

   d) В дочерней хромосоме с вероятностью p_{ch} выбирается группа G_{ch}.

      а) $p_{ch} = 0{,}6$. Выбирается группа со значением центроида наиболее близким к центроиду группы G_p, выбранной на третьем шаге.

      б) $p_{ch} = 0{,}36$. Выбираем группу, имеющую наибольшее количество общих элементов с группой G_p.

      в) $p_{ch} = 0{,}02$. Выбираем группу с наименьшим значением средней меры сходства.

      г) $p_{ch} = 0{,}02$. Группа выбирается случайным образом.

   e) Перестройка элементов в дочерней хромосоме. Элементы, находящиеся в группе G_p, но не находящиеся в группе G_{ch}, т. е. элементы $\in G_p \backslash G_p \cap G_{ch}$, добавляются в G_{ch}. Элементы $\in G_{ch} \backslash G_p \cap G_{ch}$ извлекаются из G_{ch} и размещаются в других группах. Группа, в которую добавляется элемент, выбирается с вероятностью $p_{ch}^{'}$ следующим образом:

1) $p_{ch}^{'} = 0{,}95$. Среди двух групп в дочерней хромосоме наиболее близких к группе G_p в родительской хромосоме выбирается группа со значением центроида наиболее близким к добавляемому элементу.

2) $p_{ch}^{'} = 0{,}05$. Группа выбирается случайным образом.

Необходимые значения вероятностей p_{ch} и $p_{ch}^{'}$ определялись экспериментально. Для фиксированного числа кластеризируемых документов выбираем значения вероятностей p_{ch} и $p_{ch}^{'}$, при которых энтропийный критерий и критерий, основанный на значении средней меры сходства, дают наилучший результат. Изменяя состав документов, данную процедуру повторяем 100 раз. В качестве значений искомых вероятностей p_{ch} и $p_{ch}^{'}$ берем их среднее.

Для получения второй дочерней хромосомы родительские хромосомы p_{11} и p_{21} требуется поменять местами.

Модифицированный оператор мутации работает следующим образом:

a) Осуществляем поиск групп $G_{\max}$ и $G_{\min}$ с наибольшим и наименьшим значением средней меры сходства соответственно.

b) В группе $G_{\max}$ ищем элемент, максимально удаленный от центроида группы $G_{\max}$.

c) Удаляем найденный элемент из группы $G_{\max}$ и добавляем его в группу $G_{\min}$.

Введенный оператор мутации применяется к хромосоме с вероятностью $p_m \in [0, \frac{3}{10}]$.

7) Формирование новой популяции.

8) Выбор наилучшей хромосомы.

Шаги 3–7 повторяются, пока не выполнится условие остановки алгоритма.

Тестирование

Тестирование практической реализации данного метода производилось на базе документов «Reuters 21578» [16]. Было произведено сравнение работы метода кластеризации, основанного на модифицированном генетическом алгоритме, с алгоритмом кластеризации методом k-средних и алгоритмом кластеризации, в основе которого лежал классический генетический алгоритм. В качестве критериев, определяющих

качество разбиения, были выбраны энтропийный критерий и критерий, основанный на значении средней меры подобия [17].

Энтропийный критерий. Энтропия известна как численное выражение неупорядоченности системы. Энтропия разбиения достигает минимума при наибольшей упорядоченности в системе (в случае четкого разбиения энтропия равна нулю). То есть чем больше степень принадлежности элемента одному кластеру (и меньше степень принадлежности всем остальным кластерам), тем меньше значение энтропии и тем более качественно выполнена кластеризация.

Пусть i — количество категорий в эталонном разбиении, p_{ij} — вероятность того, что документ группы j попадет в группу i. Вероятность рассчитывается по формуле: $p_{ij} = n_{ij}/n_j$, где n_{ij} — число документов группы i, попавших в группу j, n_j — общее число документов группы j. Для каждой категории i эталонного разбиения определяем значение вероятности p_{ij}.

Энтропия для каждой группы рассчитывается по формуле

$$H_j = -\sum_i p_{ij} \log(p_{ij}).$$

Тогда значение энтропии для конечного разбиения определяется:

$$H = \frac{\sum_{j=1}^{k} n_j H_j}{N},$$

где N — общее число документов, k — число кластеров.

Степень сходства полученного разбиения с эталонным обратно пропорциональна значению энтропии H. Чем меньше значение энтропии H, тем выше степень сходства с эталонным разбиением.

В ходе тестирования были получены следующие зависимости значений энтропии от числа документов, подвергаемых кластеризации:

Критерий, основанный на значении средней меры подобия

Средняя мера подобия для группы j, состоящей из n_j элементов определяется формулой:

$$M_j = \frac{1}{n_j^2} \sum_{i=1,j=1}^{n_j} \frac{d_i\, d_j}{||d_i||\, ||d_j||},$$

d_i и d_j — документы, представленные согласно модели TFIDF.

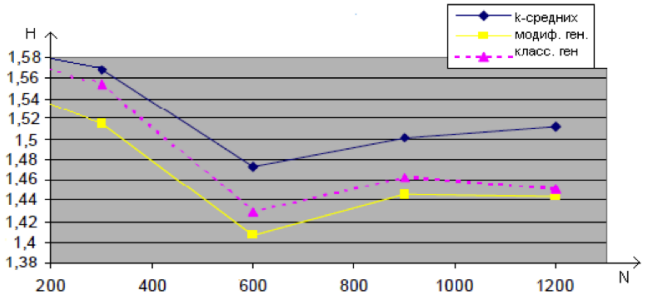


Рис. 1. График зависимости энтропии от числа документов

Тогда средняя мера подобия для всего разбиения определяется:

$$M = \frac{\sum_{j=1}^{k} n_j M_j}{N},$$

где N — общее число документов, k — количество кластеров.

Значение средней меры подобия прямо пропорционально качеству разбиения: чем больше значение средней меры подобия, тем выше качество полученного разбиения.

В ходе эксперимента были получены следующие результаты:

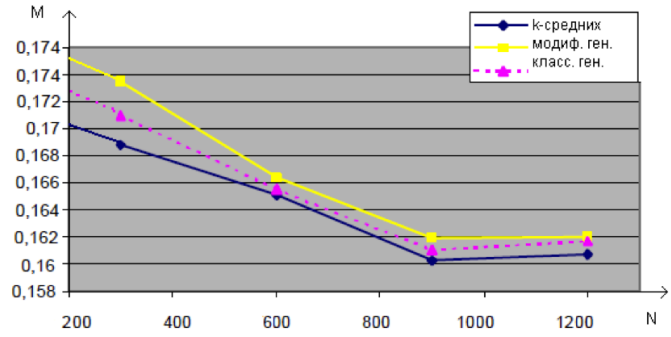


Рис. 2. График зависимости средней меры подобия от числа документов

Для подтверждения результатов использовался непараметрический знаковый ранговый тест Вилкоксона [18]. Случайным образом выбираем

1200 документов из базы «Reuters 21578». Данные документы разбиваются на 5, 10, 15 и 20 групп с помощью классического генетического алгоритма, модифицированного генетического алгоритма и метода k-средних. Для каждого полученного разбиения фиксируем значения энтропии и средней меры подобия. Процедуру повторям 20 раз. Для каждого критерия получаем 80 значений энтропии и средней меры подобия соответственно. Вводится статистика

$$T^* = \frac{T - \dfrac{n(n-1)}{4}}{\sqrt{n(n+1)(2n+1)}},$$

где n — объем выборки, T — сумма рангов величин $z_i = x_i - y_i > 0$.

T^* аппроксимируется стандартным нормальным распределением [19]. Задаем $\alpha = 0,95$. Гипотеза сдвига принимается с достоверностью α, если $T^* < |u_{\frac{1+\alpha}{2}}|$, где u_γ — γ-квантиль стандартного нормального распределения. В противном случае принимается одна из альтернативных гипотез: H_1, если $T^* > u_{\frac{1+\alpha}{2}}$, то средние значения элементов первой выборки превосходят средние значения элементов второй выборки; H_2, если $T^* < -u_{\frac{1+\alpha}{2}}$, то значения элементов второй выборки превосходят значения элементов первой выборки.

Табл. 1. Результат теста Вилкоксона для $\alpha = 0{,}95$. В таблице указаны значения статистики T^* при сравнении метода k-средних, классического генетического алгоритма (ГА) и модифицированного генетического алгоритма (МГА) по энтропийному критерию (ЭК) и по критерию, основанному на значении средней меры подобия (КСМП).

	КСМП	ЭК
метод k-средних и ГА	$-4{,}807$	$-1{,}076$
метод k-средних и МГА	$-8{,}561$	$4{,}713$
ГА и МГА	$-6{,}807$	$3{,}971$

Из таблицы видно, что наибольшая упорядоченность (наименьшее значение энтропии) и наибольшее значение средней меры подобия достигается при кластеризации с помощью модифицированного генетического алгоритма.

Таким образом, непараметрический знаковый ранговый тест Вилкоксона подтверждает результаты проведенных исследований.

Выводы

В данной работе предложен метод кластеризации текстовых документов, в основе которого лежит генетический алгоритм. Дополнительно к классическим операторам скрещивания и мутации были разработаны и реализованы новые.

Проведенное тестирование показало возможность использования модификации генетического алгоритма при решении задач кластеризации текстовых документов. В дальнейшем, данный метод кластеризации может найти свое применение в системах поиска текстовой информации.

Список источников

1. Hammouda P. M., Kamel M. S. Efficient phrase-based document indexing for web document clustering // IEEE Transactions on Knowledge and Data Engineering. — October 2004. — V. 16, № 10. — P. 1279–1296.
2. Runkler T. A., Bezdek J. C. Web mining with relational clustering // International Journal of Approximate Reasoning. — February 2003. — V. 32, № 2–3. — P. 217–236.
3. Rodrigues E. M., Sacks L. A scalable hierarchical fuzzy clustering algorithm for text mining // Proceedings of the 5th International Conf. on Recent Advances in Soft Computing. — Nottingham
4. Allan J. Topic detection and tracking: event-based information organization // Kluwer Academic Publishers. — 2002. — 280 p.
5. Jin H., Wong M.-L., Leung K.-S. Scalable model-based clustering for large databases based on data summarization // IEEE Transactions on Pattern Analysis and Machine Intelligence. — November 2005. — V. 27, № 11. — P. 1710–1719.
6. Hu P., He T., Ji D., Wang M. A study of Chinese text summarization using adaptive clustering of paragraphs // Proc. of the 4th International Conf. on Computer and Information Technology (CIT'04). — Wuhan (China). — September 14–16, 2004. — P. 1159–1164.
7. Алгулиев Р. М., Алыгулиев Р. М., Багиров А. М. Глобальная оптимизация в резюмировании текстовых документов // Автоматика и вычислительная техника. — 2005. — Vol. 39, №6. — С. 52–59.
8. Radev D., Otterbacher J., Winkel A., Blair-Goldensohn S. NewsInEssence: summarizing online news topics // Communications of the ACM. — October 2005. — V. 48, № 10. — P. 95–98.
9. Алгулиев Р .М., Алыгулиев Р. М. Быстрый генетический алгоритм решения задачи кластеризации текстовых документов // Искусственный интеллект. — 2005. — №3. — С. 698-707.

10. Khan M. S., Khor S. W. Web document clustering using a hybrid neural network // Applied Soft Computing. — September 2004. — V. 4, № 4. — P. 423–432.
11. Desai M., Spink A. An algorithm to cluster documents based on relevance // Information Processing and Management. — September 2005. — V. 41, № 5. — P. 1035–1049.
12. Strehl A., Ghosh J. Impact of Similarity Measures on Web-page Clustering // AAAI-2000: Workshop of Artificial Inteligence for Web Search.
13. Гладков Л. А., Курейчик В. В., Курейчик; В. М. Генетические алгоритмы // М.: Физматлит, 2010. — 368 с.
14. Рутковская Д., Пилиньский Л., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы // М.: Горячая линия-Телеком, 2008. — 452 с.
15. Алыгулиев Р. М. Метод кластеризации коллекции документов и алгоритм для оценки оптимального числа классов // Искусственный интеллект. — 2006. — № 4. — С. 21–29.
16. http://www.daviddlewis.com/resources/testcollections/reuters21578/
17. Барсегян А. А., Куприянов М. С., Холод И. И., Тесс М. Д., Елизаров С. И. Анализ данных и процессов: учеб. пособ. // СПб. : БХВ-Петербург, 2009. — 512 с.
18. Кобзарь А. И. Прикладная математическая статистика. Для инженеров и научных работников // М. : Физматлит, 2006. — 816 с.
19. Iman L. M. Use of t-statistic as an approximation to the exact distribution of the Wilcoxon singed rank test statistic // Commun.Statist. 1974. V. 3. P. 795–806.

Рекомендательные системы: тематический обзор

А. Константинов

andrey.v.konst@gmail.com

Функции и общие принципы работы РС

Стремительное развитие Интернета в последние годы привело к тому, что люди оказались "заложниками" прогресса — пользователь является потребителем слишком большого количества информации, потоки которой доставляются ему различными способами. Данное явление получило название *информационной перегрузки* [29]. Одной из основных задач, которая стоит перед исследователями, пытающимися решить проблему информационной перегрузки, является поиск релевантной информации для пользователя в больших массивах данных. В данной ситуации особое значение приобрели такие области обработки и анализа данных как информационный поиск (Information Retrieval — IR) и информационная фильтрация (Information Filtering – IF). Примерами IR систем являются такие поисковые ресурсы как Google и Yandex. Примером IF системы является Twitter, позволяющий из общей ленты сообщений пользователей подписываться только на интересующие вас каналы.

На стыке IF и IR находится еще один класс систем, который получил широкое распространение в последние годы — рекомендательные системы (РС). РС могут быть определены как персональные информационные агенты, которые выбирают для пользователя объекты информационной системы (товары, услуги и т. д.), которые являются наиболее релевантными его интересам [7]. Довольно распространены РС для пользователей Интернета. Первой работой, посвященной рекомендательным системам в Интернете принято считать работу [11], в которой пользователю сети Usenet рекомендовались потенциально интересные сообщения. В качестве современных примеров использования коммерческих рекоменда-

Игнатов Д. И., Яворский Р. Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16–18 марта, 2012

тельных систем можно привести такие известные Интернет-ресурсы как Amazon.com (рекомендация товаров, предлагаемых на сайте), MovieLens и Netflix (рекомендация фильмов), Last.fm и Pandorra (рекомендация музыкальных композиций).

На данный момент существуют десятки различных подходов к построению РС (см. [21]). Общий принцип, на котором основываются все РС, заключается в выявлении пользовательских предпочтений. Предпочтения могут быть выражены в виде числа (например оценки объектов в бинарной, или интервальной шкале) или с помощью бинарных отношений (полный или частичный порядок на множестве объектов). Выявленные предпочтения используются для рекомендации наиболее полезных объектов. Формальное определение данного принципа для РС с числовыми предпочтениями пользователей можно записать следующим образом:

Пусть есть множество пользователей U и множество товаров I. Пусть также задана функция предпочтения p

$$p : U \times I \to R \tag{1}$$

Здесь R — некоторая числовая оценка предпочтительности объекта для пользователя. Тогда задача рекомендации одного объекта определяется следующим образом: Найти для каждого пользователя такой объект, который бы максимизировал функцию p.

$$\forall u \in U(i^* = arg\max_{i \in I} p(u, i)). \tag{2}$$

Приведенный выше принцип с незначительными изменениями применим и к РС, которые отбирают для пользователя не один объект, а множество объектов.

Важной особенностью РС является то, что значительная часть значений функции p неизвестна, а основной задачей исследователей в данной области является получение эффективных алгоритмов нахождения оценок пользовательских предпочтений $\hat{p}$. Многие алгоритмы, используемые в области разрабоки данных были использованы для выявления предпочтений пользователей и построения рекомендаций. Кроме выявления предпочтений существуют следующие направления исследований в области РС [3, 17, 18, 26]: уменьшение временных и ресурсных затрат, улучшение алгоритмов оценки качества РС, увеличение разнообразия рекомендуемых объектов. Существует значительное количество опубликованных работ, посвященных РС. Например, начиная с 2007 года в трудах конференции RecSys[1] было опубликовано более 400 работ.

[1] http://recsys.acm.org/

Значительное количество методов и их модификаций усложняет построение их классификации вручную. В дальнейшем будет проведен обзор всех работ конференций ACM начиная с 2007 года с использованием методов анализа формальных понятий (АФП). В результате будет получена решетка понятий, описывающая методы построения РС. Данный подход можно использовать для автоматического группирования схожих методов.

Далее будет приведен краткий обзор наиболее распространенных методов построения РС, чтобы посвятить читателя, не закомого с данной областью разнообразием существующих подходов и частично обосновать актуальность создания классификации на основе АФП.

Контентные методы

Исторически первая появившаяся группа методов — это контентные методы построения РС. В основе подхода лежит идея анализа тех объектов, которые были оценены пользователем. Для каждого объекта, обладающего высоким уровнем предпочтения, определяются наиболее схожие с ним объекты, которые еще не были оценены. Эти объекты рекомендуются пользователю.

Степень сходства между объектами $sim\,(i_1, i_2)$ определяется с помощью различных мер сходства. Наиболее распространены следующие подходы к определению сходства сходство на основе расстояния (Евклида, Минковского, Махаланобиса, Хэмминга), сходство как корреляция, косинусная метрика сходства, сходство как условная вероятность, и сходство как коэффициент Жаккара (см., например, [13] и [9]). Данные подходы применимы и для коллаборативных методов, о которых будет рассказано ниже.

В [19] приводятся примеры использования контентных рекомендаций в реальных Интернет-системах.

Среди преимуществ контентных методов можно отметить:

- Независимость от количества пользователей в системе и их активности. В процессе получения рекомендаций рассматривается только профиль текущего пользователя.
- “Прозрачность” получаемых рекомендаций. Пользователю при желании можно явным образом объяснить, почему ему показан данный объект (например, “Рекомендуем Вам этот фильм, так как Вам нравятся еще 27 фильмов с Расселом Кроу” [27]).
- Отсутствие проблемы “холодного старта”[2].

[2]Под холодным стартом (*cold start*) в случае контентных РС понимается проблема генерации рекомендаций в отсутствии достаточного количества информации о пользовательских предпочтениях относительно конкретного объекта.

Среди недостатков контентных систем выделяют:

- Необходимость предварительного анализа базы объектов. Здесь подразумевается, что нужно до начала построения оценок $\hat{p}$ представить объекты в виде векторов в пространстве признакоов. Для фильмов такими признаками могут быть год выпуска, режиссер, главные актеры и жанр. В случае, если рекомендуемые объекты имеют различную природу, выделить общие признаки может быть затруднительно.
- Проблема переобучения и отсутствие новизны в рекомендациях. Например, если пользователь оценивал высоко только фильмы Стенли Кубрика, то он будет в рекомендациях видеть только его фильмы.
- Проблема нового пользователя схожа с проблемой переобучения, но заключается в том, что если пользователь оценил только несколько объектов, система не сможет выдать качественные рекомендации из-за ограниченности информации о клиентском профиле. Для полученя точных и разнообразных рекомендаций приходится стимулировать пользователя оценивать объекты.

Коллаборативная фильтрация

Методы коллаборатичной фильтрации (CF) в отличие от контентых методов позволяют использовать предпочтения не только целевого пользователя, а всех пользователей системы. CF методы позволяют генерировать достаточно точные рекомендации без предварительного анализа и описания предметной области. Развитие коллаборативных методов применительно к Интернет РС началось в 90-х годах прошлого столетия (см. первую работу [11]).

Принцип CF методов заключается в анализе данных о пользовательских действиях (выставление оценки, посещение страницы и т. д.), которые могут быть представлены либо протоколом транзакций T, где любой элемент $t \in T$ это тройка вида $t = (u, i, r)$, а $u \in U$, $i \in I$, $r \in \mathrm{Re}$, либо матрицtq D размером $|U \times I|$. Элементы матрицы D — это пользовательские рейтинги, присвоенные объектам $d_{u,i} = r$, где $u \in U$, $i \in I$, $r \in \mathrm{Re}$.

CF методы можно разделить на две основные группы — анамнестические и модельные. Анамнестические методы позволяют генерировать рекомендации проводя анализ имеющихся данных on-line (такой подход принято называть lazy-learning), модельные же методы нуждаются в предварительном этапе обучения — построения модели, на основе которой в последствии формируются рекомендации.

Анамнестические методы. Анамнестические алгоритмы (memory-based) не используют в своей работе заранее расчитанных моделей, а используют непосредственно данные о пользовательских профилях. Все имеющеся данные — это матрица D, элементами которой являются оценки $d_{i,j}$, присвоенные пользователем i объекту j. Классификация — один из наиболее ранних методов получения рекомендаций (см. [11, 24, 25]). Одним из наиболее простых, и в то же время востребованных является метод k ближайших соседей (*kNN*), который используется в РС Amazon.com [14]. Под классом в данном случае понимается значение рейтинга, которое пользователь u поставил бы объекту i. Результатом классификации является оценка предпочтения пользователя $p(\hat{u}, i)$, получаемая путем агрегирования известных предпочтений k пользователей, наиболее схожих с ним. Для этого вначале определяется соседство N_u (neighbouhood) пользователя u, состоящее из k наиболее схожих пользователей:

$$N_u = \{u_i \in U \mid \sum_{i=1..k} sim(u, u_i) \underset{u_i \in U}{\longrightarrow} max\} \tag{3}$$

Затем вычисляется оценка предпочтения пользователя u:

$$\hat{p}(u, i) = \frac{\sum\limits_{j=1..k} sim(u, u_j) p(u_j, i)}{\sum\limits_{j=1..k} sim(u, u_j)}, \tag{4}$$

где k — количество наиболее близких рассматриваемых пользователей в окрестности пользователя u.

Модельные методы. Принципиальное отличие модельных алгоритмов от рассмотренных выше анамнестических заключается в том, что для их работы необходим предварительный этап обучения, в ходе которого строится некоторая модель. Построенная модель хранится в памяти РС и в тот момент, когда необходимо предоставить рекомендации, модель применяется к данным, предоставляя на выходе множество рекомендуемых объектов. Применение модели гораздо менее ресурсоемкий процесс, чем ее построение, поэтому пользователь быстрее получает рекомендации.

Существует значительное количество методов для построения модели РС — Байесовский классификатор [20], анализ скрытых факторов (Latent Semantic Analysis) [12], латентное размещение Дирихле (Latent Dirichlet Allocation) [4], принцип максимума энтропии (Maximum Entropy), машины Больцмана (Boltzmann Machines) [22], метод опорных векторов (Support Vector Machines), сингулярное разложение матриц (Singular Value Decomposition) [6]. Далее мы рассмотрим наиболее часто применяемые методы.

Ассоциативные правила. Одним из используемых подходов построения РС является использвание ассоциативных правил (например, такие рекомендации использует YouTube [8]). Ассоциативные правила представляют собой завимости вида $A \rightarrow B$, где $A, B \subseteq I$, полученные в результате анализа лога транзакций. I является множеством всех объектов, которые участвуют в транзакциях, а запись $A \rightarrow B$ читается как "пользователи, которые использовали объекты из множества A, так же использовали объекты из B". Под транзакциями в данном случае могет пониматься корзины товаров пользователей в Интернет-магазинах; "Виш-листы" (wish-list) пользователей; журнал просмотров страниц.

Показателями качества ассоциативных правил являются величины поддержки (*support*) и достоверности (*confidence*). Поддержка показывает, для какой доли транзакций A и B присутствуют в транзакции одновремено. Достоверность показывает, какая доля транзакций, содержащих объекты A, содерит также и B. Значения минимальной поддержки используются для принятия решения об использовании ассоциативного правила при порождении рекомендаций.

Существует работа ([26]), описывающая применение ассоциативных правил в РС системах для определения схожести между объектами (совместно с другими методами). Кроме того, применяя импликации к тем объектам, которые были высоко оценены пользователем, можно выделить объекты, которые скорее всего также окажутся предпочтительными.

Кластеризация. Кластеризация — еще один метод, используемый для построения моделей РС. Этот метод позволяет находить группы пользователей, обладающих сходными предпочтениями. После отнесения пользователя к тому или иному кластеру, можно использовать обычный user-based или item-based kNN подход, описанный выше, но все расчеты производить только для пользователей внутри кластера, так как они заведомо обладают некоторым сходством ([23]). Традиционные методы кластеризации, такие как K-средних и иерархическая кластеризация позволяют отнести каждого конкретного пользователя только к одному кластеру. Для анализа сложных предпочтений используется нечеткая кластеризация. Для того, чтобы одновременно группировать пользователей и объекты в РС используют различные методы билкастеризации, например алгоритмы Ченга, Bimax, xMotif, алгоритм, основывающийся на объектных и признаковых замыканиях из АФП ([1]).

Классификация. Выше был рассмотрен один метод классификации — kNN. Ниже мы рассмотрим два других метода, которые предусматривают построение модели — деревья решений и байесовские классификаторы.

Деревья решений (ДР) представляют собой семейство алгоритмов классификации из области машинного обучения, отличительными особенностями которых являются понятная для человека структура правил и быстрота классификации. Применительно к РС деревья решений могут быть использованы для построения модели, которая позволит отнести объект к той или иной группе пользователей, объединенных общими интересами. В [5] рассматривается пример РС, в которой дерево строится для каждого пользователя и позволяет в процессе использования наглядно объяснять на чем основывается такая оценка для данного объекта.

Байесовские классификаторы (БК) довольно популярны как средство построения РС. Чаще всего БК используются в контентных системах, но есть примеры их использования и в коллаборативных РС. В работе [10] рассматривается пример РС, построенной на байесовском классификаторе. В работе Miyahara и Pazzani [15] описывается реализация рекомендательной системы, в которой в качестве модели выступает наивный байесовский классификатор, относящий объекты к одному из классов — “нравится” и “не нравится”.

Матричная и тензорная факторизация. Матрица оценок D может иметь значительный размер $|U \times I|$, что, во-первых, затрудняет использование таких подходов как kNN из-за большого количества требуемых вычислительных ресурсов, а, во-вторых, многие метрики схожести становятся неадекватными при большом количестве измерений. Для преодоления данной проблемы используются различные методы понижения размерности, например — матричная факторизация (МФ). МФ ([6]) соотносит пользователей и объекты с единым пространством скрытых факторов.

Данный подход может использоваться как предварительный этап перед использованием других методов, а может применяться как самостоятельный способ получения рекомендаций. В целях уменьшения размерности исходных данных применяются такие методы как SVD, LSA и PLSA ([16]), PCA. Основные усилия исследователей в данной области направлены на развитие методов инкрементального обновления модели для применения в РС “реального времени” (instant-RS).

Существуют таже такие системы, в которых данные носят мультимодальный характер, например, т.н. фолксономии, базовая структура данных для которых — это тернарное отношение $Y \subseteq U \times T \times R$, показывающее что пользователь u пометил тегом t ресурс r, где $(u, t, r) \in Y$. В этом случае данные также можно описать трехмерным тензором и применять методы понижения размерности для многомерных данных.

Гибридные методы

Помимо рассмотренных выше коллаборативных и контентных методов используют различные их сочетания — *гибридные* методы. Основная идея построения гибридных методов заключается в устранении недостатков одного метода с помощью возможностей другого, например устранение проблемы "холодного старта" для CF систем путем добавления предварительного контентного анализа ([2]). Пример гибридизации применительно к Интернет РС рассмотрен в [7]. В данной работе делается попытка описания совместимости различных методов. В [28] рассматривается совмещение бикластеризации и kNN классификации в целях улучшения скорости работы алгоритма.

Задачи предстоящей работы

Данный обзор позволит читателю получить поверхностное представление о тех методах, которые используются в современных РС. Как уже было упомянуто в начале работы, количество опубликованных работ довольно велико, а темпы развития данной области стремительно растут. В связи с этим, перед исследователем, начинающим знакомиться с РС, встает задача выбора релевантных статей и методов, которые он может использовать в своей работе.

Автором была поставлена задача создания обобщенной объектно-признаковой классификации с использованием анализа формальных понятий (АФП). В качестве данных будет использована база публикаций конференции RecSys за все время ее существования. Каждая работа будет представлена в виде вектора атрибутов, выделенных на основе анализа названия, аннотации и служебных полей документа. В результате будет получена тройка $K = (G, M, I)$, называемая формальным контекстом, где G — множество всех документов, M — множество всех выделенных признаков, $I \subseteq G \times M$ — бинарное отношение. Объект $g \in G$ обладает атрибутом $m \in M$, если выполняется gIm.

На множествах $A \subseteq G$ и $B \subseteq M$ определена пара отображений:

$$A' = \{m \in M \mid \forall\, g \in A : gIm\} \tag{5}$$

$$B' = \{g \in G \mid \forall\, m \in B : gIm\} \tag{6}$$

Эти отображения задают соответствие Галуа между частично упорядоченными множествами. Оператор $(\cdot)''$ является оптратором замыкания. Если для рассмотренных выше A и B выполняется $A' = B, B' = A$, то пара A и B называется формальным понятием. A называют *объемом* понятия, а B — *содержанием* понятия. A и B являются замкнутыми множествами.

Поиск формальных понятий на описанном формальном контексте позволит сгруппировать работы, обладающие общим множеством признаков, и таким образом облегчит исследователю поиск работ, близких к его тематике. Увеличивая или уменьшая множество атрибутов текстов можно будет выбирать между релевантностью и количеством рекомендуемых документов. Как один из результатов рассматривается создание мета-рекомендательной системы, позволяющей находить релевантные работы для исследователей, занимающихся РС.

Список источников

1. Игнатов Д. И., Каминская А. Ю., Кузнецов С. О., Магизов Р. А. Метод бикластеризации на основе объектных и признаковых замыканий. // *Интеллектуализация обработки информации: 8-я международная конференция. Республика Кипр, г. Пафос, 17-24 октября 2010 г.: Сборник докладов.* — М.: МАКС Пресс, С. 140–143, 2010.
2. Adomavicius G. and Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.*, 17(6):734–749, 2005.
3. Adomavicius G., Tuzhilin A., Berkovsky S., De Luca E. W. and Said A. Context-Awareness in Recommender Systems: Research Workshop and Movie Recommendation Challenge. *Human Factors*, page 60558, 2010.
4. Blei D. M., Ng A. Y. and Jordan M. I. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
5. Bouza A., Reif G., Bernstein A., and Gall H. SemTree: Ontology-Based Decision Tree Algorithm for Recommender Systems. In *International Semantic Web Conference*, 2008.
6. Brand M. Fast online SVD revisions for lightweight recommender systems. In *In Proceedings of the Third SIAM Conference on Data Mining. Montreal: SIAM, 2003*, pages 37–46., 2003.
7. Brusilovsky P., Kobsa A., and Nejdl W., editors. *The Adaptive Web, Methods and Strategies of Web Personalization*, volume 4321 of *Lecture Notes in Computer Science*. Springer, 2007.
8. Davidson J., Liebald B., Liu J., Nandy P., Van Vleet T., Gargi U., Gupta S., He Y., Lambert M., Livingston B., and Sampath D. The youtube video recommendation system. In Xavier Amatriain, Marc Torrens, Paul Resnick, and Markus Zanker, editors, *RecSys*, pages 293–296. ACM, 2010.
9. Deshpande M. and Karypis G. Item-based top-n recommendation algorithms. *ACM Trans. Inf. Syst.*, 22:143–177, January 2004.

10. Ghani R. and Fano A. Building recommender systems using a knowledge base of product semantics. In *2nd International Conference on Adaptive Hypermedia and Adaptive Web Based Systems*, 2002.
11. Goldberg D., Nichols D., Oki B. M., and Terry D. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, December 1992.
12. Hofmann T. Latent Semantic Models for Collaborative Filtering. *ACM Transactions on Information Systems (TOIS)*, 22(1):89–115, 2004.
13. Ignatov D. I., Poelmans J., Dedene G., and Viaene S. A new cross-validation technique to evaluate quality of recommender systems. Kundu M. K., Mitra S., Mazumdar S., and Pal S. K., editors, *PerMIn*, volume 7143 of *LNCS*, pages 195–202. Springer, 2012.
14. Linden G., Smith B., and York J. Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, January 2003.
15. Miyahara K. and Pazzani M. Collaborative filtering with the simple bayesian classifier. In Riichiro Mizoguchi and John Slaney, editors, *PRICAI 2000 Topics in Artificial Intelligence*, volume 1886 of *Lecture Notes in Computer Science*, pages 679–689. Springer Berlin / Heidelberg, 2000.
16. Musto C. Enhanced vector space models for content-based recommender systems. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 361–364, New York, NY, USA, 2010. ACM.
17. Nanopoulos A., Radovanović M., and Ivanović M. How does high dimensionality affect collaborative filtering? *Proceedings of the third ACM conference on Recommender systems — RecSys '09*, page 293, 2009.
18. Park Y.-J. and Tuzhilin A. The long tail of recommender systems and how to leverage it. In *Proceedings of the 2008 ACM conference on Recommender systems*, RecSys '08, pages 11–18, New York, NY, USA, 2008. ACM.
19. Pazzani M. J. and Billsus D. 10 Content-Based Recommendation Systems. *The Adaptive Web*, pages 325 — 341, 2007.
20. Pronk V., Verhaegh W., Proidl A., and Tiemann M. Incorporating user control into recommender systems based on naive bayesian classification. In *Proceedings of the 2007 ACM conference on Recommender systems*, RecSys '07, pages 73–80, New York, NY, USA, 2007. ACM.
21. Ricci F., Rokach L., Shapira B., and Kantor P. B., editors. *Recommender Systems Handbook*. Springer, 2011.

22. Salakhutdinov R. and Mnih A. Probabilistic Matrix Factorization. In *In 25th International Conference on Machine Learning (ICML-2008)*, pages 1–8, 2008.
23. Sarwar B., Karypis G., Konstan J. A., and Riedl J. T. Application of Dimensionality Reduction in Recommender System — A Case Study. In *ACM WebKDD 2000 Web Mining for E-Commerce Workshop*, 2000.
24. Sarwar B. M., Karypis G., Konstan J. A., and Riedl J. Item-based collaborative filtering recommendation algorithms. In *WWW*, pages 285–295, 2001.
25. Ben Schafer J., Frankowski D., Herlocker J. L., and Sen S. Collaborative Filtering Recommender Systems. *The Adaptive Web*, pages 291 — 324, 2007.
26. Smyth B. and McClave P. Similarity vs. diversity. In *Proceedings of the 4th International Conference on Case-Based Reasoning: Case-Based Reasoning Research and Development*, ICCBR '01, pages 347–361, London, UK, 2001. Springer-Verlag.
27. Symeonidis P., Nanopoulos A., and Manolopoulos Y. Moviexplain: a recommender system with explanations. In *Proceedings of the third ACM conference on Recommender systems*, RecSys '09, pages 317–320, New York, NY, USA, 2009. ACM.
28. Symeonidis P., Nanopoulos A., Papadopoulos A. N., and Manolopoulos Y. Nearest-biclusters collaborative filtering based on constant and coherent values. *Information Retrieval*, 11(1):51–75, December 2007.
29. Van Setten M. *Supporting people in finding information*. Telematica Instituut, 2005.

Интеллектуальное автодополнение для электронных таблиц

А. Мелентьев

amelentev@gmail.com

Уральский Федеральный Университет имени первого Президента России Б. Н. Ельцина, Екатеринбург, Россия

Аннотация. Представьте, что Excel автоматически распознает ваши намерения с первых нажатий клавиатуры и автоматически заполняет оставшийся текст в ячейке. Это не только бы существенно улучшило технологии автодополнения и проверки ошибок в Excel, но и сделало бы ввод на сенсорных устройствах гораздо более удобным. В работе представлена система, позволяющаяся существенно улучшить предложения для автодополнений в электронных таблицах. Система находит наиболее вероятные шаблоны среди введённых данных в столбце и использует эти шаблоны для генерации предложений. Система реализована как библиотека .NET (на языке F#) и также как расширение для Microsoft Excel 2010. Эксперименты на реальных таблицах (EUSES Spreadsheet Corpus[2]) показывают, что разработанная интеллектуальная система для генерации автодополнений позволяет существенно сократить необходимость использования клавиатуры для ввода. Также замечена значительная корреляция эффективности системы с энтропией данных, что теоретически обосновывает хорошие экспериментальные результаты.

Ключевые слова: автодополнение; электронные таблицы; IntelliSense.

Игнатов Д. И., Яворский Р. Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16–18 марта, 2012

Введение

Ввод данных во многие электронные таблицы (spreadsheets) однообразен и утомителен. Во вводимых данных можно обнаружить много закономерностей и повторений. Часто получается, что возможных вариантов для следующей ячейки немного. Зачем вводить все символы, если можно просто выбрать подходящий вариант из списка?

Для примера рассмотрим последовательность из табл. 1(a). Последовательность состоит из заголовков или подзаголовков с соответствующим номером. Очевидно, что следующий элемент последовательности может быть одним из двух элементов: “Заголовок” с номером на 1 больше предыдущего либо “Подзаголовок” с тем же номером заголовка и номером подзаголовка на 1 больше предыдущего, либо “1”, если предыдущий элемент был заголовком. Таким образом, вместо того, чтобы вводить все символы (более 11) следующего элемента, можно его просто выбрать из двух вариантов.

Табл. 1. Примеры

(a)

Заголовок 1
Подзаголовок 1.1
Подзаголовок 1.2
Заголовок 2
Подзаголовок 2.1
Подзаголовок 2.2
?
Варианты:
Подзаголовок 2.3
Заголовок 3

(b)

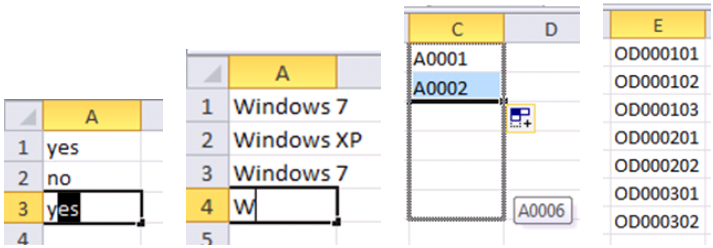
A0000001
A0000002
B0000001
B0000002
B0000003
C0000001
?
Варианты:
C0000002
D0000001

(c)

Windows 7 Professional
Windows 7 Home Basic
Windows XP Home
Windows 7 Home Basic
Windows 7 Ultimate
. . .
?
Варианты:
Различные редакции windows

Рассмотрим следующий пример из табл. 1(b). Тут можно заметить, что если буква не изменяется, то число изменяется на 1. А если буква изменились (на 1), то число становится равным “0000001”. Соответственно следующим элементом последовательности может быть либо “C0000002” либо “D0000001”.

Последний пример из табл. 1(c) содержит различные редакции Windows. Их, естественно, ограниченное число. А последовательность может быть очень большой. Поэтому в этом случае разумно просто предоставить пользователю на выбор список всех редакций.



(a) Тривиальный случай (b) Нетривиальный случай (c) Прогрессия (d) “Нестрогая” прогрессия

Рис. 1. Возможности автодополнения/автозаполнения в электронных таблицах

Возможности электронных таблиц

Рассмотрим, как могут решить данную проблему программы для работы с электронными таблицами (Microsoft Excel, OpenOffice Calc). Для упрощения ввода в них имеются следующие возможности:

— Автодополнение для тривиальных случаев.
Работает только тогда, когда по префиксу можно однозначно определить всю строку. Подходит для тривиальных случаев, таких как последовательность из yes/no на рис. 1(a), но не подходит для примера редакций Windows из табл. 1(c), т. к. там все элементы начинаются одинаково.

— Автозаполнение арифметических прогрессий, повторений.
Выделяете два первых члена и указываете желаемую длину прогрессии и выбранный диапазон заполнятся арифметической прогрессией с заданным шагом (см. рис. 1(c)).

Получается, что программы предлагают только *один вариант* для автодополнения. Но этого часто бывает недостаточно, т. к.

— Возможно несколько вариантов для следующего значения.
см. табл. 1 и рис. 1(b).

— “Нестрогая” прогрессия: с перескоками, повторениями.
см. рис. 1(d).

Также можно упомянуть исследовательский проект QuickCode[1] который позволяет найти строковую формулу описывающую закономерность данных на основе примеров. Его можно использовать и как средство автодополнения, но он также предлагает только один вариант. QuickCode реализован как расширение MS Excel.

Идея

Как можно улучшить автодополнение в электронных таблицах? В данной работе предлагается (см. рис. 2):

— По заданным *истории ввода* и *префиксу* текущего элемента
— Вычислять ***распределение*** возможных исходов

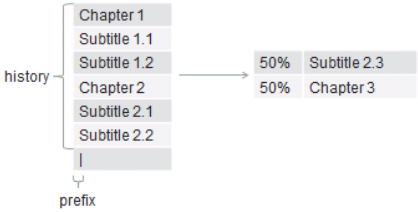


Рис. 2. Схема идеи

Это позволит:

— Существенно улучшить возможности автодополнения в электронных таблицах
— Уменьшить количество ошибок (меньше печатаем — меньше ошибок)
— Радикально упростить ввод для устройств с сенсорными экранами (смартфонов, планшетов)
  Т.к. гораздо легче нажать пальцем на вариант, чем вводить все символы на неудобной сенсорной (экранной) клавиатуре.

Реализация

Автором было реализовано расширение к Microsoft Excel 2010, см. рис. 3, которое добавляет специальную панель для выбора вариантов автодополнений и интерфейс выбора используемых шаблонов (см. раздел «Шаблоны»). Основной целью работы являлось разработка алгоритма генерации автодополнений, так что интерфейсу уделено немного времени.

Шаблоны. Немного о реализации. Система использует примитивные шаблоны для описания составных. Примитивный шаблон может быть:

— Константой
  Например, последовательность из yes и no может быть описана двумя примитивными шаблонами: константами yes и no.

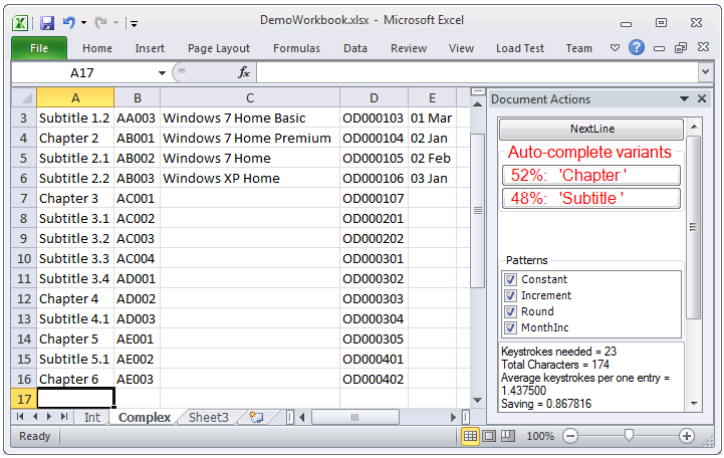


Рис. 3. Скриншот Microsoft Excel 2010 с авторским расширением

— Инкрементом
Например, простая арифметическая прогрессия с шагом 1 описывается примитивным шаблоном $+1$.

Можно добавлять любые другие шаблоны, предоставив 2 специальные функции из раздела «Функции».

Составные шаблоны образуются как зависимости между примитивными. Например, последовательность из табл. 1(b) может быть описана следующим образом:
Если буква не меняется, то число увеличивается на 1
$(letter + 0 \rightarrow num + 1)$.
Если буква увеличивается на 1, то число сбрасывается в 1
$(letter + 1 \rightarrow num = 1)$.

Функции. Для описания примитивного шаблона используются 2 *частичные* функции:

— $Construct$: (история, текущее значение) $\rightarrow$ параметры шаблона
Используется для распознавания шаблонов.
По истории и текущему значению возвращает параметры шаблона.

— $Apply$: (история, параметры шаблона) $\rightarrow$ вариант автодополнения
Используется для генерации вариантов автодополнения.
По истории и параметрам шаблона возвращает один вариант автодополнения.

Примеры:

- Инкремент
  $Construct(history, cur) = cur - history.last$
  $Apply(history, delta) = history.last + delta$
  ($history.last$ — последний элемент в истории)
- Константа
  $Construct(history, cur) = cur$
  $Apply(history, c) = c$
- Округление вверх
  $Apply(history, prec, offset) = \lceil \frac{history.last}{10^{prec}} \rceil 10^{prec} + offset$
  Пример:
  $Apply(history.last = 159, prec = 2, offset = 1) = 201$

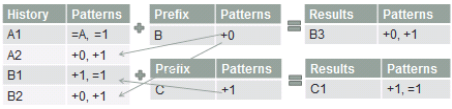


Рис. 4. Пример работы алгоритма

Алгоритм. Продемонстрируем работу основного алгоритма 1 на примере (см. рис. 4). Допустим, пользователь ввёл столбец (history) из значений *“A1”; “A2”; “B1”; “B2”*. Из этих значений с помощью функции $Construct$ получаются следующие шаблоны: *“=A,=1”; “+0,+1”; “+1,=1”; “+0,+1”* (здесь *“=C”* означает константа C, а *“+I”* — инкремент на I. инкремент относится к соответствующему (положению) токену в предыдущей ячейке. для наглядности приведены только необходимые шаблоны). Теперь если пользователь в следующей ячейке ввёл символ “B”, что соответствует шаблону *“+0”*, алгоритм ищет в истории шаблоны, начинающиеся на *“+0”* и находит две одинаковых последовательности *“+0,+1”*. Из этой последовательности применением функции Apply получается результат *“B3”*. А если пользователь ввёл “C”, что соответствует шаблону *“+1”*, то в истории найдётся только одна похожая последовательность *“+1,=1”*, и в результате получится *“C1”*.

Оценка сложности алгоритма $= \mathcal{O}(max(N(P+1)K, NKlogNK))$, где N — длина истории (столбца), P — количество токенов в префиксе (хранить больше $P+1$ токенов не нужно), K — количество примитивных шаблонов. Теоретическое максимальное количество генерируемых вариантов автодополнений $= NK$. $NKlogNK$ возникает из-за необходимости сортировки при построении распределения и выбора наиболее частых вариантов. На оценку также влияют и сложность функций $Construct$ и $Apply$, но в данной работе использованные шаблоны имели константную сложность. Если использовать алгоритм инкрементально

Algorithm 1: Основной алгоритм

Data: $history$ — история ввода (столбец);
$prefix$ — префикс (значение текущей ячейки)
$patterns$ — множество используемых примитивных шаблонов
Result: распределение возможных вариантов для автодополнения

begin
  $history.add(prefix)$ `// добавить префикс в конец истории`
  $N = |history|$
  `// разбить строчки истории на токены (числа, буквы и пр.)`
  for $i = 0 \,..\, N-1$ **do**
    $historyt[i] = tokenize(history[i])$
  `/* По токенам построить последовательность множеств примитивных шаблонов (используя функции Construct) */`
  for $i = 0 \,..\, N-1$ **do**
    for $j = 0 \,..\, |row| - 1$ **do**
      for $pattern \in patterns$ **do**
        $historyp[i][j].add(pattern.Construct(historyt[0..i - -1], row[i]))$
  `/* Искать в` $historyp$ `последовательности, похожие на префикс. Две последовательности(a,b) похожи, если их множества пересекаются в каждом индексе, т. е.` $\forall i < min(|a|, |b|)\ \ a[i] \cap b[i] \neq \emptyset$ `*/`
  $prefixp = historyp[N-1]$
  $similar = findsimilar(historyp[0..N-2], prefixp)$
  `/* Применить к найденным последовательностям функцию Apply и по получившемуся набору построить дискретное распределение */`
  $result = Apply(historyt[0..N-2], similar)$

(то есть кэшировать результаты между запусками), то первый аргумент максимума можно уменьшить до PK.

Испытания

В качестве данных для испытаний была выбрана база “Spreadsheet Corpus”[2] от консорциума End Users Shaping Effective Software (EUSES). Это большой набор реальных таблиц, предназначенный для тестирования и анализа инструментов для работы с электронными таблицами. Размер $\approx$ 650 мегабайт, 5607 файлов.

В качестве критерия для оценки алгоритма в работе используется *Keystrokes saving*, который рассчитывается так:
Если необходимо ввести все данные электронных таблиц заново по столбцам, и мы можем вводить символы с клавиатуры то, как много нажатий на клавиатуру мы сможем избежать если мы дополнительно можем выбирать из 10 вариантов автодополнения предоставленных алгоритмом.
Для перехода на следующую ячейку клавиатура не используется. (Клавиатура используется только для ввода символов, копировать/вставлять нельзя) Если мы не используем автодополнение, то очевидно Keystrokes saving = 0.

Результаты испытаний показаны на рис. 5.

Как видно из результатов, разработанный алгоритм позволяет избежать 65% нажатий на клавиатуру. Использование специализированных шаблонов не улучшает оценку, т.к. они полезны только на небольшом количестве данных.

Корреляция с энтропией. При испытаниях также была обнаружена зависимость эффективности алгоритма от энтропии данных. В качестве аппроксимации для энтропии использована степень сжатия стандартным алгоритмом DEFLATE (используемом в архиваторах zip, gz). Корреляция между степенью сжатия и keystrokes saving получилась = 0,51
(0 — нет зависимости, 1 — строгая линейная зависимость)
Корреляция теоретически объясняет хорошие экспериментальные результаты алгоритма, т. к. чем меньше энтропия, тем больше степень сжатия и тем больше можно найти закономерностей, и тем лучше работает алгоритм.

Выводы

В работе представлен алгоритм для генерации вариантов автодополнений, который существенно превосходит встроенные в программы электронных таблиц возможности. Разработанный алгоритм позволяет

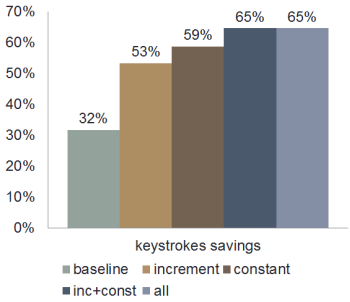


Рис. 5. Результаты
$baseline$ — тривиальный алгоритм, который просто предлагает значение предыдущей ячейки.
$increment$ — авторский алгоритм с использованием инкремента как единственного примитивного шаблона
$constant$ — с использованием константы как единственного примитивного шаблона
$inc + const$ — с использованием инкремента и константы
all — инкремент + константа + два специализированных шаблона: округление сверху и инкремент дат (пример $January + 1 = February$)

избежать 65 % нажатий на клавиатуру при вводе данных (см раздел «Испытания»). Алгоритм способен за приемлемое время адекватно предсказывать как простые, так и довольно сложные закономерности данных(см. табл. 1).

Работа была представлена отделу разработки Microsoft Excel. Отдел рассматривает алгоритм для улучшения и включения в следующие версии Excel.

Возможные направления будущего развития:

— Расширить алгоритм для учёта нескольких столбцов.
  В текущем виде алгоритм работает только с одним столбцом. Но данные в ячейке разделяются на токены, что аналогично как таблица разделяется на столбцы. Поэтому достаточно легко можно расширить алгоритм для работы с несколькими столбцами.
— Улучшение интерфейса.
  Для демонстрации сейчас используется дополнительная панель в Excel, реально использовать которую не очень удобно. В будущем планируется интегрироваться во встроенную систему автодополнения Excel и добавить выпадающий список вариантов.

— Интеграция в программы электронных таблиц для смартфонов и планшетов
  Алгоритм будет наиболее полезен на сенсорных устройствах
— Применить для других данных
  Алгоритм можно применить для любых интерфейсов ввода последовательностей.

Благодарности

Данная работа была выполнена автором летом 2011 года в интернатуре Microsoft Research под руководством Sumit Gulwani и Madan Musuvathi.

Список источников

1. *Gulwani, S.* Automating string processing in spreadsheets using input-output examples / S. Gulwani // POPL. — ACM, 2011. — Pp. 317–330.
2. *Ii, M. F.* The EUSES Spreadsheet Corpus: A Shared Resource for Supporting Experimentation with Spreadsheet Dependability Mechanisms / M. F. Ii, G. Rothermel // In 1st Workshop on End-User Software Engineering. — 2005. — Pp. 47–51.

Автоматизация подготовки исходных текстовых данных из сети интернет для дальнейшего анализа

Найденов Никита

Российской академии наук Вычислительный центр им. А.А.Дородницына РАН

Аннотация. Данная работа посвящена созданию инструментов автоматической обработки информационных интернет ресурсов и несет в себе практическую значимость в задачах анализа текста. Во введении обосновывается актуальность выбранной темы, формулируются цель и задачи исследования, указывается объект и предмет исследования. Рассматривается такая задача, как сбор и первичная обработка текстовых данных из новостных источников. Автор предоставляет механизм автоматической обработки большого количества HTML страниц, приводит практическое применение данного подхода на реальных данных открытых ресурсов Государственной телерадиокомпании. Приводится сравнение последовательной и параллельной обработки источников, перечисление достоинств и недостатков таких методов и представление результатов, полученных в ходе выполнения работы. Статья содержит подробное описание метода разбора HTML структуры веб-страниц, а также применимость такого подхода для обработки источников, имеющих похожую архитектуру, удобную для новостных ресурсов.

Ключевые слова: Сбор данных, краулер, новостные источники, обработка HTML, анализ текста.

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16-18 марта, 2012.

Введение

Все данные, представленные в глобальной сети Интернет, можно назвать неструктурированными, ввиду индивидуальности и специфичности архитектуры каждого ресурса. В основном, такие данные – это HTML страницы, т.е. текстовые структуры. В настоящее время, в связи с постоянным ростом информации во всемирной паутине, необходимо развитие технологий, позволяющих использовать ее для решения различных производственных задач предприятий и организаций, вследствие чего активно развивается область анализа текстовых данных и неструктурированной информации. Общепринятым является разбиение задачи предварительной обработки данных на 3 этапа: консолидация, трансформация и очистка [1]. Самым трудоёмким этапом является консолидация данных, которая включает в себя сбор данных. Если исследования ведутся с большой выборкой, то для того, чтобы накопить достаточное количество материала, могут уйти недели или месяцы кропотливого труда. Данная работа посвящена разработке методов автоматического сбора информации из открытых интернет источников.

Описание данных

Данные, которые требовалось собрать и обработать, представляют собой статьи в новостных ресурсах. Объектом данных является статья (или новость), у которой есть заголовок, дата появления в СМИ и сам текст сообщения. Таким образом, данные представлены в формате HTML. Общий объем извлеченных данных составил около 220 Мб. Источником данных являются 43 региональных сайта Государственной Телерадиокомпании.

Постановка задачи

Задачу предварительной обработки данных, как упоминалось выше, можно разложить на 3 этапа: консолидация, трансформация и очистка.

Консолидация – это комплекс методов и процедур, направленных на извлечение данных из различных источников, обеспечение необходимого уровня их информативности и качества, преобразование к единому формату, в котором они могут быть загружены в хранилище данных или аналитическую систему.

Трансформация – комплекс методов и алгоритмов, направленных на оптимизацию представления и форматов данных с точки зрения решаемых задач и целей анализа. Трансформация данных не ставит целью

изменить информационное содержание данных. Ее задача – представить эту информацию в таком виде, чтобы она могла быть использована наиболее эффективно.

Очистка – процесс выявления и исключения различных факторов, мешающих корректному анализу данных: аномальных и фиктивных значений, пропусков, дубликатов и противоречий, шумов и т.д.

В рамках данной работы рассматривается задача консолидации данных. Опишем постановку реальной задачи, которая решалась в процессе работы: требуется собрать новостные материалы из открытых источников за последние полтора года. Как говорилось выше, нас интересуют сайты Государственной Телерадиокомпании. Результатом является отсортированный список новостей с датами их появления и заголовками. Сортировка осуществляется по дате появления новости в СМИ.

Обзор существующих решений

Существует множество систем для выполнения задач извлечения текстовых данных из интернет-источников. Вот некоторые из них: TSIMMIS, WebOQL, FlORID, XWRAP, RoadRunner, Lixto, RAPIER, SRV, WHISK. Последние три направлены на работу с относительно грамматически связными текстами, некоторые привязаны к определенной однотипной структуре данных. В основном, они узко специализированы под конкретные задачи, поэтому использование данных технологий не является уместным в рамках данной работы.

Ход выполнения работы

Для ручной обработки ресурса, в котором за год накопилось около 4000 новостей, занимает огромное количество времени. Задача извлечения информации из интернет-источников была бы гораздо проще, если бы существовал единый стандарт построения сайтов. Но, к сожалению, такие стандарты отсутствуют – все многообразие сайтов и web-страниц объясняется фантазией веб-дизайнеров. Единственное, что их объединяет, – это язык HTML, который определяет внешний вид Интернет-ресурсов, но не может описать его содержание.

Большинство новостных сайтов имеют возможность предоставлять новостную ленту в формате RSS, что позволяет легко использовать существующие обработчики данных в таком виде. Однако в конкретной задаче могут встречаться источники, в которых не предусмотрен такой новостной формат. Таковыми являются ресурсы в нашей работе, поэто-

му применение существующих RSS-технологий неприемлемо. Несмотря на то, что данные сайты работают независимо, некоторое сходство у них все же есть. Все они информационные источники, которые предоставляют данные в виде новостей. «Новость» в нашем понимании – это объект данных, представляющий собой информацию о заголовке, дате создания и о самом тексте сообщения. Практически у всех ресурсов можно выделить трехуровневую структуру – это календарь с датами (архив), где каждой дате соответствует список новостей за этот день. Этот список содержит краткую информацию о событии и ссылку на него. Третий уровень представляет собой полноценную новость, т.е. интересующие нас данные плюс некоторое обрамление. У конкретных источников некоторых полей может не быть, или вместо них могут появиться другие. Также блок содержит некоторый «информационный» мусор, т.е. данные, не имеющие отношения к конкретной извлекаемой новости или ее реквизитам, который при извлечении сообщения надо постараться отфильтровать

Необходимо для каждого ресурса обойти все такие новости за интересующий период и собрать их в один список. Основной задачей извлечения данных из сети интернет является: получение определенных фрагментов информации (поля) из указанных HTML документов.

Около 30 источников имеют практически одинаковую структуру, поэтому для этой группы был написан один общий скрипт. *Поисковый робот (web crawler, веб-краулер)* - специальная программа, основная задача которой является сканирование веб-страниц с последующей обработкой данных. В дальнейшем будем использовать слово «краулер» для программы-сборщика данных.

Для начала рассмотрим пример для одного источника. Краулер обходит архив за интересующий нас период и последовательно просматривает все новости, т.е. имитирует действия обычного пользователя только с огромной скоростью. Когда сборщик открывает страницу непосредственно с новостью, то ему надо указать, какую часть заносить в свою базу. А именно, какие поля в HTML нас интересуют как компоненты одной новости. Применение методов, основанных на работе с HTML-разметкой, позволяет не принимать во внимание лингвистические особенности текста [2]. Поэтому в данном случае используется технология CSS/XPath Selectors [3]. Краулер накапливает данные и при завершении обхода сохраняет их в нужном формате. Приведем подробное описание технологии подключения к источникам и сбора информационных полей.

Так как на момент выбора технологии, с помощью которой будет осуществляться подключение к ресурсам, у меня уже была динамически подключаемая библиотека, написанная на C#, то выбор пал на нее. Во-первых, она обладает всем необходимым функционалом для соединения

с открытыми источниками, а во-вторых, ввиду того, что она создана мной, то принцип ее использования и все возможности очень хорошо известны.

Вся необходимая нам информация (заголовок, дата создания, текст сообщения) находится на одной HTML странице. Также эта структура содержит «лишние» данные: рекламы, ссылки с новостями, контактные данные и т.д. Для извлечения нужной информации используются CSS\XPath селекторы, которые задают шаблон, соответствующий тегу в HTML структуре. Соответственно, для каждого источника необходимо указать три таких шаблона: для заголовка, даты и текста. Иногда текст внутри информационных тегов содержат символы HTML-разметки, поэтому после извлечения данных происходит преобразование текста, так называемая, «очистка» от структурных элементов.

Каждый новостной ресурс содержит архив новостей, где для его просмотра необходимо указать интересующую дату. В большинстве случаев это делается путем передачи параметра в GET запросе при обращении к источнику, поэтому добавляется еще один параметр для источника – формат url запроса по датам.

Как показала практика, сбор информации из одного источника за период в один год занимает около 20 минут. Конечно, многое зависит от скорости самого сервера, на котором расположен ресурс, его ограничения и т.д., но будем придерживаться среднего значения по всем источникам. 20 минут – это, безусловно, быстрее, чем сбор данных вручную (для сравнения: 3 человека обработали один источник за 2 дня). Однако это достаточно длительный период для получения результатов, особенно, если источников более 40. Здесь появляется подзадача оптимизации сбора.

Самым очевидным и, пожалуй, самым верным решением является распараллеливание всей работы краулера. Если сервер с данными не имеет жестких ограничений на подключение, то данный метод вполне допустим. Был выбран следующий механизм обхода:

- Краулер обходит весь календарь (архив) по месяцам.
- В каждом месяце начинается распараллеливание на дни. Если в месяце каждый день имеет какую-либо новостную ленту, то появляется около 30 так называемых «рабочих» потока
- Каждый поток, в свою очередь, в зависимости от количества новостей также разделяет свою работу на несколько потоков
- Каждый такой «рабочий» записывает данные в один общий словарь (список)

Достоинства и недостатки

Главное достоинство такого подхода заключается в том, что скорость обработки одного ресурса практически оптимальна. Однако есть несколько недостатков: во-первых, необходимо четко вылавливать все исключения и ошибки, потому что в такой группе потоков можно очень легко потерять данные или пропустить важное исключение в структуре источников; также такой метод очень сильно зависит от характеристик сервера – часто мы имеем дело с серверами, у которых стоят ограничение на количество подключений с одного адреса. В таком случае, необходимо правильно настроить данный инструмент, используя оптимальные параметры.

Параллельность обработки интернет ресурса позволила сократить среднее время с 20 до 3 минут. К сожалению, ввиду вышесказанных недостатков, не ко всем источникам удалось применить такой подход. Например, сайт ГТРК Ставропольского края имеет достаточно жесткое ограничение на подключения к серверу, поэтому его пришлось обрабатывать последовательно.

Объем данных от источников варьировался от 1 Мб до 22 Мб (рис. 1), время обработки одного источника от 1 до 7 минут (не принимая в расчет сайт с ограничением). Проведенные работы явно показывают применимость и выгоду предложенного метода.

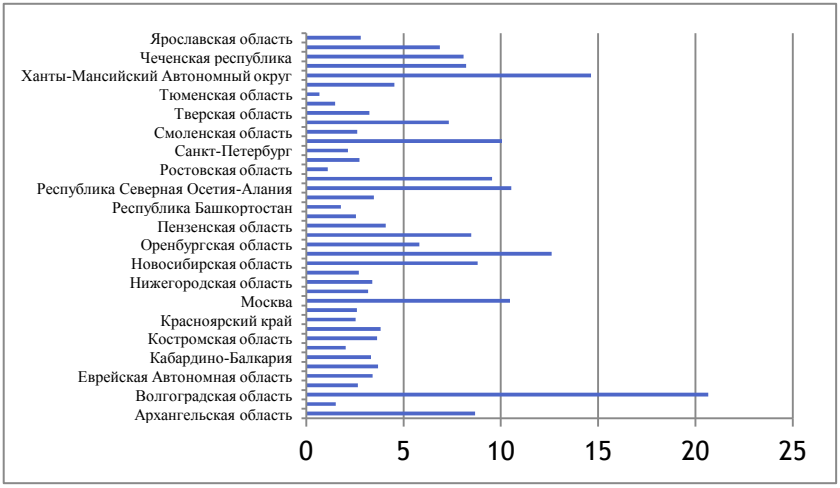


Рис. 1. Объем собранных данных из источников в Мб[1].

[1] На рисунке подписаны не все регионы ввиду их большого количества.

Возможные направления работы

Основные направления развития работы возникают в процессе появления ошибок или трудностей в ходе выполнения поставленной задачи. Можно выделить наиболее важные из них:

Создание универсального обработчика HTML страницы. Так как данная работа затрагивает только новости (или статьи и т.п.), то можно легко построить модель объекта – новость: заголовок, дата, содержание. Такая модель позволяет построить универсальный обработчик страницы. Рассматриваемый HTML документ содержит интересующую нас информацию, которую можно выделить, применяя некоторые эвристики или алгоритмы распознавания образов. Например, содержание статьи – это, в основном, наиболее крупный массив символов; дата представляет собой цифровые значения в определенном формате, а заголовок непосредственно находится «рядом» с содержанием. Также статья обычно находится в центре страницы, что тоже можно рассмотреть, как критерий выбора. Имея такой инструмент, нет никакой необходимости создавать отдельные краулеры для различных сайтов, даже если они не совсем похожи своей структурой.

Применение алгоритмов оптимизации для настройки параметров метода распараллеливания запросов. При оптимальной настройке метода сбора информации можно максимально сократить время обработки ресурса, что естественно является главным преимуществом автоматизации процесса.

Использование инструмента автоматического определения кодировки текста. Все сайты разрабатываются на разных платформах и с помощью различных технологий. Из-за этого все они имеют различную кодировку. Хорошо, когда она явно указана в структуре HTML страницы, однако, это не всегда так. При использовании инструмента автоматического определения кодировки текста можно сделать краулер наиболее универсальным в своем применении.

Выводы

Была решена задача автоматического сбора большого количества текстовых данных из различных источников.

Был реализован механизм распараллеливания обработки ресурса, что значительно увеличило скорость работы.

Можно реализовать универсальный механизм извлечения новостных данных из страницы, что позволит масштабировать данный подход и применять его при обработке новостных источников.

Разнородность технических характеристик и ограничений различных ресурсов порождает задачу оптимизации в подключении к таким источникам.

Список источников

1 . Паклин Н.Б., Орешков В.И. Бизнес-аналитика: от данных к знаниям // Спб.: Питер, 2009 - 624 с

2 A.H.F. Laender, B. A. Ribeiro-Neto, Juliana S.Teixeria. A brief survey of web data extraction tools // ACM SIGMOD Record 31(2), pp 84-93. 2002

3 . W3Schools. – http://www.w3schools.com

Извлечение семантических отношений из статей Википедии с помощью алгоритмов ближайших соседей

А. И. Панченко[2,1], С.А. Адейкин[1], А.В. Романов[1] и П.В. Романов[1]

{panchenko.alexander, adeykin90, jgc128ra, romanov4400}@gmail.com

[1] МГТУ им. Н.Э. Баумана, каф. Системы Обработки Информации и Управления

[2] Catholic University of Louvain, Center for Natural Language Processing (CENTAL)

Аннотация. В данной работе представлены методы извлечения семантических отношений из статей Википедии с помощью алгоритмов ближайших и взаимных ближайших соседей и двух метрик семантической близости. Мы производим анализ методов и приводим результаты их работы. Точность извлечения с помощью одного из методов достигает 83%. Кроме этого, мы представляем систему с открытым исходным кодом, которая эффективно реализует описанные алгоритмы.

Ключевые слова: семантические отношения, извлечение информации, Википедия, KNN, мера семантической близости

Введение

Существует множество типов *семантических отношений* между словами – синонимы, меронимы, антонимы, ассоциации и т.п. В рамках данной работы под семантическими отношениями понимаются синонимы, гиперонимы и ко-гиперонимы (слова имеющие общий гипероним). Подобные отношения успешно применяются в задачах *автоматической обработки текста*, таких как разрешение омонимии [1], расшире-

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16-18 марта, 2012.

ние поискового запроса [2], классификация текстовых документов [3] или создание вопросно-ответных систем [4]. Семантические отношения фиксируются в различного типа лингвистических ресурсах, к числу которых относятся, прежде всего, тезаурусы, онтологии, терминологические классификаторы и словари синонимов. Однако существующие ресурсы часто недоступны или недостаточны для конкретного приложения, предметной области или языка. При этом ручное создание требуемых семантических ресурсов – крайне дорогостоящий и трудоемкий процесс. В связи с этим, актуальной задачей является разработка методов автоматического извлечения семантических отношений.

Распространенный подход к извлечению отношений основан на лексико-синтаксических шаблонах, создаваемых вручную [5]. Недостатками данного подхода является сложность написания правил извлечения и применимость правил только для одного языка. Подходы, основанные на дистрибутивном анализе [6,7], не требуют ручной работы, но показывают невысокие результаты в задаче извлечения отношений [8]. В то же время недавно были предложены метрики семантической близости между словами на основе Википедии (www.wikipedia.org), показывающие отличные результаты [9,10,11]. Википедия привлекательна для анализа, так как она достаточно полно покрывает основные предметные области и языки, а также постоянно пополняется пользователями. Однако в предыдущих исследованиях мало внимания было уделено применению метрик, основанных на Википедии, для извлечения семантических отношений.

Данная работа восполняет этот пробел и фокусируется на применении подобных метрик для извлечения отношений. Цель предлагаемого в данной статье метода – найти для множества входных слов C (к примеру, терминов заданной предметной области) пары семантически связанных слов R. Рассматриваемые методы не возвращают тип найденной связи т. е. $R \subseteq C \times C$. Метод, предлагаемый в данной статье, характеризуется эффективностью, применимостью для языков доступных в Википедии и достаточной точностью. Новизна нашей работы по сравнению с существующими исследованиями и разработками заключается в следующем:

1) Предложены, реализованы и проанализированы новые *методы извлечения семантических отношений* из текстов статей Википедии, основанные на алгоритмах ближайших и взаимных ближайших соседей и двух метриках семантической близости слов (косинусе угла между векторами определений и количестве общих лемм в определениях).

2) Разработана программная система Serelex с открытым исходным кодом (лицензия LGPLv3), эффективно реализующая предложенные методы.

Методы извлечения семантических отношений

Данный раздел организован следующим образом. Сначала описываются данные, с которыми мы работаем, и способ их предварительной обработки. Затем обсуждаются алгоритмы извлечения семантических отношений и метрики семантической близости. В заключении описываются основные детали реализации системы Serelex.

Данные и их предварительная обработка

В качестве входных данных алгоритмы извлечения отношений получают множество определений D, для каждого из входных слов $c \in C$. Мы используем данные, доступные на DBPedia.org, для того чтобы построить множество определений английских слов (мы не включаем в этом множество словосочетания)[1]. Для каждого входного слова мы строим множество пар «слово;определение», где «слово» – это точное название статьи Википедии, а «определение» – текст первого параграфа этой статьи (аннотация к статье или т. н. extended abstract).

Аннотации к статьям были предварительно обработаны. Во-первых, из текста была удалена разметка и специальные символы. Во-вторых, был произведен морфологический анализ с помощью анализатора TreeTagger [12], в результате чего каждое слово было представлено в виде тройки «токен#ЧАСТЬ-РЕЧИ#лемма», к примеру «proved#VVN#prove». Приведем пример части определения термина «axiom», представленного в подобном формате:

axiom; in#IN#in traditional#JJ#traditional logic#NN#logic ,#,#, an#DT#an axiom#NN#axiom or#CC#or postulate#NN#postulate is#VBZ#be a#DT#a proposition#NN#proposition that#WDT#that is#VBZ#be not#RB#not proved#VVN#prove or#CC#or demonstrated#VVN#demonstrate but#CC#but considered#VVN#consider to#TO#to be#VB#be either#RB#either self-evident#JJ#self-evident ,#,#, or#CC#or subject#JJ#subject to#TO#to necessary#JJ#necessary decision#NN#decision .#SENT#.

Эксперименты, описанные в данной работе, были произведены на материале статьей, заглавие которых представляет собой одно слово без цифр и специальных символов. Данным критериям соответствовали 327167 статей Википедии. Для наших экспериментов мы подготовили два набора данных – малый (содержащий определения 775 слов (824Кб)) и большой (содержащий определения 327167 слов (237Мб)). Полученные «определения», скрипт предварительной обработки статей и результаты извлечения доступны по адресу:

[1] Мы использовали файл с расширенными аннотациями (long abstracts): http://downloads.dbpedia.org/3.7/en/long_abstracts_en.nt.bz2

http://cental.fltr.ucl.ac.be/team/~panchenko/def/.

Сенлар [19] и другие исследователи методов извлечения отношений [7] отмечают, что подходы, основанные на синтаксическом анализе, зачастую достигают более высоких результатов, чем подходы, использующие только морфологический анализ. Тем не менее, в нашей работе мы сознательно не используем синтаксический анализ по двум причинам. Во-первых, в силу его высокой вычислительной сложности. Во-вторых, в силу того, что применение глубокого лингвистического анализа делает метод извлечения менее робастным. Предыдущие исследования показывают, что парсеры для различных языков обладают радикально отличным качеством. Кроме этого, стандартные парсеры делают много ошибок при анализе имен собственных и технических терминов – лексических единиц, представляющих наибольший интерес при извлечении семантических отношений.

Алгоритмы извлечения семантических отношений

Методы извлечения семантических отношений, рассматриваемые в данной статье, основаны на компонентном анализе [13,14], принцип которого заключается в том, что семантически близкие слова имеют подобные определения. Предложенные алгоритмы используют одну из двух метрик подобия определений – количество общих слов [15] или косинус угла между векторами определений [16]. В качестве входных данных алгоритмы извлечения семантических отношений принимают множество слов C, между которыми необходимо вычислить отношения и их определения D. Допустим, что на вход алгоритму поступило 5 слов, т.е. $C = \{alligator, animal, building, house, telephone\}$. Тогда задача алгоритма – распознать множество семантических отношений $R = \{\langle alligator, animal\rangle, \langle building, house\rangle\}$ из всех 10 возможных пар слов.

Первый алгоритм вычисляет семантические отношения с помощью метода ближайших соседей KNN, второй – с помощью метода взаимных ближайших соседей MKNN (Mutual KNN). Единственный метапараметр алгоритмов – количество ближайших соседей k. Псевдокод алгоритмов представлен на Рисунке 1.

Работа алгоритмов состоит в следующем. Сначала вычисляется мера семантической близости всех возможных пар определений (строка 6). На основе вычисленного значения заполняем массив наиболее близких слов `Rmatrix` для каждого определения (строки 1-12). При этом мы поддерживаем число элементов этого массива равным `k` (количеству ближайших соседей) – это позволяет сильно сократить потребление памяти без потери информации о связности слов. После заполнения массива наиболее близких слов для каждого определения все что оста-

ется сделать для получения результирующего набора отношений R в методе KNN – просто заполнить выходное множество, для метода MKNN – дополнительно проверить для каждого определения входит ли оно в массив наиболее близких слов свой пары и если входит – добавить в результирующее множество (строки 13-21).

Сложность разработанных алгоритмов пропорциональна количеству поданных на вход слов $|C|$. Временная сложность равна $O(|C^2|)$, пространственная сложность также пропорциональна количеству ближайших соседей k и равна $O(k|C|)$.

```
R = ExtractRelations(C, D, k, isMKNN)
Input: C – слова, D – определения слов, k – количество ближайших соседей,
isMKNN – если true использовать алгоритм MKNN, иначе KNN
Output: R – множество семантических отношений <c_i,c_j> in C X C
   1. //Вычисление попарной близости между всеми словами C
   2. Rmatrix = void
   3. for i=0; i<count(C); i++ {
   4.         for j=i; j<count(C); j++ {
   5.                 // Вычисляем семантическую близость двух слов
   6.                 s_ij = similarity(D(i), D(j))
   7.                 // Сохраняем наиболее подобные слова
   8.                 if( count(Rmatrix(C(i))) < k || s_ij > min(Rmatrix(C(i)) ){
   9.                         Rmatrix(C(i)).addOrReplaceMin(C(j))
   10.                }
   11.        }
   12.}
   13.// Вычисление семантических отношений
   14.R = void
   15.foreach c_i in Rmatrix {
   16.        foreach c_j in Rmatrix(c_i) {
   17.                if(!isMKNN || Rmatrix(c_j) contains c_i){
   18.                        R.add(<c_i, c_j>)
   19.                }
   20.        }
   21.}
   22.return R
```

Рисунок 1. Псевдокод алгоритмов извлечения семантических отношений KNN и MKNN.

Метрики семантической близости слов

Функция `similarity` (строка 6) в алгоритмах KNN и MKNN вычисляет меру семантической близости между двумя словами на основе подобия их определений. Чем больше семантическая близость, тем более близок «смысл» слов. Мы используем две функции подобия определений. Первая метрика – количество общих лемм в определениях d_i, d_j слов c_i, c_j без учета совпадений стоп слов:

$$similarity(c_i, c_j) = \frac{2\left|(d_i \cap d_j) \,/\, stopwords\right|}{|d_i| + \left|d_j\right|}.$$

Здесь числитель равен количеству общих слов в двух определениях без учета стоп-слов; $|d_j|$ – количество слов в определении d_j; $stopwords$ – множество стоп-слов.

Вторая метрика – косинус угла между векторами определений $\boldsymbol{f}_i, \boldsymbol{f}_j$ представляющих слова c_i, c_j:

$$similarity(c_i, c_j) = \frac{\boldsymbol{f}_i \cdot \boldsymbol{f}_j}{\|\boldsymbol{f}_i\| \, \|\boldsymbol{f}_j\|} = \frac{\sum_{k=1,N} f_{ik} f_{jk}}{\sqrt{\sum_{k=1,N} f_{ik}^2} \sqrt{\sum_{k=1,N} f_{jk}^2}}.$$

Здесь f_{ik} – частота леммы c_k в определении d_i. Обе метрики подобия используют леммы (к примеру, animals#NNS#**animal**), не учитывают совпадения стоп-слов и учитывают совпадения лемм только со следующими частями речи: VV, VVN, VVP, JJ, NN, NNS, NP (существительные, глаголы и прилагательные).

Программный комплекс Serelex

Программное решение реализовано в виде консольного приложения на языке C++ и доступно для платформ Windows и Linux. Система состоит из классов определения, компонентного анализа, класса глобальных переменных, а также из нескольких вспомогательных классов и функций (см. Рисунок 2).

Основные функции программы заключаются в (1) загрузке файлов стоп-слов и слов, между которыми нужно найти отношения C; (2) загрузке с учетом стоп-слов и слов C файла дефиниций D; (3) вычислении семантической близости; (4) формировании списка наиболее близких слов R.

Для повышения быстродействия при загрузке дефиниций каждому слову сопоставляется уникальный числовой идентификатор и в дальнейшем вся работа по сравнению слов ведется с ним – это позволяет во много раз повысить быстродействие программы. В программном комплексе широко используется Стандартная библиотека шаблонов (STL) языка C++, что позволяет быстро, удобно и просто организовать хранение данных и работу с ними. Система Serelex имеет открытый исходный код, доступный на условиях лицензии LGPLv3 по адресу https://github.com/jgc128/Serelex.

Результаты

Мы исследовали работу алгоритмов KNN и MKNN с двумя описанными выше метриками близости и различными значениями количества ближайших соседей k (см. Рисунок 3). Полученные результаты свидетельствуют о практически линейном росте количества найденных от-

ношений в зависимости от параметра k для обоих алгоритмов. При этом количество найденных отношений мало зависит от используемой метрики подобия. Алгоритм KNN извлекает больше отношений, чем MKNN, при равном количестве ближайших соседей k. Это происходит потому что MKNN удаляет пары, которые не являются взаимными соседями, в отличие от KNN.

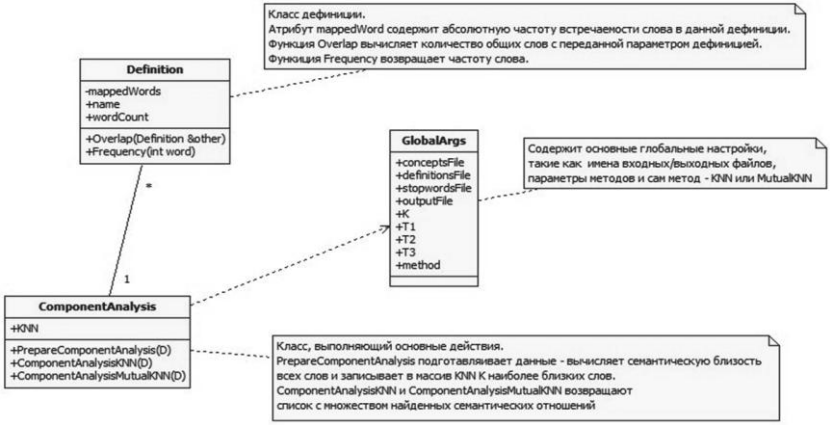


Рисунок 2. Основные классы системы извлечения семантических отношений Serelex.

Мы также провели оценку точности работы алгоритмов KNN и MKNN для $k = 2$ на множестве из 775 определений. Для этого мы разметили вручную файлы с извлеченными отношениям и вычислили точность извлечения как количество верных отношений к общему количеству извлеченных отношений. Результаты приведены в Таблице 1. Примеры извлеченных отношений между множеством из 775 слов с помощью алгоритма MKNN ($k = 2$) и количества общих слов в качестве метрики подобия приведены ниже[1]:
$R = \{\langle acacia, pine\rangle, \langle aircraft, rocket\rangle, \langle alcohol, carbohydrate\rangle, \langle alligator, coconut\rangle, \langle altar, sacristy\rangle, \ldots \langle object, library\rangle, \langle object, pattern\rangle, \langle office, crew\rangle, \langle onion, garlic\rangle, \langle saxophone, violin\rangle, \langle saxophone, clarinet\rangle \langle tongue, mouth\rangle, \ldots, \langle watercraft, boat\rangle, \langle watermelon, berry\rangle, \langle weapon, warship\rangle, \langle wolf, coyote\rangle, \langle wood, paper\rangle\}$.

В силу большого количества извлеченных отношений (см. Рисунок 3), оценка вручную качества извлечения для всех значений k затруднительна. Для больших значений k точность извлеченных отношений

[1] Все извлеченные отношения с помощью данной конфигурации – http://cental.fltr.ucl.ac.be/team/~panchenko/def/results-775/overlap_mknn_2.csv

должна уменьшаться. При использовании метода мы рекомендуем использовать $k \in [1; 10]$. В будущем, мы планируем использовать WordNet [17] и стандартные проверочные наборы семантических отношений, такие как BLESS [18], для более точной оценки качества извлечения.

Скорость работы алгоритма при всех проведенных оптимизациях достаточно высокая. К примеру, 755 дефиниций обрабатываются чуть меньше чем за 3 секунды на сервере Linux 2.6.32-cs-kernel с процессорами типа Intel(R) Xeon(R) CPU E5606@2.13GHz (программа не использует многопоточность); алгоритм KNN при метрике подобия «количество общих слов» обрабатывает файл с 327167 дефинициями за 3 дня 3 часа и 47 минут.

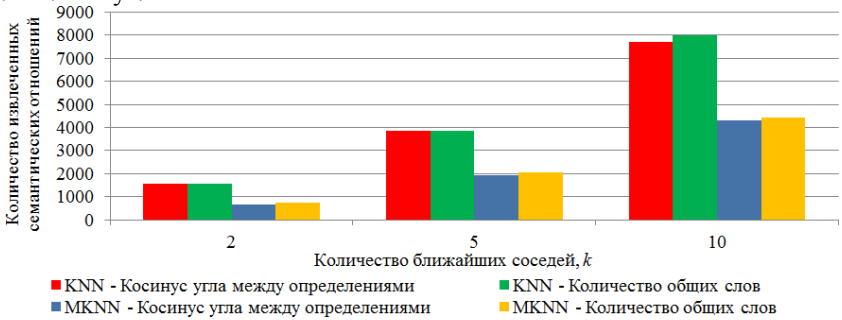


Рисунок 3. Зависимость количества извлеченных отношений от количества ближайших соседей k.

Таблица 1. Точность извлечения с помощью алгоритмов KNN и MKNN для $k = 2$ из 775 слов.

Алгоритм	**Мера подобия**	**Извлечено**	**Правильных**	**Точность**
KNN	Косинус угла	1548	1167	0.754
	Количество общих слов	1546	1176	0.761
MKNN	Косинус угла	652	499	0.763
	Количество общих слов	724	603	**0.833**

Обзор существующих методов

Сенлар [19] приводит обширный обзор методов извлечения семантических отношений основанных на дистрибутивном анализе и электронных словарях. Приводимые результаты оценивались на других лексиконах и поэтому не могут быть строго сравнены с нашими результа-

тами, однако дают некоторую информацию об эффективности других систем и подходов. Так, система автоматического построения тезауруса SEXTANT извлекает отношения между существительными с точностью около 75%. Метрики семантической близости, основанные на Веб, достигают точности в задаче выбора синонима из четырех вариантов около 74%.

Метод и система, наиболее похожая на нашу работу, – это WikiRelate!, предложенная Струбе и Понзетто в 2006 году [9]. Основные отличия нашего подхода и системы заключаются в следующем:

- Serelex извлекает семантические отношения, а WikiRelate! вычисляет только меру связности слов;
- Serelex реализует две метрики близости (косинус и количество общих слов), а WikiRelate! только количество общих слов. При этом в совпадение n-грамма в данной системе считается как n^2 общих слов;
- Serelex не используют решетку категорий Википедии;
- Serelex может быть использован для вычисления отношений между определениями не только Википедии, но и других источников дефиниций, если они представлены в соответствующем формате;
- Исходный код системы WikiRelate! недоступен, а бинарная версия доступна только для использования в научных целях, в то время как Serelex имеет открытый исходный код и коммерчески дружественную лицензию LGPLv3.

В силу того что WikiRelate! не извлекает отношения, мы не можем напрямую сравнить ее эффективность с эффективностью Serelex. WikiRelate! достигает корреляции с суждениями человека о семантической близости до 0.59, однако корреляция равна 0.22, если система используется только текст статей без решетки категорий Википедии.

В работах [10] и [11] были представлены альтернативные метрики семантической близости между словами на основе текстов Википедии. Однако эти системы менее похожи на Serelex чем WikiRelate!. В частности, слова в них представляются в пространстве из всех статей Википедии, в то время как Serelex использует пространство лемм. Накаяма и др. [20] предложили еще один метод, основанный на Википедии, который значительно отличается от нашего – авторы используют структуру гиперссылок между статей Википедии для извлечения отношений между словами. Наконец, Мильн и др. [21] предложили способы извлечения синонимов, гиперонимов и ассоциативных отношений из решетки категорий и других структурных и навигационных элементов Википедии.

Заключение

Мы предложили и исследовали методы извлечения семантических отношений из Википедии с помощью алгоритмов KNN и MKNN и двух метрик семантической близости. Предварительные эксперименты показали, что наилучшие результаты (83%) предоставляет метод, основанный на алгоритме MKNN и метрике подобия «количество общих слов». Мы также представили систему с открытым исходным кодом, которая реализует описанные алгоритмы.

Предложенные методы характеризуются вычислительной эффективностью и достаточной точностью. Большой охват лексикона достигается за счет того, что слова представляются текстами статей Википедии. Поэтому метод потенциально применим для извлечения отношений между 3.8 миллионами терминов на английском языке и 17 миллионами терминов на других 282 языках Википедии (при наличии соответствующих морфологических анализаторов). Кроме этого, система Serelex может быть использована для извлечения отношений и между другими источниками определений, такими как Викисловарь или традиционные словари, если определения представлены в соответствующем входном формате.

Основные направления дальнейшего исследования следующие: (1) применение разработанного метода для извлечения отношений на русском, немецком и французском языках; (2) повышение точности извлечения за счет применения алгоритмов анализа структуры полученного графа семантических отношений между словами.

Благодарности

Работа была выполнена под руководством Юрия Николаевича Филипповича в рамках курса «Семиотика информационных технологий» МГТУ им. Н.Э. Баумана. Исследования Александра Панченко поддерживаются стипендией IN.WBI фонда Wallonie-Bruxelles International. Мы благодарим Ольгу Морозову, Ивана Зеленцова, Марину Даньшину, Екатерину Выломову и двух анонимных рецензентов за ценные комментарии.

Список источников

1 . Patwardhan S., Pedersen T. Using WordNet-based context vectors to estimate the semantic relatedness of concepts. EACL , page 1-12, 2006

2 . Hsu M.H., Tsai M.F., Chen H.H. Query expansion with conceptnet and wordnet: An intrinsic comparison. Information Retrieval Technology, pages 1–13, 2006

3 . Tikk D., Yang J.D., Bang S.L. Hierarchical text categorization using fuzzy relational thesaurus. KYBERNETIKA-PRAHA, 39(5): 583–600, 2003.

4 . Sun R., Jiang J., Fan Y., Hang T., Tatseng K., Yen Kan C.M. Using syntactic and semantic relation analysis in question answering. In Proceedings of the TREC, 2005

5 . Hearst M.A., Automatic acquisition of hyponyms from large text corpora, Proceedings of the 14th conference on Computational linguistics COLING '92, 1992

6 . Lin D. Automatic retrieval and clustering of similar words. Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics, 768-774, 1998

7 . Heylen K., Peirsman Y., Geeraerts D., Speelman D. Modelling word similarity: an evaluation of automatic synonymy extraction algorithms. Proceedings of the Sixth International Language Resources and Evaluation (LREC'08), 3243-3249, 2008

8 . Curran J.R. and Moens M. Improvements in automatic thesaurus extraction. Proceedings of the ACL-02 workshop on Unsupervised lexical acquisition. 59-66, 2002.

9 . Strube, M. and Ponzetto, S.P., WikiRelate! Computing semantic relatedness using Wikipedia. Proceedings of the National Conference on Artificial Intelligence, 1419-1429, 2006.

10 . Gabrilovich E., Markovitch S. Computing semantic relatedness using wikipedia-based explicit semantic analysis. International Joint Conference on Artificial Intelligence, 12-20.2007.

11 . Zesch T., Müller C., Gurevych I. Extracting lexical semantic knowledge from wikipedia and wiktionary. In Proceedings of the LREC, pages 1646–1652, 2008.

12 . Helmut Schmid. Probabilistic Part-of-Speech Tagging Using Decision Trees. pages 44–49, 1994.

13 . Филиппович Ю.Н., Прохоров А.В., Семантика информационных технологий: опыты словарно-тезаурусного описания. Серия «Компьютерная лингвистика». М.:МГУП, 2002 http://it-claim.ru/Library/Books/CL/CLbook.htm

14 . Кобозева И. М. Компонентный анализ лексического значения. Лингвистическая семантика: 4-е изд. М.: Книжный дом «ЛИБРОКОМ», стр. 109-122, 2009.

15 . Banerjee S., Pedersen T. Extended Gloss Overlaps as a Measure of Semantic Relatedness,In Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, 2003.

16 . Jurafsky D., Manning H. M., An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Second Edition. 697-701, 2009.

17 . Fellbaum, C. WordNet. Theory and Applications of Ontology: Computer Applications, 231--243, Springer, 2010.

18 . Baroni, M. and Lenci, A. How we BLESSed distributional semantic evaluation. In Proceedings of GEMS 2011, 2011.

19 . Senellart P., Blondel V. D. Automatic Discovery of SimilarWords. Survey of Text Mining II.2008, 1, 25-44, Springer London, 2008.

20 . Nakayama K., Hara T., and Nishio S. Wikipedia Mining for an Association Web Thesaurus Construction, Web Information Systems Engineering – WISE, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 322-334, 2007.

21 . Milne D., Medelyan O., and Witten, I.H. Mining Domain-Specific Thesauri from Wikipedia: A Case Study. Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, 442-448, IEEE Computer Society, 2006

Алгоритм ГИС-анализа данных для оценки вероятности возникновения лесных пожаров в ИСДМ-Рослесхоз

А.С. Подольская, Д.В. Ершов, П.П. Шуляк

alexandra@ifi.rssi.ru

Центр по проблемам экологии и продуктивности лесов РАН, Москва, Россия

Аннотация. Статья посвящена решению задачи оценки возможности применения детерминированно-вероятностной методики нахождения вероятности возникновения лесных пожаров на федеральном уровне. Разработан и апробирован алгоритм ГИС-анализа данных, состоящий из необходимых и достаточных шагов для определения вероятности возникновения пожаров. Полученные количественные оценки вероятности возникновения пожара по периодам пожароопасного сезона представлены картографически и доступны на сайте ИСДМ-Рослесхоз.

Ключевые слова: ГИС-анализ, вероятность возникновения пожаров, детерминированно-вероятностный подход.

Введение

В проведенном исследовании была поставлена задача оценить возможность применения детерминированно-вероятностной методики нахождения вероятности возникновения лесного пожара на федеральном уровне. Определение вероятности выполнялось на основе многолетнего ретроспективного статистического анализа данных о произошедших пожарах и учета оперативных метеорологических данных в конкретный день пожароопасного сезона (ПС) [1].

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16-18 марта, 2012.

В качестве математической модели определения вероятности возникновения лесных пожаров используется модель, положенная в основу детерминировано-вероятностного подхода, который был предложен учеными из Томского государственного университета [2-5]. Метод определения вероятности возникновения лесных пожаров учитывает процессы сушки и зажигания слоя лесных горючих материалов, характеристики лесотаксационных описаний, грозовой активности и антропогенной нагрузки [3]. Согласно методу вероятность возникновения пожара определяют три составляющие – антропогенная, природная и вероятность по метеоусловиям [2]. Антропогенная и природная составляющие вычисляются как произведение соответствующих априорных вероятностей (1).

$$Pj=\sum[Pij(A)*Pij(FF/A) + Pij(M)*Pij(FF/M)]*Pij(C) \qquad (1)$$

где Pj – вероятность возникновения лесного пожара для j-го интервала на контролируемой лесной территории, индекс j соответствует дню ПС; индекс i – тип леса (по лесотаксационным описаниям соответствует территории отдельно взятого выдела); Pij(A) – вероятность антропогенной нагрузки; Pij(FF/A) – вероятность возникновения пожара от антропогенной нагрузки; Pij(M) – вероятность молниевого разряда; Pij(FF/M) – вероятность возникновения пожара от молниевого разряда; Pij(C) – вероятность по метеоусловиям.

Для нахождения вероятности по метеоусловиям авторы предлагают отличный от детерминировано-вероятностной методики подход, который будет описан ниже.

Априорные вероятности таких показателей как антропогенная нагрузка, возникновение пожара вследствие антропогенной нагрузки, вероятность молниевого разряда и возникновение лесного пожара от молниевого разряда в соответствии с методикой определяются через частоту событий (2).

$$P_{ij}(A) = N_a / N_{пс} \qquad P_{ij}(M) = N_м / N_{пс}$$
$$P_{ij}(FF/A) = N_{па} / N_{кп} \qquad P_{ij}(FF/M) = N_{пм} / N_{кп} \qquad (2)$$

где Na – число дней в периоде с антропогенной нагрузкой; Nпа – число пожаров от антропогенной нагрузки; Nм – число дней с молниевыми разрядами; Nпм – число пожаров от молниевых разрядов; Nпс – общее число дней в периоде пожароопасного сезона (ПС); Nкп – общее число пожаров в периоде.

Исходные данные

Оценка вероятности возникновения лесных пожаров проводилась для расширения линейки тематических продуктов, предоставляемых Информационной системой дистанционного мониторинга лесных пожаров (ИСДМ) Рослесхоз. Обозначенная система позволяет ежедневно получать, обрабатывать и интерпретировать информацию о лесопожарной обстановке на всей территории лесного фонда России [6-8].

ГИС-анализ оперировал геопространственными данными, предоставляемыми ИСДМ:

1. Данные метеонаблюдений – ежедневные (во время пожароопасного сезона) значения индекса пожарной опасности. Ежедневная метеорологическая информация с более чем двух тысяч наземных метеорологических станций по всей России доступна благодаря системе сбора и распространения информации о синоптической обстановке Гидрометеоцентра. База данных метеонаблюдений в ИСДМ-Рослесхоз содержит накопленную с 2006 года информацию. Извлеченная из БД информация сохранялась в формате DBASE.

2. Данные о пожарах с 1987 года - пространственно координированные очаги лесных пожаров, зарегистрированные на охраняемой региональными службами охраны леса от пожаров территории лесного фонда. Данные поступают от субъектов РФ в течение пожароопасного сезона с периодичностью не реже одного раза в месяц через диспетчерский центр ФГУ «Авиалесоохрана». На основе координат очагов пожаров создавались шейп-файлы точечной топологии (формат ESRI shape).

Алгоритм ГИС-анализа данных

1. Связь с БД лесных пожаров ФГУ «Авиалесоохрана». Из БД извлекается информация о дате обнаружения пожара и причина возникновения.

2. Определение проекции слоя пожаров. Используя возможности ГИС данные с очагами лесных пожаров переводятся в требующуюся проекцию: равнопромежуточную коническую с заданными параметрами центрального меридиана и стандартной параллели.

3. Пространственная идентификация пожаров относительно слоя зон ответственности метеостанций средствами ГИС-анализа (рис. 1). Поскольку метеоусловия являются ключевым фактором возгорания, элементарной пространственной единицей оценки вероятности возникновения пожара рассматривается зона ответственности метеостанции, представляющая собой полигон Тиссена. Сеть полигонов Тиссена - это геометрическая конструкция, образуемая относительно множества точек таким образом, что границы полигонов являются отрезками перпен-

дикуляров, восстанавливаемых к линиям, соединяющим две ближайшие точки [9]. Функциональные возможности ГИС позволяют построить для пунктов метеостанций нерегулярную сеть полигонов, которая лучше отражает характер реального расположения пунктов метеостанций, чем регулярная.

Рис. 1. Идентификация очагов пожаров относительно зон ответственности метеостанций

4. Классификация пожаров по причине возникновения. Благодаря наличию причины возникновения пожаров представляется возможной их классификация на природные и антропогенные (рис. 1). К природным отнесены пожары с причиной возникновения от гроз. Остальные пожары, возникшие по вине местного населения, от сельхозпалов, железных дорог МПС, на местах лесозаготовок и т.д., рассматривались как антропогенные.

5. Связь с БД метеонаблюдений. Выполняется SQL запрос на выборку значений индексов пожарной опасности на каждый день пожароопасного сезона конкретного года (с 01.04 по 31.10) для каждой метеостанции.

6. Вычисление априорных вероятностей через частоту событий. Расчет количества пожаров и дней для антропогенной и природной составляющих модели вероятности осуществлялся по периодам ПС. По-

жароопасный сезон разбивается на 6-ти дневные интервалы: текущие сутки + 5 суток вперед. Учитываются пожары, произошедшие за рассматриваемый временной интервал.

Для каждой метеостанции по всем периодам ПС за исследуемые годы находятся следующие количественные показатели: общее количество пожаров; число пожаров от антропогенного фактора; число пожаров от природного фактора; количество дней с антропогенными пожарами; количество дней с природными пожарами. Согласно формулам (2) по каждому показателю рассчитываются средние многолетние значения вероятности антропогенной и природной составляющих модели.

7. Нахождение вероятности пожара по метеоусловиям. Для нахождения вероятности пожара по метеоусловиям вместо физико-математической модели, описывающей время сушки лесных горючих материалов, предлагается использовать статистическую оценку значений индексов пожарной опасности (ИПО), при которых возникают лесные пожары. Этот подход предложен ввиду отсутствия на федеральном уровне повыдельных данных, необходимых для расчетов вероятности в соответствии с физико-математической моделью. Выражение для нахождения вероятности возникновения пожара по метеоусловиям выглядит следующим образом:

$$P(C) \approx \text{ИПОтекущий} / \text{ИПОкритический} \quad (3)$$

где ИПОтекущий – значения индекса пожарной опасности в каждый день пожароопасного сезона; ИПОкритический – значение ИПО, при котором лесные горючие материалы созрели до готовности к воспламенению и, таким образом, вероятность возникновения пожара определяется наличием антропогенного или природного источников огня. В связи с этим, принимается $P(C)=1$, если ИПОтекущий $\geq$ ИПОкритический.

8. Нахождение вероятности возникновения пожара в соответствии с формулой (1).

На рисунке 2 представлена принципиальная схема работы алгоритма ГИС-анализа данных для оценки вероятности возникновения пожаров. Схема отражает исходные данные, основные операции над ними и полученные результаты. Как видно из схемы состав баз данных, которыми располагает ИСДМ, расширяется за счет создания БД вероятностей возникновения пожаров, включающей антропогенную, природную составляющую и вероятность по метеоусловиям. В БД содержатся таблицы значений вероятностей для текущего года, которые будут использоваться для расчетов вероятности в следующем году. Таким образом, реализуется ежегодное обновление значений вероятности.

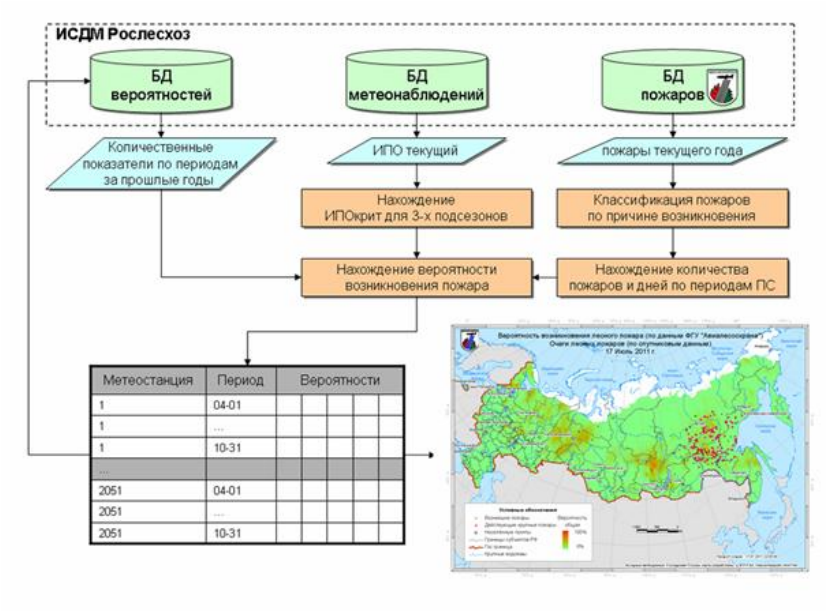


Рис. 2. Принципиальная схема работы алгоритма ГИС-анализа данных для оценки вероятности возникновения пожаров

Полученные результаты

Значения вероятностей помимо табличной формы представления отображаются пространственно, для чего используется интерполяция методом обратно взвешенных расстояний. В результате интерполяции строится растровое изображение, которое оформляется в виде карты вероятности возникновения лесного пожара. Доступ к ежедневно обновляемым картам осуществляется посредством информационного сервера ИСДМ (www.pushkino.aviales.ru).

Карты вероятности возникновения пожаров строятся на текущие и пять суток вперед по всей территории России. Для визуального анализа содержание карты текущих суток дополнено действующими крупными и возникшими очагами лесных пожаров, детектированными в указанные сутки по спутниковым данным. Прогнозная карта вероятности пожаров на следующие сутки содержит изолинии, показывающие общее число пожаров на 1 млн. га.

Оценка полученных значений вероятности возникновения пожаров проводилась методом построения гистограммы распределения количества зарегистрированных пожаров по вероятностям за пожароопасный сезон

2011 г. (рис. 3). Значения априорных составляющих вероятности (антропогенной и природной) были найдены по данным 1987-2010 гг. ИПО критический рассчитан по данным 2006-2010 гг.

Для каждого зарегистрированного в течение ПС 2011 года пожара определялась зона ответственности метеостанции, на территории которой он возник. Из БД вероятностей для даты регистрации пожара и зоны ответственности метеостанции известны априорные значения антропогенной и природной составляющих вероятности. Ежедневные метеоданные 2011 года предоставили значения ИПО текущего. Далее индекс сравнивался с ИПО критическим для этой метеостанции.

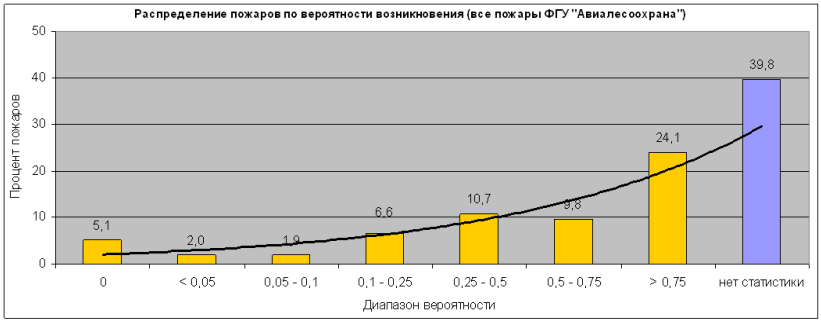


Рис. 3. Гистограммы распределения зарегистрированных пожаров по диапазонам вероятности за пожароопасный сезон 2011 г.

Оценка показала, что в дни с бóльшим значением вероятности регистрируется бóльший процент пожаров. Так, 24,1% всех пожаров за ПС 2011 года, зарегистрированных службой Авиалесоохрана, произошли в дни с вероятностью более 75% и только 2% пожаров зафиксированы в дни с малыми значениями вероятности (менее 5%). Как видно из гистограммы, процент зарегистрированных пожаров растет с увеличением значения вероятности. Часть пожаров (5%) не была спрогнозирована – вероятность их возникновения в дни регистрации была нулевой. Порядка 40 процентов пожаров произошли на территории зон ответственности метеостанций и в периоды ПС, для которых не была накоплена многолетняя статистика за прошлые годы.

Заключение

Проведенное исследование показало возможность применения модифицированного детерминированно-вероятностного подхода к оценке вероятности возникновения пожаров на федеральном уровне. Модифи-

кация затронула принцип нахождения вероятности пожаров по метеоусловиям.

Разработан и апробирован алгоритм ГИС-анализа данных, состоящий из необходимых и достаточных шагов для определения вероятности возникновения пожаров. Количественные оценки вероятности возникновения пожара по периодам ПС ежедневно представляются в виде карт на сайте ИСДМ-Рослесхоз.

Среди возможных направлений развития наиболее перспективными могут быть следующие:

1. увеличение количества метеостанций, для которых имеется многолетняя статистика по пожарам. Это достигается за счет использования данных о пожарах, детектированных со спутников.

Как видно из пространственного распределения пожаров, зарегистрированных региональными службами охраны лесов от пожаров (рис. 1), охраняемая территория не полностью покрывает все земли лесного фонда России. Для большинства покрытых лесом северных территорий Северо-Западного, Уральского, Сибирского и Дальневосточного федеральных округов нет данных о зарегистрированных очагах пожаров из-за отсутствия регулярных наблюдений наземными и авиационными методами. Для формирования статистики пожаров по этим регионам предлагается использовать многолетние данные спутниковых наблюдений. Низкая разрешающая способность действующих спутниковых систем и периодичность наблюдения территории не позволяют решать задачи обнаружения пожаров на ранних стадиях развития. Поэтому при оценке вероятности возникновения пожара необходимо учитывать эти факторы. Кроме того, причина возникновения таких пожаров не определена.

2. использование данных грозопеленгации для оценки природной составляющей вероятности возникновения пожаров. Посредством системы пеленгации молниевых разрядов, функционирующей в рамках ИСДМ Рослесхоз, доступны время и географические координаты зарегистрированных разрядов.

Прохождение грозового фронта способно привести к возникновению нескольких десятков лесных пожаров. После молниевого разряда очаг горения в стадии тления может находиться внутри слоя лесного горючего материала длительное время – от 5 до 10 суток в зависимости от вида горючего материала и условий погоды [10]. Следует отметить возможную удаленность пожаров с природной причиной возникновения от населенных пунктов и транспортных путей. Поэтому районирование территории по источникам возникновения пожаров в зависимости от плотности населения и транспортной инфраструктуры позволяет предположить возможную причину пожаров, детектированных по спутниковым данным. Районирование территории России и результаты исследо-

ваний по определению зависимостей количеств пожаров от молниевых разрядов приводятся в статье [11].

Список источников

1. Подольская А.С., Ершов Д.В., Шуляк П.П. Применение метода оценки вероятности возникновения лесных пожаров в ИСДМ-Рослесхоз // Современные проблемы дистанционного зондирования Земли из космоса: Физические основы, методы и технологии мониторинга окружающей среды, потенциально опасных явлений и объектов. Сборник научных статей. Том 8. Номер 1. – М.: ООО «ДоМира», 2011. - С. 118-126.

2. Барановский Н.В., Гришин А.М., Лоскутникова Т.П. Информационно-прогностическая система определения вероятности возникновения лесных пожаров // Вычислительные технологии, 2003, № 2. С. 16-26.

3.Барановский Н.В. Математическое моделирование наиболее вероятных сценариев и условий возникновения лесных пожаров //Автореферат диссертации на соискание ученой степени кандидата физико-математических наук. Томск, 2007. 20 с.

4. Гришин А.М., Фильков А.И. Прогноз возникновения и распространения лесных пожаров: Монография. – Кемерово: Изд-во Практика, 2005. – 202 с.

5. Кузнецов Г.В., Барановский Н.В. Прогноз возникновения лесных пожаров и их экологических последствий. Новосибирск: Изд-во СО РАН, 2009. 301 с.

6. Ершов Д.В.и др. Российская система дистанционного мониторинга лесных пожаров // ArcReview, 2004. – №4. – С.21-23.

7. Беляев А.И., Коровин Г.Н., Лупян Е.А. Состояние и перспективы развития Российской системы дистанционного мониторинга лесных пожаров // Современные проблемы дистанционного зондирования Земли из космоса. Физические основы, методы и технологии мониторинга окружающей среды, потенциально опасных явлений и объектов. Сборник научных статей. Выпуск 3. Москва: ООО «Азбука-2000». 2006. т. 1. С. 341-350.

8. Галеев А.А. и др. Система оперативного доступа удаленных пользователей к информационным ресурсам информационной системы дистанционного мониторинга лесных пожаров // Современные проблемы дистанционного зондирования Земли из космоса. Сборник научных статей. Выпуск 3. Том I. – М. – ООО «Азбука-2000», 2006. – С. 351-358.

9. Цветков В.Я. Геоинформационные системы и технологии. М.: Финансы и статистика, 1998. 288 с.

10. Иванов В.А. Методологические основы классификации лесов Средней Сибири по степени пожарной опасности от гроз // Автореферат диссертации на соискание ученой степени доктора сельскохозяйственных наук. Красноярск, 2006. – 42 с.

11. Подольская А.С., Ершов Д.В., Малинников В.А. Исследования оценки риска возникновения лесных пожаров от молний // Известия высших учебных заведений. Геодезия и аэрофотосъемка. – М., 2009. №2 – С. 3-11.

Автоматическое снятие морфологической неоднозначности при разметке корпуса текстов

Екатерина Протопопова

protoev@gmail.com

Санкт-Петербургский Государственный Университет (СПбГУ)

Аннотация. Автоматическое разрешение морфологической неоднозначности – одна из основных проблем морфологической разметки текстов. Статья представляет собой обзор существующих методов автоматического снятия морфологической омонимии при разметке корпусов текстов. Рассматриваются детерминированные, вероятностные и комбинированные алгоритмы. Кроме того, в статье сделана попытка выбрать один из описанных алгоритмов в качестве основы для разработки собственного инструмента для снятия неоднозначности в корпусе, работа по созданию которого ведется в данный момент кафедрой математической лингвистики СПбГУ.

Ключевые слова: морфологическая аннотация, корпус текстов, natural language processing, морфология, машинное обучение, вероятностные модели

Введение

Проблема разрешения морфологической неоднозначности – одна из ключевых проблем автоматического морфологического анализа. Наиболее сложным случаем морфологической омонимии является межчастеречная омонимия, например *трем* (лемма: три) и *трем* (лемма: тереть). Существующие подходы к снятию морфологической неоднозначности можно разделить на три группы: детерминированные, вероятностные и

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16-18 марта, 2012.

комбинированные. Следует учитывать, что большинство алгоритмов разрешения морфологической омонимии были разработаны в рамках создания морфологических анализаторов. Это, в частности, определяет и выбор одного из вышеназванных подходов: например, детерминированные алгоритмы удобнее применять для языков с относительно «простой» морфологией.

В данной работе предполагается рассмотреть некоторые существующие алгоритмы снятия морфологической омонимии и сравнить эффективность их работы (accuracy). Поскольку не все описываемые алгоритмы были использованы при работе с русскоязычным текстом, предполагается оценить возможность подобного их применения. В дальнейшем мы планируем использовать одну из описанных моделей для разметки корпуса текстов на русском языке и сравнить её точность с уже полученными результатами.

Описание существующих алгоритмов

В целом подходы к разрешению морфологической неоднозначности обычно разделяются на детерминированные и вероятностные. Первые основаны на синтаксических правилах: система проводит синтаксический анализ текста, а затем с помощью правил удаляет запрещенные последовательности грамматических классов или запрещенные синтаксические связи. Начало использования данных алгоритмов связано с началом развития машинной лингвистики (1960-е гг.), в СССР, в частности, первый широко известный алгоритм появился при разработке лингвистического процессора ЭТАП.

Вероятностные алгоритмы появились значительно позднее; они основаны в первую очередь на алгоритмах машинного обучения и статистических моделях (HMM и VMM [1]). Такие алгоритмы могут значительно различаться в зависимости от используемой модели, однако все они требуют наличия уже размеченного корпуса для обучения системы и имеют довольно высокую вычислительную сложность.

Довольно популярным является комбинированный подход – этот алгоритм сперва использует машинное обучение для извлечения из уже размеченного корпуса некоторых правил, а затем применяет их к неразмеченным текстам. Наиболее известная комбинированная модель – морфологический анализатор Брилла [2] или Transformation-Based Learning Algorithm. Этот алгоритм будет рассмотрен ниже вместе с некоторыми другими примерами как детерминированных, так и вероятностных моделей.

1. Детерминированные алгоритмы

Алгоритмы этого типа используются в лингвистических системах довольно давно. Одним из ярких примеров является блок так называемого «предсинтаксического анализа» лингвистического процессора ЭТАП (в версии «ЭТАП-2» [3]). В ориентированной на машинный перевод системе снятие омонимии (как лексической, так и морфологической) осуществляется по правилам. Связано это, вероятно, с тем, что в английском и французском языках, с которыми преимущественно и работает система, широко развита межчастеречная омонимия (например, «глагол/причастие» или «предлог/наречие»). Этот вид омонимов обрабатывается с помощью правил, описывающих левый контекст анализируемого омонима. Так, например, омонимия английских глаголов и вторых причастий разрешается следующим образом:

Тег «глагол» стирается в следующих случаях:

А) когда рассматриваемая словоформа очевидным образом входит в состав аналитической глагольной формы, то есть слева от неё на небольшом расстоянии стоит глагол HAVE или BE; например,

(have brought 'принес' ⇒)
$HAVE_{mf}\{BRING_{pst}/BRING_{pp}\} \Rightarrow HAVE_{mf}\ BRING_{pp}$;

где mf – основная форма глагола, pst – глагол в прошедшем времени, pp – причастие II.В квадратные скобки заключаются омонимичные варианты.

В) когда наличие слева от обрабатываемой словоформы артикля или детерминатива однозначно определяет её как причастие в функции определения перед существительным; ср.

(the created system 'созданная система' ⇒)
$THE\ \{CREATE_{pst}/CREATE_{pp}\}\ SYSTEM_{sg} \Rightarrow THE\ CREATE_{pp}\ SYSTEM_{sg}$.

где sg – имя существительное в единственном числе.

Для русского языка омонимия разрешается на уровне синтаксических связей: строится полный набор связей, а затем удаляются недопустимые сочетания.

2. Вероятностные алгоритмы

Широко распространён вероятностный алгоритм, основанный на скрытой марковской модели. Применительно к английскому языку (с несильно развитым словообразованием) точность этого алгоритма довольно высока – в некоторых случаях достигает 98%. Попытки использования схожих алгоритмов для русского языка предпринимались лишь

несколько раз. Одну из таких моделей представляет алгоритм, описанный в [4]: «модуль Tagger представляет собой скрытую марковскую модель, способную запоминать последовательности длиной от 4 до 6 синтаксических единиц. Коэффициенты вероятностей выбора морфологических значений вырабатываются в цепи путем обучения марковской модели на размеченном тексте. Каждой словоформе в размеченном тексте присваивается морфологическая помета (tag). Для того, чтобы сократить размеры как самой скрытой модели, так и размеченного текста, необходимого для обучения, используются усеченные морфологические пометы, которые позволяют сократить комбинаторно возможные варианты синтаксических контекстов».

Другой алгоритм (Trigram), основанный на CMM, описан и проиллюстрирован более подробно в работе [5]. Он состоит из двух частей: трёхграммной модели для тегов (учитывается тег данного слова и два предыдущих тега; $p(t_i \mid t_{i-2}, t_{i-1})$) и биграммной модели для словоформ (лексическая вероятность; $p(w_i \mid t_i, t_{i-1})$). Для каждого входного предложения Trigram определяет наиболее вероятные теги каждого слова по следующим формулам:

$$P(T) = \Pi_i=3..n\ p_{smooth}\ (t_i \mid t_{i-2}, t_{i-1})$$

$$P(W|T) = \Pi_i=3..n\ p_{smooth\_lex}\ (w_i \mid t_i, t_{i-1}).$$

Вероятности p_{smooth} строятся с помощью сглаживания («smoothing»):

$$p_{smooth}\ (t_i \mid t_{i-2}, t_{i-1}) = \lambda_3\ p(t_i \mid t_{i-2}, t_{i-1}) + \lambda_2\ p(t_i \mid t_{i-1}) + \lambda_1\ p(t_i),$$

где $\lambda_1+\lambda_2+\lambda_3 = 1$. Сглаживание используется для того, чтобы иметь возможность работать и со случаями, когда комбинация $<t_i, t_{i-2}, t_{i-1}>$ не встречалась в обучающем корпусе (то есть чтобы алгоритм мог по тегам t_{i-2} и t_{i-1} приписать исследуемому слову тег t_i).

Схожим образом проводится сглаживание для биграммной модели. Формулы заимствованы из работы по созданию сходных алгоритмов для чешского языка [6]. В качестве обучающего множества используется выборка из Национального корпуса русского языка, объем множества в работе не указан. В качестве морфологической разметки изпользуется стандарт морфологических модулей компании «АОТ». В работе также приводится оценка эффективности работы алгоритма: в сравнении с модулем анализ именных групп Synan [7] неоднозначность разрешается правильно примерно в 94,5% случаев при полном анализе и в 98,65% при частичном анализе. Точность работы алгоритма при сравнении с вероятностной моделью на основе нормализующих подстановок (описана ниже) составляет 97,18%. В статье приведены сравнительные таблицы эффективности различных протестированных алгоритмов, в основном эта точность составляет не менее 90%.

3. Комбинированные алгоритмы

Наиболее известным алгоритмом этого типа является морфологический анализатор Брилла, описанный в 1995 году. Он использует набор правил в сочетании с вероятностной моделью. Первый морфологический анализатор, разработанный Бриллом и описанный в работе [8], был основан исключительно на правилах, но по точности превосходил существовавшие в то время вероятностные (stochastic) модели. Позднее алгоритм был доработан на основе модели обучения, названной Transformation-Based Learning Algorithm. Обучение работает следующим образом: на вход подается неразмеченное множество текстов, которое проходит первичную обработку с помощью автоматического разметчика. Затем результат обработки сравнивается с идеально (вручную) размеченным текстом, и обучаемый алгоритм выводит из этого сравнения правила (rewrite rules) и условия (a triggering environment) трансформации тегов, а далее и собственно трансформации, например

Change the tag of a word from VERB to NOUN if the previous word is a DETERMINER.

Полученная информация снова применяется для разметки текста. В качестве обучающего множества использовалась выборка объемом 950 тысяч словоупотреблений из Penn Treebank Tagged Wall Street Journal Corpus. Алгоритм вывел 243 правила для разметки неизвестных слов и 447 контекстных правил. В статье отмечается высокая точность разметки (96.6%) и скорость обработки данных. Помимо английских корпусов, алгоритм Брилла используется для разметки корпусов текстов на других европейских языках, например, на датском.

Попытка применить комбинированный алгоритм для снятия морфологической неоднозначности в текстах на русском языке осуществляется в работе [9]. При создании данного алгоритма использовались простая вероятностная модель и построенные самим алгоритмом правила. Снятие омонимии происходит следующим образом. По результатам морфологического разбора входного текста для каждого омонима и его ближайших соседей строятся нормализующие подстановки – правило, по которому на основании трех последних букв слова для него определяются соответствующие леммы. Затем из словаря контекстов выбирается наиболее вероятная для данного контекста лемма. Точность данного алгоритма, обученного на выборке из НКРЯ, составляет порядка 94-95%.

Постановка задачи

Целью данной работы является выбор модели для разработки алгоритма автоматического снятия морфологической неоднозначности. Данные о точности результатов работы алгоритмов приводятся далеко не во всех указанных выше статьях, и это существенно затрудняет выбор определенного алгоритма для использования в разметке корпуса. Достаточно очевидно, что разработка простого детерминированного алгоритма для русского языка – задача весьма трудоемкая. С другой стороны, вероятностная модель, скорее всего, будет нуждаться в добавлении написанных вручную правил.

Из описанных выше алгоритмов наиболее эффективным и простым представляется алгоритм Брилла, который мы и предполагаем использовать для снятия морфологической неоднозначности при разметке русскоязычного текста. Следует учитывать, что непосредственное применение алгоритма, разработанного для английского языка, к русскоязычному тексту вряд ли оправдано. Для обучения алгоритма предполагается использовать выборку из Национального корпуса русского языка (ruscorpora.ru). Алгоритм Брилла первоначально обучался на выборке объемом около миллиона словоупотреблений, поэтому первое тестирование предполагается проводить на том же объеме, но, возможно, с учетом особенностей русской морфологии этот объем будет необходимо увеличить. В дальнейшем следует подсчитать необходимый и достаточный объем корпуса для обучения данного алгоритма.

Открытым вопросом также остается выбор набора тегов для разметки текста. Экспериментальные данные, свидетельствуют о том, что при увеличении набора входных тегов снижается как скорость работы программы, так и её точность (см. [4]). Одной из первых задач является тестирование алгоритма с тегами для частей речи: предполагается использовать частеречные теги НКРЯ. Предполагается даже, что набор этих тегов может быть сокращен без ущерба для эффективности работы анализатора[1]. Однако на основе этих тегов алгоритм не будет работать с «внутричастеречной» омонимией, например *большой* (прилагательное ед.ч. м.р. им.п./прилагательное ед.ч. ж.р. род.п./прилагательное ж.р. предл.п.). Ожидается, что в процессе разработки алгоритма описанные выше проблемы будут решены.

[1] Мы предполагаем не использовать те частеречные теги, которые даже теоретически не могут вступать в омонимичные отношения (например, тег «местоименное наречие» можно заменить тегом «наречие»), а также те, которые затрудняют морфологический анализ (например, омонимия слов категории состояния и наречий довольно трудно разрешима без опоры на семантический контекст).

Выводы

Описана проблема морфологической неоднозначности и основные методы её автоматического разрешения – детерминированный, вероятностный и комбинированный методы.

Приведены некоторые данные об эффективности работы описанных алгоритмов.

Из рассмотренных алгоритмов выбран алгоритм морфологического анализатора Брилла, как один из наиболее быстрых и точных.

При создании алгоритма снятия морфологической неоднозначности для русского языка следует принимать во внимание следующие проблемы: неопределенность и вариативность набора тегов, сильно развитое русское словообразование.

Список источников

1 Hidden Markov Model (Скрытая марковская модель) и Variable Memory Markov Model (термин не переведен) – см., например, Xuedong Huang, M. Jack, Y. Ariki . Hidden Markov Models for Speech Recognition. - Edinburgh University Press, 1990.

2 Brill E. Transformation-based error-driven learning and natural language processing: a case study in POS-tagging. // Computational Linguistics. – V. 21, № 4. – MIT Press, 1995.

3 Апресян Ю.Д. и др. Лингвистическое обеспечение системы ЭТАП-2. / Апресян Ю.Д., Богуславский И.М., Иомдин Л.Л., Лазурский А.В., Перцов Н.В., Санников В.З., Цинман Л.Л. – М., 1989.

4 Ножов И.М. Морфологическая и синтаксическая обработка текста (модели и программы). Кандидатская диссертация. – М., 2003.

5 Сокирко А.В., Толдова С.Ю. Сравнение эффективности двух методик снятия лексической и морфологической неоднозначности для русского языка (скрытая модель Маркова и синтаксический анализатор именных групп). URL: http://www.aot.ru/docs/RusCorporaHMM.htm

6 Jan Hajic, Pavel Krbec, Pavel Kveton, Karel Oliva, and Vladimr Petkevic. Serial Combination of Rules and Statistics: A Case Study in Czech Tagging. In Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL 2001),Toulouse, France, 2001.

7 Гершензон Л.М. и др. Синтаксический анализ в системе РМЛ./ Гершензон Л.М, Ножов И.М., Панкратов Д.В., Сокирко А.В. URL: http://www.aot.ru/docs/synan.html

8 Brill E. A simple rule-based part-of-speech tagger // Speech and Natural Language: Proceedings of a Workshop. - Harriman, New York, 1992.

9 Зеленков Ю.Г. и др. Вероятностная модель снятия морфологической омонимии на основе нормализующих подстановок и позиций соседних слов./ Зеленков Ю.Г., Сегалович И.В., Титов В.А. // Компьютерная лингвистика и интеллектуальные технологии. Труды международного семинара Диалог'2005. – М., 2005.

Распознавание образов при помощи динамических NK-сетей, состоящих из бинарных динамических элементов

Д. М. Пучкова

dpuchkova@gmail.com

ИКИС, ТГНГУ, Тюмень, Россия

Аннотация. Настоящая работа представляет новый метод обработки изображений при помощи трехмерной нейросети, состоящей из бинарных логических элементов. Уникальной особенностью данной сети является самоорганизация через период работы сети и демонстрация уникальных признаков изображения через полпериода работы. В процессе работы исследован эффект самоорганизации NK-сети бинарных логических элементов, показана возможность его применения в системах распознавания образов и анализа изображений, проанализированы отличительные черты и преимущества предлагаемого метода. Описанный способ обработки изображений может быть применен для распознавания образов различных фигур и объектов а также для решения задач отделения сигнала от шума.

Ключевые слова: NK-сеть, самоорганизация, распознавание образов, нейросистема, нейросеть, бинарные логические элементы.

Целью данной работы является представление нового метода обработки изображений при помощи трехмерной (в виде гладкого многообразия) NK-сетью бинарных логических элементов или нейросетью. Данная сеть представляет собой набор пороговых элементов, объединенных в тор, каждый из которых взаимодействует с окружающими его

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16-18 марта, 2012.

восьмью элементами. Каждый элемент может принимать два сигнала: 1 и 0. При этом передаточной функцией элемента является строгая дизъюнкция:

$$y_n = \sum_{i=1}^{8} x_n^i \pmod 2, \qquad n > 3, \tag{1}$$

где:

y_n - сигнал на выходе n-го элемента,

x_n^i - сигнал на i-м входе n-го элемента.

Работа производится на сетях, состоящих из $(2^n+1)\times(2^n+1)$ элементов (развертка сети представляет собой квадратную матрицу из $(2^n+1)\times(2^n+1)$ элементов), поскольку на таких размерностях матрицы при заданном выше числе входов наблюдается описанный ниже эффект.

На указанную сеть проецируется анализируемое изображение путем возбуждения (установки в значение 1) части его элементов таким образом, что возбужденные элементы на фоне всей матрицы формируют образ, идентичный поданному. Каждый элемент сети может быть однозначно идентифицирован по двум координатам (вертикальной и горизонтальной, отсчитывая от верхнего левого элемента), заданным на матрице, представляющей собой развертку тора: $i = \overline{0, N}$, $j = \overline{0, N}$, где $N = 2^n$. Анализ изображения производится в результате потактовой смены состояний элементов сети по следующим формулам (под суммой подразумевается суммирование по модулю 2):

$$a_{i,j}(p) = \sum_{k=-1,1}\sum_{l=-1}^{1} a_{i+k,j+l}(p-1) + \sum_{l=-1,1} a_{i,j+l}(p-1), \quad 0 < i, j < N, \tag{2}$$

$$a_{0,j}(p) = \sum_{k=n,1}\sum_{l=-1}^{1} a_{k,j+l}(p-1) + \sum_{l=-1,1} a_{0,j+l}(p-1), \quad 0 < j < N, \tag{3}$$

$$a_{i,0}(p) = \sum_{l=n,1}\sum_{k=-1}^{1} a_{i+k,l}(p-1) + \sum_{k=-1,1} a_{i+k,0}(p-1), \quad 0 < i < N, \tag{4}$$

$$a_{N,j}(p) = \sum_{k=N-1,0} \sum_{l=-1}^{1} a_{k,j+l}(p-1) + \sum_{l=-1,1} a_{N,j+l}(p-1) \quad ,0 < j < N \quad (5)$$

$$a_{i,N}(p) = \sum_{l=N-1,0} \sum_{k=-1}^{1} a_{i+k,l}(p-1) + \sum_{k=-1,1} a_{i+k,N}(p-1) \quad , 0 < i < N \quad (6)$$

$$a_{0,0}(p) = \sum_{k=n,1} \sum_{l=n,0,1} a_{k,l}(p-1) + \sum_{l=n,1} a_{0,l}(p-1) \quad , \quad (7)$$

$$a_{N,0}(p) = \sum_{k=N-1,0} \sum_{l=N,0,1} a_{k,l}(p-1) + \sum_{l=N,1} a_{0,l}(p-1) \quad , \quad (8)$$

$$a_{0,N}(p) = \sum_{k=N,1} \sum_{l=N-1,n,0} a_{k,l}(p-1) + \sum_{l=N-1,0} a_{0,l}(p-1) \quad , \quad (9)$$

$$a_{N,N}(p) = \sum_{k=N-1,0} \sum_{l=N-1,N,0} a_{k,l}(p-1) + \sum_{l=N-1,0} a_{N,l}(p-1) \quad , \quad (10)$$

где:

$a_{i,j}(p)$ – значение выходного сигнала элемента, имеющего координаты (i, j) на p-м такте. Т.е. состояние элемента на данном такте определяется состоянием восьми его соседей на предыдущем такте. При проведении анализа используется 2^n тактов. При проведении 2^{n-1} тактов сеть формирует из возбужденных элементов образ, содержащий признаки анализируемого изображения. Признаки отображаются в область (контур), идентичный области (контуру) анализируемого изображения, и расположение этих признаков и первоначального изображения связаны между собой. Развертку сети можно условно разделить на четыре квадранта, разграниченных полосами элементов с координатами $(i,2^{n-1})$ и $(2^{n-1}, j)$. Следует отметить, что в различных квадрантах находятся различные признаки. Так, второй квадрант содержит признаки вертикальных размеров и очертаний исходного изображения, третий – о количестве и расположении углов в исходном изображении, четвертый – признаки горизонтальных размеров и очертаний исходного изображения. Полученные таким образом признаки ярко выражены и являются уникальными для данного изображения. Примеры исходных изображений (подаваемых в первый квадрант) и их признаков представ-

лены на рис.1 (а,б). Следует отметить, что в текущей работе исследовалась матрица 128x128 элементов.

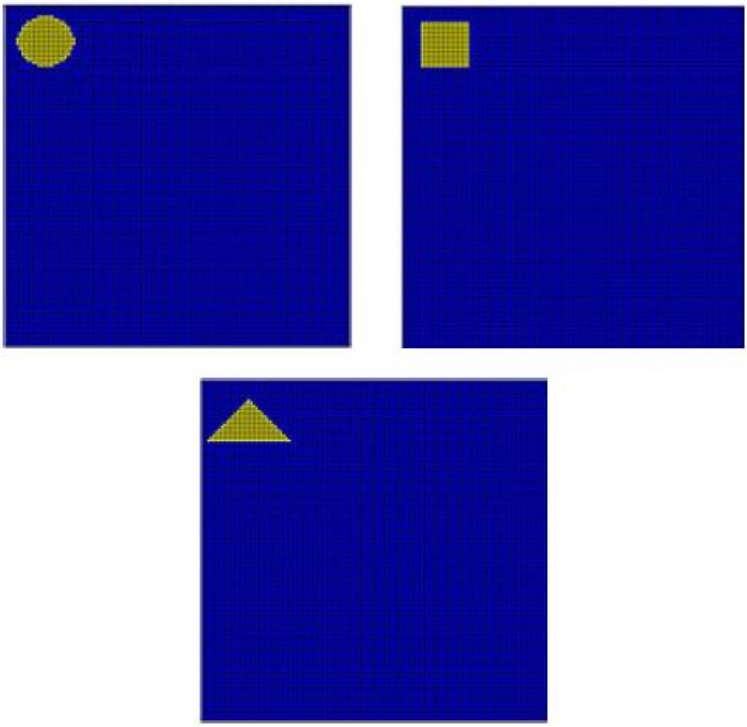


Рис.1.а. Примеры исходных изображений (подаваемых в первый квадрант).

Анализ признаков может осуществляться как визуально, так и при помощи стандартной схемы распознавания образов (например, на основе перцептрона Розенблатта). Другим способом анализа является вычисление спектра зависимости числа возбужденных элементов в сети от числа тактов. Этот спектр также является уникальным для каждого изображения и позволяет говорить об изменении свойств изображения и характере этих изменений по данным сравнения спектров этих изображений. Следует отметить, что вычисление данного спектра и его анализ требует меньших вычислительных затрат, чем при спектральном анализе исходного изображения, поскольку рассчитывается по небольшому числу точек (порядка количества тактов) и является одномерным, а не двумерным или трехмерным.

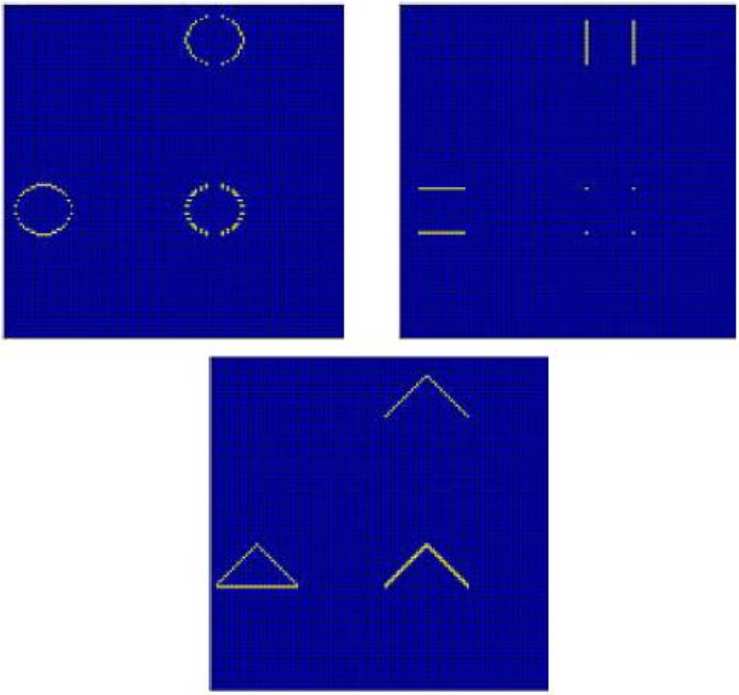


Рис.1.б. Примеры признаков, соответствующих подаваемым изображениям.

После прохождения 2^n тактов сеть самоорганизуется и отображает первичное изображение, закончив один цикл преобразований. При зашумлении исходного изображения признаки также видоизменяются. Подобные фигуры имеют схожие спектры и признаки. Спектры одного цикла работы сети для рассмотренных примеров исходных изображений, представлены на рис.2. Изображение, признаки и спектр уменьшенного треугольника представлены на рис.3. Изображение, признаки и спектр зашумленной фигуры представлены на рис.4.

Следует отметить, что в предлагаемом методе работа производится с целыми числами, все величины являются целыми, а не вещественными числами, и, следовательно, требуют меньших ресурсов памяти и обеспечивают большую скорость обработки. Более того, поскольку работа производится не просто с целыми числами, а с числами 0 и 1, то для их хранения и передачи достаточно одного бита, что дает возможность весьма простой и эффективной аппаратной реализации всего комплекса, причем на различной элементарной и самотехнической базе

(электронной, оптической, оптоэлектронной). В процессе работы используются бинарные логические и арифметические операции, требующие гораздо меньше вычислительных затрат, чем реализация функций нечеткой логики или спектрально-корреляционного анализа изображений.

Таким образом, в работе исследован эффект самоорганизации NK-сети бинарных логических элементов, показана возможность его применения в системах распознавания образов и анализа изображений, проанализированы отличительные черты и преимущества предлагаемого метода.

Описанный способ обработки изображений может быть применен для распознавания образов различных фигур и объектов а также для решения задач отделения сигнала от шума.

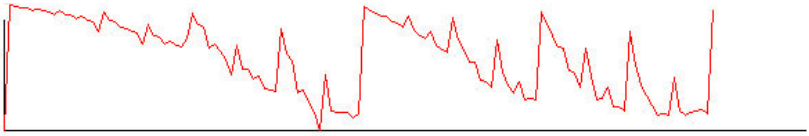


Рис.2.а. Спектр одного цикла работы сети анализа исходного изображения в виде круга.

Рис.2.б. Спектр одного цикла работы сети для анализа исходного изображения в виде квадрата.

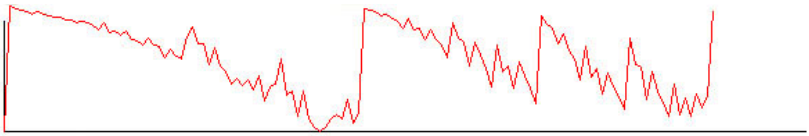


Рис.2.в. Спектр одного цикла работы сети анализа исходного изображения в виде треугольника.

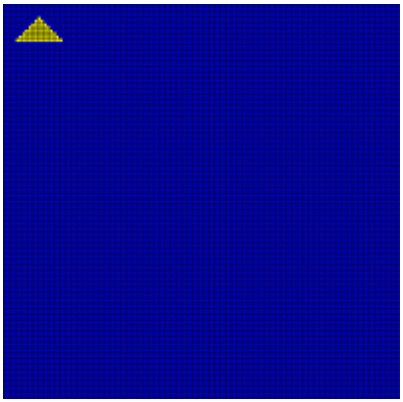


Рис.3.а. Изображение уменьшенного треугольника.

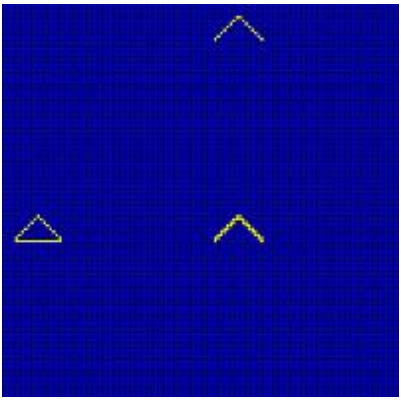


Рис.3.б. Признаки уменьшенного треугольника.

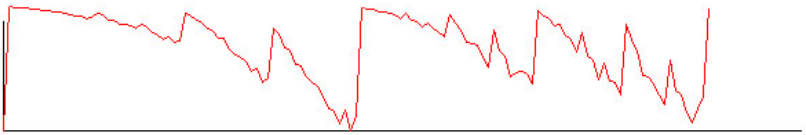


Рис.3.в. Спектр одного цикла работы сети для анализа исходного изображения в виде уменьшенного треугольника.

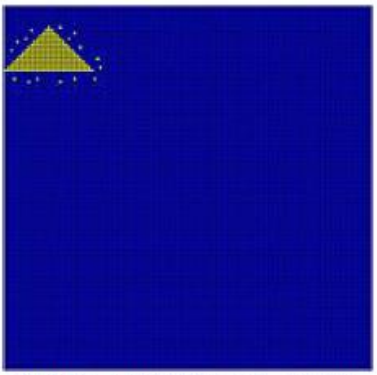


Рис.4.а. Изображение зашумленного треугольника.

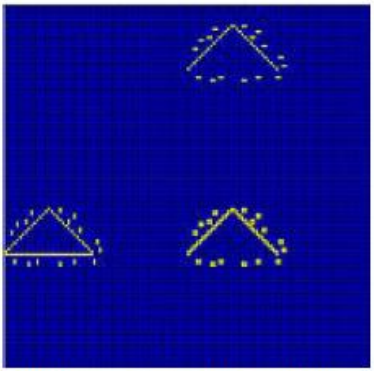


Рис.4.б. Признаки зашумленного треугольника.

Список источников

1. Д.Н.Юрьев, Тритная система распознавания образов, Математические методы распознавания образов: доклады X Всероссийской конференции, с.160 – 161, 2001.

2. Д.Н.Юрьев, С.С.Постнов, Тритный перцептрон, Математические методы распознавания образов: доклады X Всероссийской конференции, с.161 – 162, 2001.

3. Пучкова Д.М., Постнов С.С., Анализ изображений при помощи динамических NK-сетей из бинарных логических элементов, журн.: Математическое программирование и распознавание образов, с.289 – 292, 2004.

Метод спектральной трикластеризации для систем совместного пользования ресурсами

З. Р. Секинаева [1], Д. И. Игнатов[2]

[1]zari6a@gmail.com, [2]dignatov@hse.ru

[1,2]НИУ ВШЭ, Россия, 101000, г. Москва, ул. Мясницкая, д. 20

Аннотация. Статья посвящена разработке метода трикластеризации на основе графовой спектральной кластеризации. В серии экспериментов на реальных данных исследована эффективность и пригодность метода к анализу данных систем совместного пользования ресурсами, т.н. фолксономий.

Ключевые слова: Анализ сетевых (графовых) данных, системы совместного пользования ресурсами, спектральная трикластеризация, разработка данных (Data Mining).

Введение

За последние два года количество пользователей социальных систем использования тэгов (Social resource tagging system) таких как, Flickr (http://www.flickr.com/) – фотогалерея, del.icio.us (http://delicious.com/) – сервис закладок, Bibsonomy (http://www.bibsonomy.org/) – сервис библиографических закладок, сильно возросло. Популярность таких сервисов связана с тем, что пользователь «ставит закладку» на тот или иной ресурс, которая хранится в виде записи на удаленном сервере, и доступна с любого компьютера. Сервис социальных закладок, обладающий рекомендательным механизмом, становится, более полезным и приобретает статус рекомендательной системы на основе тегов.

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16-18 марта, 2012.

Трехмерная структура данных «пользователь-тэг-ресурс», так называемая фолксономия, лежит в основе таких систем. Фолксономия состоит из трёх множеств U, T, R – пользователей, ресурсов и тегов, а также тернарного отношения Y. Для поиска групп пользователей, использующих одинаковые теги для пометки некоторого множества ресурсов, применяются модели и методы анализа формальных понятий (АФП).

Анализ формальных понятий – прикладная алгебраическая дисциплина, которая развивается в течение последних пятнадцати лет и использует алгебраическую теорию решеток для формализации понятия как единицы человеческого мышления – понятия. Это означает, что понятие должно быть однородным и замкнутым. АФП опирается на так называемый диадический подход, т.е. объектно-признаковое представление данных, что дает возможность формализовать понятие как пару (объем, содержание), такой что объем – объекты, обладающие всеми признаками из содержания, а содержание – признаки, которыми обладают объекты из содержания. Хотя классическая диадическая модель успешно применяется во многих приложениях, во многих ситуациях требуется расширение размера формальных понятий, т.е. добавления третьей компоненты - условия. В качестве формального «условия» можно задавать отношения, методы, значения, причины, цели и т.д. Триадический подход в АФП основан на формализации тернарного отношения, связывающего объекты, признаки и условия [1]. Также как и диадический контекст задается с помощью двумерной таблицей, триадический контекст или триконтекст задается трехмерной.

С ростом популярности и размеров социальных систем использования тэгов, в основе которых лежит фолксономия, возникла острая необходимость в эффективных алгоритмах трикластеризации трехмерных бинарных данных, т.е. поиск плотных троек вида $\langle пользователь, тег, ресурс \rangle$. Существуют эффективные алгоритмы поиска формальных понятий и трипонятий. Но методы поиска трипонятий для реальных данных (несколько миллионов записей $\langle пользователь, тег, ресурс \rangle$), оказываются вычислительно сложными.

Кластеризация – хорошо изученная область в анализе данных. Термин «бикластеризация» впервые предложил Б.Г. Миркин в 1996 г., потому возникновение трикластеризации и n-трикластеризации было лишь делом времени. Существуют разнообразные алгоритмы бикластеризации. Авторы [2] предложили подход по поиску бикластеров, состоящих из двух множеств – объектов и признаков, а также предложили определение бикластера, который обладает полезным свойством: любое формальное понятие исходного контекста содержится в некотором бикластере этого контекста (покомпонентное вложение) при нулевом значе-

нии порога плотности. Для случая триконтекстов авторы [3] ввели определение (плотного трикластера) и предложили алгоритм TRICL для поиска таких плотных трикластеров. Алгоритм TRIAS – один из хорошо известных алгоритмов для разработки данных фолксономий [4], который порождает формальные трипонятия.

Также существует метод для разработки данных многомерных отношений. Его реализация (Data-Peeler) [5] ищет замкнутые множества, удовлетворяющие заданным антимонотонным ограничениям, и превосходит аналогичный алгоритм CubeMiner [6] для разработки замкнутых трехмерных множеств или тримножеств.

Спектральная кластеризация может быть отнесена к дивизимной кластеризации, основанный на спектральных свойствах матрицы данных. Если рассматривать данные в виде графа, то задача кластеризации данных эквивалента задаче разбиения графа. Для двумерных данных граф является двудольным [7], для трехмерных – трехдольным [8,9].

В данной работе предлагается рассмотреть трикластеризацию трехмерных бинарных данных с помощью алгоритма спектральной трикластеризации и провести эксперименты на данных социальной системы использования тэгов Bibsonomy. Сравнить полученные данные с результатами работы алгоритмов TRICL и TRIAS.

Математическая модель

Формальный триадический контекст или *триконтекст* - это четверка (G, M, B, Y), где G – множество объектов, M - множество признаков, B – множество условий и Y – тернарное отношение между этими множествами, т.е. $Y \subseteq G \times M \times B$.

Понятием триадического контекста или *трипонятием* называется тройка (объекты, признаки, условия), которая также обладает свойствами однородности и замкнутости. Однородность достигается в случае, если каждый объект обладает каждым признаком при каждом условии в рамках трипонятия. Замкнутость достигается, когда понятие максимально относительно покомпонентного вложения.

Трикластером назовем тройку вида $T=(g^{\square}, m^{\square}, b^{\square})$, где $(g, m, b) \in Y$, где введены некоторые порождающие *бокс-операторы* (см. [3,8,9]). Плотным трикластером назовем трикластер T, плотность которого превышает минимальное значение – порог, заданный пользователем, т.е.

$$\rho(T)= |g^{\square} \times m^{\square} \times b^{\square} \cap Y| / |g^{\square} \times m^{\square} \times b^{\square}| \geq \rho_{min}.$$

Рассмотрим метод спектральной трикластеризации на примере трехмерных данных, так называемой фолксономии «пользователь-тэг-

ресурс». Тернарное отношение такой структуры можно задать с помощью матрицы смежности трехдольного графа.

Матрица трехдольного графа может быть представлена в виде двумерной матрицы:

$$M = \begin{pmatrix} 0 & A_{UT} & A_{UR} \\ A_{UT}^{T} & 0 & A_{TR} \\ A_{UR}^{T} & A_{TR}^{T} & 0 \end{pmatrix}, \text{ где}$$

A_{UT} - матрица смежности, показывающая какие тэги какой пользователь использует. Аналогично, A_{UR} - матрица смежности пользователь-ресурс, A_{TR} - матрица смежности тэг-ресурс. Таким образом, отношение «пользователь-тэг-ресурс» в графе имеет вид треугольника, соединяющего вершины «пользователь», «тэг», «ресурс».

Пусть n – количество пользователей, m - количество тэгов, k – количество ресурсов, тогда матрица M имеет размер (n+m+k)×(n+m+k).

Разбиение трехдольного графа дает вектор собственных значений матрицы Лапласа – L, соответствующий второму наименьшему собственному значению системы обобщенной задачи нахождения собственных значений матрицы L:

$$Lx = \lambda Wx$$

Как и в случае двудольного графа, система имеет вид,

$$\begin{pmatrix} D_U & -A_{UT} & -A_{UR} \\ -A_{UT}^{T} & D_T & -A_{TR} \\ -A_{UR}^{T} & -A_{TR}^{T} & D_R \end{pmatrix} \begin{pmatrix} u \\ t \\ r \end{pmatrix} = \lambda \begin{pmatrix} D_U & 0 & 0 \\ 0 & D_T & 0 \\ 0 & 0 & D_R \end{pmatrix} \begin{pmatrix} u \\ t \\ r \end{pmatrix}$$

Вычислительную сложность работы алгоритма в худшем случае составляет $O((n + m + k)^3 \log(n + m + k))$.

Преобразовав систему в эквивалентную ей, получим:

$$\sigma = (1 - \lambda)$$
$$A_{nUT} = (x\sigma - A_{nUR}z)y^{T}$$
$$A_{nUR} = (x\sigma - A_{nUT}y)z^{T}$$
$$A_{nTR} = (y\sigma - A_{nUT}^{T}x)z^{T}$$

Исследование этой системы с целью дальнейшей оптимизации алгоритма, а также изучение ее связей с сингулярным разложением матрицы, является предметом дальнейшей работы.

Рассмотрим пример: ученые (Jäschke, Stumme, Poelmans, Ignatov, Dedene) помечают тэгами (‘Machine Learning’, ‘Ontology’, ‘Domestic Violence’, ‘Formal Concept’, ‘Triclustering’) статьи (paper1, paper2, paper3).

Трехдольный граф отношения представлен на рисунке 1:

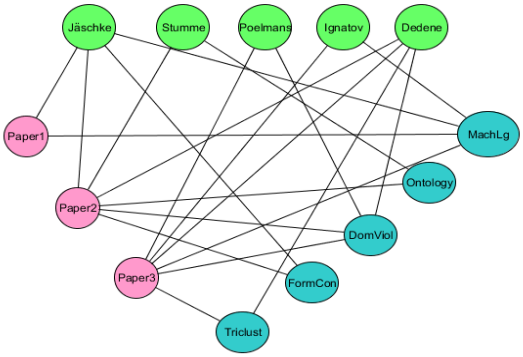


Рис. 1. Пример фолксономии (статьи, читатели, теги)

Рекурсивный вызов функции кластеризации или разбиения графа дает следующую иерархию трикластеров. Критерий остановки – трикластер, содержащий либо одного пользователя, либо один тэг, либо один ресурс.

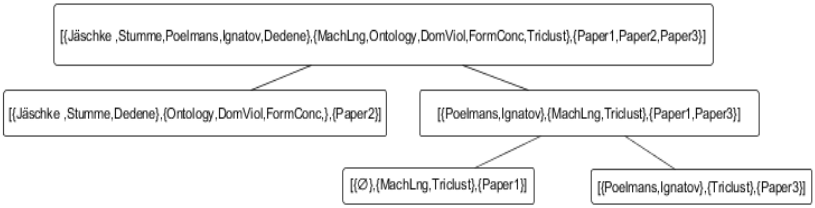


Рис. 2. Пример дерева работы алгоритма SpecTric

Итого имеем 3 кластера (исходная матрица не рассматривается в качестве трикластера). Для интерпретации данных интересны лишь те трикластеры, которые содержат более одного пользователя, тэга и ресурса.

Эксперименты на данных системы социальных закладок: Bibsonomy.org

Для экспериментов были выбраны реальные данные сервиса Bibsonomy.org, содержащие 816197 записей вида (пользователь, тег, ресурс). Фактически мы имеем триконтекст, которому соответствует параллелепипед размерами $|U||T||R|=2337\times67464\times28920$, с принадлежащими ему $|Y| = 816\,197$ тройками. В качестве экспериментального

оборудования использовался компьютер с процессором Intel(R) Core(TM) Duo CPU с частотой 2,1ГГц и с 4ГБ оперативной памяти. В качестве языка программирования выбран высокоуровневый язык Matlab версии 7.8.0(R2009a). Предварительно данные были обработаны: все тэги были приведены к нижнему регистру.

Эксперименты проводились на 100 тысячах исходных записей, которые соответствуют параллелепипеду размером $59 \times 5197 \times 28920$.

На графике представлена структура матрицы смежности графа для первых ста тысяч троек данных сервиса Bibsonomy.org

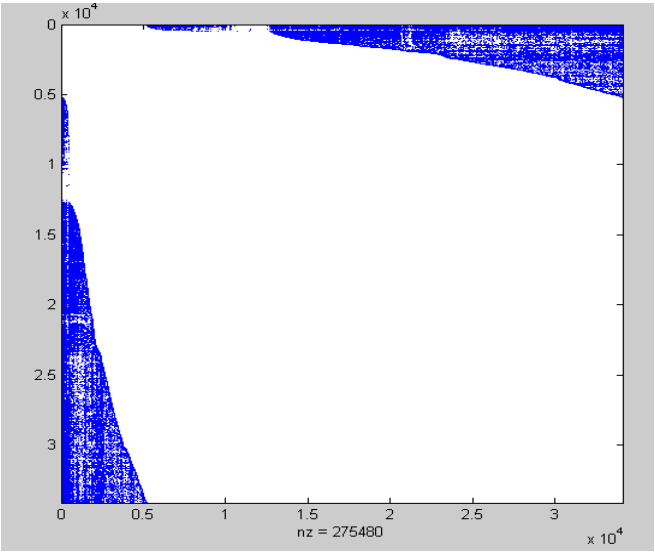


Рис. 3. Вид исходной матрицы M с помощью команды spy в системе Matlab

Плотность матрицы: $\rho = \frac{|Y|}{|U||T||R|} = 0{,}000113.$

Перед применением алгоритма интересно посмотреть статистическое распределении. На рисунке 1 показаны гистограммы распределения тэгов по количеству внесенных в систему пар («пользователь», «документ») для первых ста тысяч троек. Данные следуют степенному закону распределения $p(x) = Cx^{-\alpha}, \alpha = 3{,}6778$ и $\sigma = 0{,}0001$ в случае документов и количества пар (пользователь, тэг). Для пользователей и тегов $\alpha = 2{,}13$ и $\alpha = 1{,}8$ соответственно. Это наблюдение позволяет применять жадную стратегию поиска, то есть находить большие и плотные трикластеры, так как небольшая часть пользователей совершает присваивание (тэг, документ). Аналогично, для тэгов и документов.

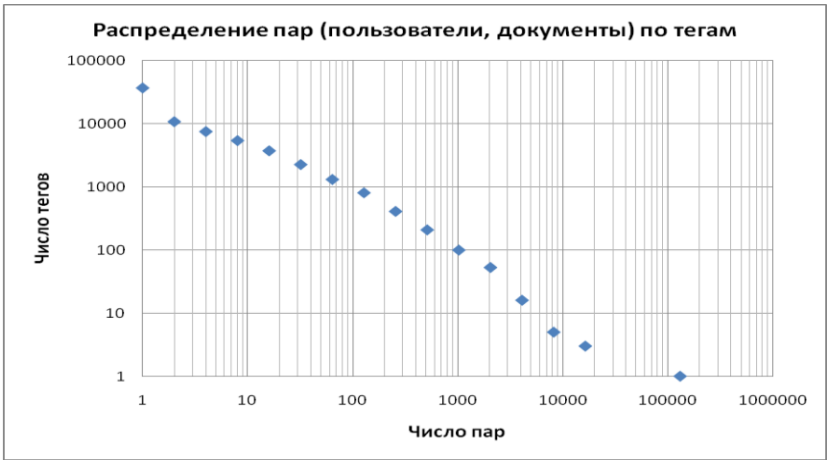


Рис. 4. Распределение числа пар (пользователи, документы) по тегам

Для первых ста тысяч записей алгоритм спектральной трикластеризации породил 131 трикластер приблизительно за 9 минут 14 секунды (без учета подсчета плотностей трикластеров) в то время как алгоритм TRICL породил 4462 трикластера примерно за 21 минуту (без учета подсчета плотностей трикластеров).

Время работы алгоритмов увеличивается при подсчете плотностей, т.к. при этом осуществляется полная проверка на принадлежность исходному отношению I тройки (g, m, b) – элемента конкретного трикластера T.

В таблице представлены результаты экспериментов на первых ста, десяти тысяч, ста и двухстах тысяч троек данных сервиса Bibsonomy с подсчетом плотности трикластеров (ρ_{min}=0):

Табл. 1. Результаты работы работы алгоритмов SPECTRIC и TRICL

Кол-во записей	|U|	|T|	|R|	$\|T_c\|$	$\|T_s\|$	TRICL, s	SPECTRIC, s
100	1	47	52	1	1	0.2	0.2
10 000	1	395	5193	1	1	2	2
100 000	59	5138	28920	4462	131	10311	6215

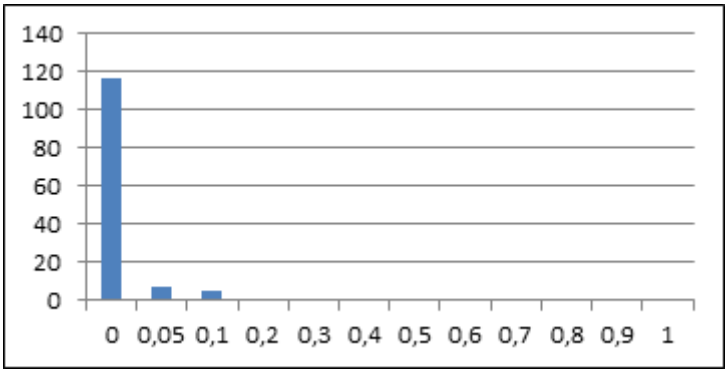


Рис. 5. Гистограмма плотностей трикластеров для первых ста тысяч записей:

Также дополнительные эксперименты показали, что метод является устойчивым при 15% уровне шума (имеется в виду инвертирование тройки (g, m, b) в соответствии с равномерным распределением), но перестает корректно выявлять подгруппы при уровне шума 20% и более.

Основная доля трикластеров, порожденных в ходе эксперимента, имеет довольно невысокие значения плотности.

Заключение

Нахождение трикластеров позволяет видеть все ресурсы (их идентификаторы) и тэги, которыми пользовались клиенты сервиса Bibsonomy.

Результаты, полученные в ходе экспериментов позволяют сделать следующие выводы: на реальных данных алгоритм спектральной трикластеризации работает в два раза быстрее существующих алгоритмов, количество порождаемых трикластеров в среднем в 30 раз меньше по сравнению с количеством трикластеров, представляющими собой формальные понятия (алгоритм TRIAS), и количеством трикластеров порождаемых алгоритмом TRICL, предложенным в работах [3,8].

Однако данные трикластеры не такие плотные по сравнению с трикластерами, порождаемыми алгоритмом TRICL, и трикластерами, представляющими собой формальные понятия, плотность которых всегда равна единице.

Эксперименты на синтетических данных позволяют говорить, что при 15% уровне «шума», то есть разного рода интересах участников сообщества, алгоритм корректно выделяет сообщества (алгоритм перестает корректно выявлять подгруппы при уровне шума 20%).

Дальнейшие исследования по данной тематике возможны в двух направлениях: с одной стороны это улучшение эффективности алгоритма трикластеризации с использованием аппарата линейной алгебры, а именно, исследования связи полученной в работе системы с сингулярными значениями матрицы, с другой стороны, введение дополнительных критериев остановки ветвления алгоритма и разработки новых мер качества.

Список источников

1 Lehmann F., Wille R.: A triadic approach to formal concept analysis. In: Ellis, G., Rich, W., Levinson, R., Sowa, J.F. (eds.) ICCS 1995. LNCS, vol. 954, pp. 32–43 Springer, Heidelberg, 1995.

2 Игнатов Д.И., Каминская А.Ю, Кузнецов С.О., Магизов Р. А. Метод бикластеризации на основе объектных и признаковых замыканий// Интеллектуализация обработки информации: 8-я международная конференция. Сборник докладов. – М.: МАКС Пресс, 2010. – С. 140 – 143.

3 Игнатов Д.И., Магизов Р.А. Анализ тримодальных данных на примере Интернет-сервисов социальных закладок// Социологические методы в современной исследовательской практике: Сборник статей. – М.: НИУ ВШЭ, 2011. – С. 315 – 321.

4 Robert Jäschke, Andreas Hotho, Christoph Schmitz, Bernhard Ganter, Gerd Stumme: TRIAS - An Algorithm for Mining Iceberg Tri-Lattices. ICDM 2006:907-911.

5 Loïc Cerf, Jérémy Besson, Céline Robardet, Jean-François Boulicaut: Data Peeler: Contraint-Based Closed Pattern Mining in n-ary Relations. SDM 2008:37-48.

6 Liping Ji, Kian-Lee Tan, Anthony K. H. Tung: Mining Frequent Closed Cubes in 3D Datasets. VLDB 2006:811-822

7 Zhukov L. Technical report Spectral Clustering of Large Advertiser Datasets Part 1. 2004.

8 Dmitry I. Ignatov, Sergei O. Kuznetsov, Ruslan A. Magizov, Leonid E. Zhukov: From Triconcepts to Triclusters. RSFDGrC 2011, LNCS/LNAI Volume 6743, Springer, 2011:257-264.

9 Игнатов Д. И., Кузнецов С. О., Пульманс Й. Разработка данных систем совместного пользования ресурсами: от трипонятий к трикластерам //Математические методы распознавания образов: 15-я Всероссийская конференция. Сборник докладов. – М.: МАКС Пресс, 2011. – С. 258 – 261.

Автоматизированная система распознания рукописных исторических документов

А. В. Скабин[1], И. А. Штеркель[2]

[1]artb00g@gmail.com, [2]shterkel_ivan@psu.karelia.ru

ПетрГУ, Петрозаводск, Россия

Аннотация. В статье описываются результаты исследования по созданию универсальной программной системы для автоматизированного распознавания исторических рукописных текстов, включая исторические стенограммы XIX и начала XX веков. Рассматривается проблема получения оригинальной графики символов исторических рукописных документов путем бинаризации пороговым методом, а так же поиск схожих график в базе знаний. Кроме этого описывается прототип автоматизированной системы распознания рукописных исторических документов.

Ключевые слова: автоматизация; распознание; исторические; рукописные; документы; бинаризация; пороговый метод; прототип; стенограммы.

Введение

В настоящее время в архивах России имеется большой объем нерасшифрованных стенографических документов. Причина – невозможность дешифровки исторических документов современными стенографистами. В течение XIX и начала XX веков стенография в России находилась в процессе становления, поэтому существующие документы зашифрованы в разных системах, к тому же современная стенография существенно отличается от исторических систем стенографии XIX века.

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16-18 марта, 2012.

Основные сложности дешифровки стенограмм заключаются в следующем:

- отсутствие людей обладающих знаниями о системах стенографической записи в XIX – начале XX вв. Существуют только старые учебники;
- стенографист при шифровании мог использовать свои нестандартные символы (обозначения), так как зачастую расшифровкой занимался он сам;
- в стенографической записи распространены: метод пропуска гласных букв или замена часто встречающихся сочетаний символов, слов одним символом;
- некоторые символы стенографической записи могут иметь схожее написание, но в зависимости от некоторых физических параметров, например таких как высота, могут иметь различное значение.

Цель данной работы - создание универсальной программной системы для автоматизированного распознавания исторических рукописных текстов, включая исторические стенограммы XIX и начала XX веков. Она призвана решить задачу описания и дешифровки исторических стенограмм, а также ввести в научных оборот новые документы. Данное исследование поддержано грантом РГНФ № 11-01-12026в (рук. Рогов А.А.).

Описание разрабатываемой системы

Отличительные свойства разрабатываемой системы: учет особенностей исторической орфографии XIX и начала XX веков, учет индивидуальных знаков разных стенографистов, возможность критического анализа, использование словаря для подсказки при дешифровке текста и т.д. [1]. Информационная система будет находиться в открытом доступе и предлагаться к использованию работниками архивов, научными сотрудниками, исследователями текстологам. Отлаживать систему было принято решение на стенограммах А.Г. Сниткиной, частично расшифрованных Ц. М. Пошемянской , и учебнике П. Ольхина [2].

Распознавание любого текста включает в себя следующие этапы:

- предобработка изображения – как правило, это бинаризация изображения;
- сегментация – выделение на предобработанном изображении текстовых областей – символов, сочетание символов, слова, строки;
- анализ полученных сегментов – установление значений, признаков, сравнение с эталонами находящимися в базе знаний;

- расшифровка – выбор наиболее подходящих словоформ из словаря соответствия с определенной моделью языка.

Дополнительные сложности при распознавании текста создают искривления строк, перепады яркости, просвечивания текста с обратной стороны и другие дефекты оригинала и изображения. Распознавание рукописного текста создает дополнительные сложности в отличие от распознавания печатного текста [3].

В рамках исследования ставится задача создания достаточно универсальной программной системы для автоматизированного распознавания исторических стенограмм, для которых автоматическое распознавание оказывается пока невозможным. Предлагаемая система автоматизированной дешифровки исторических стенограмм с возможностью интеллектуальной поддержки принятия решений при наборе позволит существенно ускорить процесс перевода рукописного текста в текстовый файл и повысит точность его дешифровки.

Разрабатываемая программная система будет обладать следующими характеристиками [1]:

- автоматизация набора предполагает использование виртуальной клавиатуры оригинальных символов различного размера;
- виртуальная клавиатура представляет собой таблицу соответствия оригинальных графем буквосочетаниям, множество графем формируется методом сегментации на основе анализа текста;
- система автоматически контролирует состояние набора и в интерактивном режиме выдает информацию пользователю;
- система возвращает пользователю варианты набора словоформ, упорядоченные по частоте встречаемости в базе данных, и информацию об отсутствии набранного слова в базе знаний.

Бинаризация исторических рукописных документов

При распознавании исторических рукописных документов возникает проблема с бинаризацией изображения. Из-за состаренности изображения и того что стенографические записи сделаны простым карандашом на пожелтевшей бумаге. Пороговый метод по цветовым компонентам (RBG), оказался не приемлемый для данной задачи, так как пиксели фона и символов имеют схожие значения цветовых компонент. Как видно на гистограммах (рис. 1) отсутствие двух явно выраженных пиков не позволяет выбрать пороговое значение для бинаризации. Такие же результаты получаются (рис. 1), если использовать разложение по цветовой схеме HSB (оттенок, насыщенность, яркость). Производя бинаризацию только по пороговому значению яркости, можно получить четкие символы, с малым количеством шума.

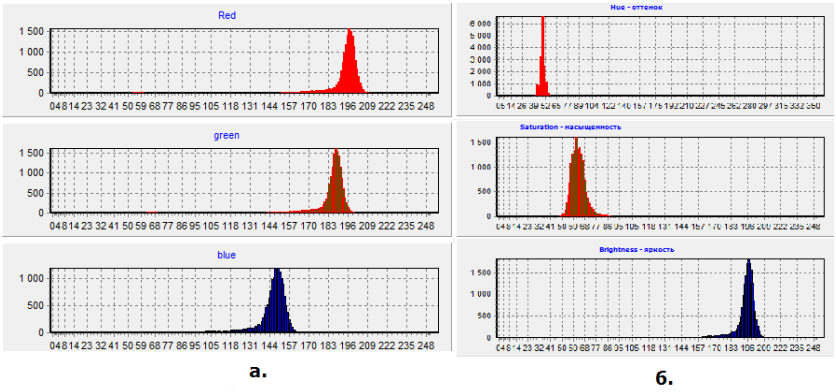


Рис. 1. Гистограммы цветовых схем RGB (а) и HBS (б)

Экспериментально было найдено порогового значения яркости, при которой символы получаются наиболее четкими, с наименьшим количеством шума. Наилучший результат достигается, если процент черных пикселей после бинаризации приближается к 13% относительно общего числа пикселей.

Модуль создания оригинальной графики символов

Система была разбита на несколько модулей. Один из них это модуль создания оригинальной графики символов. На рис. 2 представлен интерфейс модуля создании оригинально графики символов.

Основное окно программы представляет собой две области, в левой области находится оригинальное изображение (оригинальная стенограмма), на которой пользователь выделяет необходимый символ, как видно на рис. 2. Место выделения отображается на второй области. В данной области находится обработанная стенограмма, т.е. все полученные ранее символы, находящиеся в местах, соответствующих символам в оригинальном изображении.

После выделения символа пользователь должен нажать на «горячую клавишу» или их сочетание. Далее система производит бинаризацию выделенного фрагмента и его сегментацию. Если сегментов получено несколько, то система предлагает пользователю выбрать какой сегмент или сегменты соответствуют оригинальному символу. Если было выбрано несколько сегментов, то система производит связывание[4] разорванных «кусков» и предлагает пользователю результат. В случае, когда результат устраивает пользователя, символ записывается в базу знаний и располагается в правой области соответственно месту (координатам) на оригинальном изображении. Если результат не соответствует требо-

ванию пользователя, то возможно редактирование полученного символа при помощи упрощенным графическим редактором.

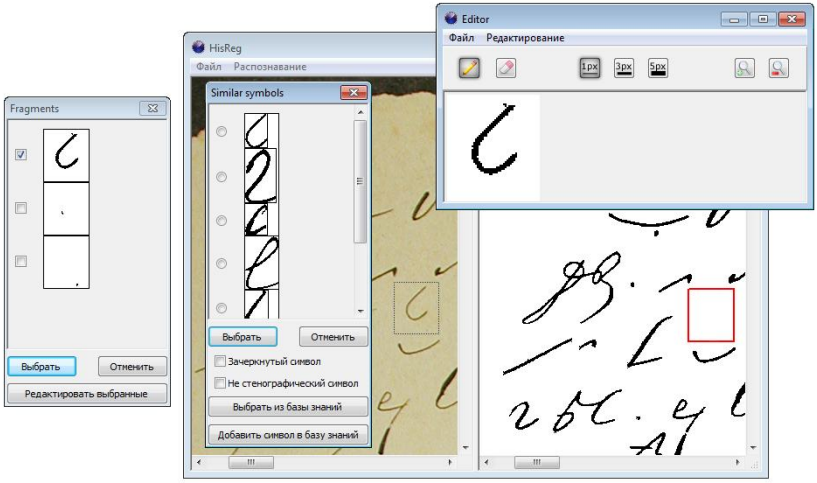


Рис. 2. Интерфейс модуля создания оригинальной графики символов.

Создание оригинальной графики затруднено следующим:

- оригинальное изображение уже довольно старое, и было написано простым карандашом на пожелтевшей бумаге, которая имеет перегибы, различные повреждения, надрывы, так же на некоторых стенограммах присутствуют сторонние записи, не имеющие смысловую нагрузку, либо прочерчены строки, пересекающиеся с символами;
- при бинаризации происходили разрывы символов, т.к. некоторые пиксели символа имели схожий цвет с пикселями бумаги;
- при сегментации возникала необходимость разбиения символов, написанных слитно, на отдельные символы.

Поиск символа в базе знаний

При создании оригинальной графики символов возникла задача формирования базы знаний символов. Данная база знаний необходима для исключения дублирования и избыточности графики символов. База знаний является пополняемой, т.е. если в процессе распознания символ ранее не встречался, то он добавляется в базу. За основу базы данных была взята выборка из 132 символов, случайно выбранных на стенограмме. Были использованы следующие методы сравнения текущего

символа с символами из базы данных: Сравнение с эталоном, сравнение со скелетом эталона, метод краевых расстояний. Сравнение данных методов на обучающей выборке из 132 символов представлено в табл. 1.

Табл. 1. Сравнение методов на обучающей выборке

Характеристика / Метод	Время сравнения	Точность
Сравнение с эталоном	3 сек. (в зависимости от размера символа)	Менее 30%
Сравнение со скелетом эталона	1-2 сек. (в зависимости от размера символа)	~40%
Метод краевых расстояний	менее 0.01 сек.	Более 60%

Низкая точность сравнения с эталоном связана с тем, что при бинаризации символ мог иметь разную толщину в зависимости от размера выделенного фрагмента стенограммы. При сравнении скелетов, скелетолизация символов производилось алгоритмом Зонга Суня [5]. Метод краевых расстояний заключается в следующем, из базы знаний выбираются символы отношения высоты к ширине которых находится в некоторой окрестности. Далее у текущего символа измеряются расстояния $\{l_1,l_2,\ldots,l_8\}$ (см. рис. 3) .

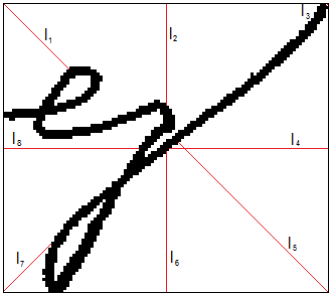


Рис. 3. Метод краевых расстояний.

Из ранее выбранных символов из базы знаний, выбираются такие у которых данные расстояния $\{\grave{l_1},\grave{l_2},\ldots,\grave{l_8}\}$ находятся в промежутке $(l_1 \cdot k - \varepsilon, l \cdot k + \varepsilon)$, где k – отношение высоты к ширине текущего символа, $\varepsilon = k \cdot l \cdot \alpha$, где $\alpha = 0.1$.

В процессе обработки 29 листов стенографических записей, было выделено более 2500 оригинальных граф символов. После этого

встает задача нахождения соответствия (значений) данных график исходя из учебника по которому училась Снеткина, и частично расшифрованных записей. Интерфейс прототипа данной системы представлен ниже.

Прототип системы автоматизированной системы распознания рукописных исторических документов

На рис. 4 представлен интерфейс прототипа автоматизированной системы распознавания рукописных исторических документов. Как видно на рисунке система имеет 4-е области. Область с оригинальным изображением, область возможных значений дешифровки символов или группы символов.

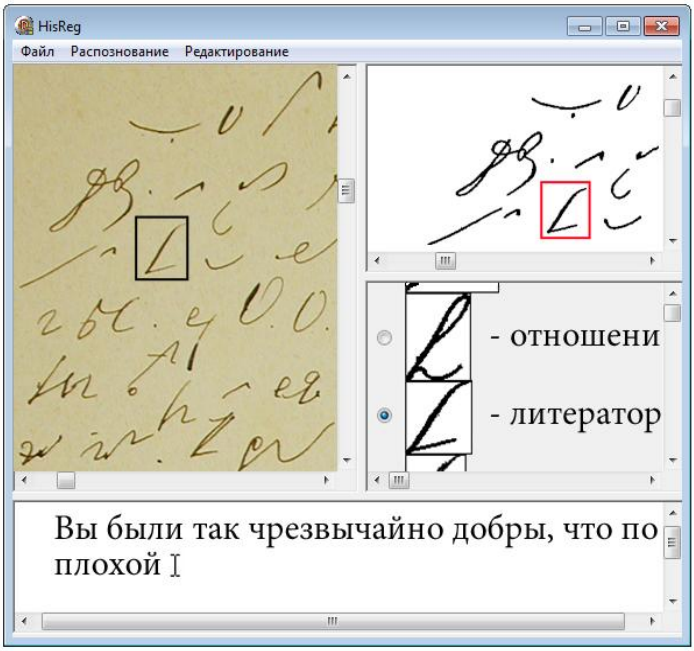


Рис. 4. Интерфейс прототипа автоматизированной системы распознания рукописных исторических документов

При выделении символа на оригинальном изображении, изображение символа располагается в области 2, в том же месте где он и находится на оригинальной стенограмме. В 4-ой области отображается дешифрованные символы. Так же система в процессе набора слова анали-

зирует его составные части и предлагает пользователю близкие по значению расшифровки из базы знаний. Система, анализируя исходное изображение при вводе символов, производит автоматическое дешифрование схожих символов или групп символов.

Основные преимущества данной системы заключается в следующем:

- использования «горячих клавиш» - ускоряет набор исторической стенограммы;
- связь графического изображения стенограммы и его текстового представления;
- интеллектуализированный набор;
- возможность автоматического распознавания в тексте схожих сочетаний символов, слов;
- возможность совместной работы нескольких пользователей с одним словарем.

Данная система призвана ускорить процесс дешифровки рукописных исторических стенограмм. В дальнейшем возможна реализация в виде Web-сервиса, для организации распределенной, удаленной работы со стенограммами.

Выводы

В модуле создания оригинальной графики символов использован пороговый метод бинаризации с подобранными параметрами, при которых бинаризация происходит наилучшим способом. Данные параметры индивидуальны для каждого типа документов, поэтому возникает необходимость использовать иной более универсальный метод бинаризации, описанные в [6].

Текущие методы поиска изображений в базе знаний не дают высокой точности, в связи с этим может возникать избыточность оригинальных граф символов. Вследствие чего, необходимо использовать метод, дающий более высокую точность. Для этого будут проанализированы методы, верификации личности по подписи [7].

Список источников

1 Рогов А.А., Талбонен А.Н., Варфоломеев А.Г. Автоматизированная система распознавания рукописных исторических документов // Электронные библиотеки: перспективные методы и технологии, электронные коллекции: Труды XII Всероссийской научной конференции RCDL’2010 (Казань, Россия, 13–17 октября 2010 г.). – Казань: Казан. ун-т РАН, 2010. - С. 469-475.

2 Ольхин П. Руководство к русской стенографии. - СПб.: Типография доктора М. Хана, 1866 г

3 Горский Н., Анисимов В., Горская Л. Распознавание рукописного текста: от теории к практике. – СПб.: Политехника, 1997 г.

4 P Nagabhushan, Basavaraj S Anami. A knowledge-based approach for recognition of handwriting Pitman shorthand language storkes. / P Nagabhushan, Basavaraj S Anami // *Sadhana. – 2002.* - Vol. 27, Part 5. -P. 685–698.

5 Zhang, T.Y. A fast parallel algorithm for thinning digital patterns / T. Y. Zhang, C. Y. Suen // Commun. ACM. − 1984. − Vol. 27, №3. – P. 236-239.

6 Ioannis Pratikakis, Basilios Gatos, and Konstantinos Ntirogiannis. Icdar 2011 document image binarization contest (DIBCO 2011). In ICDAR, pages 1506–1510, 2011

7 Кухарев Г.А. Биометрические системы: методы и средства идентификации личности человека. – Спб.: Политехника, 2001. – 240с.

Консенсус в социальных сетях: динамический подход

Ф. В. Строк

fdr.strok@gmail.com

Национальный Исследовательский Университет Высшая Школа Экономики, Москва, Россия

Аннотация. Статья посвящена анализу процесса передачи информации в сетях в случае, когда структура сети меняется со временем. Сформулирована математическая модель, чтобы описать несколько возможных закономерностей развития. Для сравнения со стандартными моделями была написана программа в среде Matlab.

Ключевые слова: марковская цепь, социальная сеть, консенсус, граф доверия.

Введение

Методы анализа социальных сетей стремительно развиваются и находят широкое применение. Основным инструментом является теория графов. Одним из основных предметов исследования – выделение ключевых вершин в сети. Существует множество подходов к решению данной задачи оценивания важности. Другой важной задачей является исследование возможности консенсуса внутри сети. Главное отличие – эта задача явно включает время. Основополагающая модель была предложена ДеГрутом [1]. Основным понятием является граф доверия. Веса на ребрах обозначают интенсивность взаимодействия. Граф неориентированный и необязательно связанный.

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16-18 марта, 2012.

Базовая модель.

Есть сообщество, в котором у каждого члена есть точка зрения по некоторому вопросу. Граф описывается квадратной матрицей:

$$T \in R_{n*n}, T_{ij} > 0, \sum_{j=1}^{n} T_{ij} = 1 \ (1)$$

Граф – взвешенный, ориентированный, с неотрицательными весами на ребрах.

Этот граф отвечает отношению доверия, то есть ребро от i к j есть, если i-ый доверяет j-ому, вес ребра – сила связи. Веса нормированы так, что сумма весов исходящих ребер равнялась 1.

Элемент T_{ij} описывает, как человек i оценивает мнение человека j, при формировании своего мнения на следующем шаге.

Начальное состояние описывается вектором:

$$p(0) = \big(p_1(0), \dots, p_n(0)\big), p_i(0) \in [0,1] \forall i = 1, \dots, n \ (2)$$

Процесс формирования «влияния» участников описывается следующим образом:

$$p(t) = T * p(t-1) = T^t p(0) \ (3)$$

Модель допускает введение средств массовой информации или идейных лидеров. Они не изменяют свое мнение, но влияют на других:

$$\forall\, i: T_{ij} = \delta_i^j \text{ и } \exists k: T_{ki} > 0 \ (4)$$

Такие вершины отвечают внешним источникам информации. Симметрично можно ввести, «чистых слушателей», которые не влияют ни на каких других агентов сети.

Консенсусом называется итоговое распределение влияний участников.

Сходимость.

Существует два основных вопроса:

Каковы условия сходимости.

Каково решение в результате сходимости.

Необходимое условие сходимости: $\forall p: \exists \lim_t T^t p$.

Здесь применимы результаты из теории марковских цепей: процесс сойдется, если матрица T апериодична и ориентированный граф связен.

Это предположение слабее обычного. Процесс сойдется, если матрица T описывает сильно связных ориентированный граф и для какой-то вершины есть петля (это гарантирует, что НОД длин циклов будет 1).

Расширение модели.

Предлагается внести изменения в правила описывающие процесс изменения мнений – сделать его из статического динамическим. Это будет отвечать ситуации, когда люди изменяют мнение не только о проблеме, но и о людях вокруг.

Добавление ребер.

Первая идея – изменять матрицу связи таким образом, что добавляются транзитивные замыкания. Если человек 1 доверяет человеку 2, а человек 2 в свою очередь доверяет человеку 3, через какое-то время вполне вероятно, что первый станет доверять третьему напрямую. Это описывает ситуацию, когда друзья знакомятся между собой.

Таким образом, граф доверия становится плотней, это приводит к тому, что сходимость происходит быстрее. Тогда нельзя использовать стандартные теоремы. Задача будет найти собственный вектор последовательности матриц. Придется использовать итерационные методы для решения этой задачи.

Глубиной изменений будем называть максимальное число ребер, которое добавилось во время перестроения сети.

Удаление ребер.

Второй подход к перестроению сети – удалять ребра из графа, отражает разногласия при формировании общего мнения.

Комбинированный подход.

Обобщение двух приведенных выше методов, когда используется и удаление, и добавление; путем установления крайних значений можно свести к предыдущим моделям.

Компьютерные эксперименты.

Рассмотрим сеть в виде цепи:

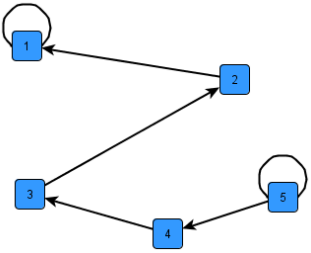


Рисунок 1. Граф-цепь.

Предложенная модель с p^{adding}=0.3 позволяет увеличить скорость сходимости – с 20 итераций до 12.

Наблюдаются изменения структуры сети (рис. 2).

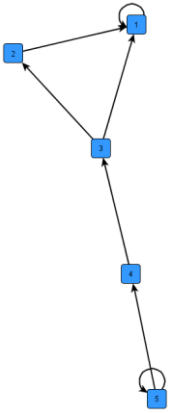


Рисунок 2. Измененная структура сети.

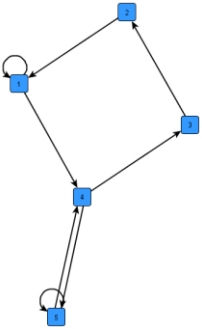


Рисунок 3. Случайные изменения сети-цепи.

Предложенная модель действительно увеличивает скорость сходимости. Однако результаты, полученные с помощью данной модели, не изменяют выход. Мы изменяем веса путей, но не добавляем ребер, нарушающих сходимость.

Свойство графа сохранились. Это интересно, если мы исследуем источники информации. Это отвечает процессу расширения аудитории средств массовой информации, например, когда друзья рекомендуют заинтересовавший их журнал.

Случайно добавлять ребра нельзя, так как мы можем изменить принципиальную структуру графа (рис. 3). Естественно характер выхода изменится значительно:

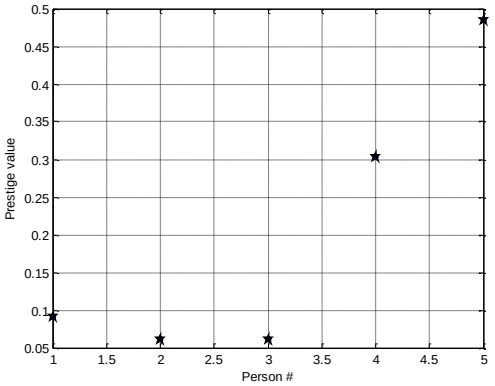


Рисунок 4. Консенсус при случайном добавлении ребер.

Удаление ребер.

Идея – ослаблять веса тех ребер, которые минимальны. Это опишет ситуацию, когда человек предпочитает стабильные связи временным. Снова важно сохранить структуру сети – поэтому ребра ослабляются, а не удаляются совсем. Таким образом, консенсус достижим, хотя на это может уйти большее число итераций.

Случайное добавление

Случайные изменения описывают ситуацию, когда некоторые связи определены неверно. Так же это отвечает модели стохастических изменений в структуре сети.

Мы можем использовать бутстреп подход для определения устойчивости консенсуса. Основной параметр алгоритма – число допустимых изменений изначальной структуры. Повторяем наш случайный процесс много раз и усредняем получившееся распределение и уже среднее сравниваем с распределением на изначальной структуре.

Рассмотрим цепь.

Проведем анализ устойчивости для разных глубин. Число повторений - 1000.

Увеличение глубины ведет к сходимости средних значений, однако одновременно растет и дисперсия (рис. 6).

Для глубины=20 увеличение дисперсии еще больше. Средние значения продолжают сближаться.

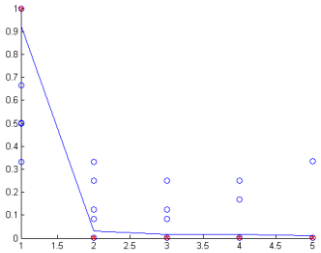


Рисунок 5. Цепь,
глубина изменений 1.

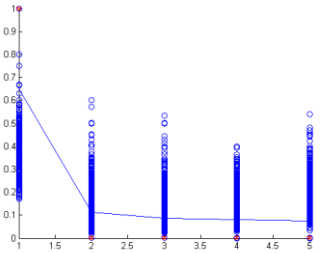


Рисунок 6. Цепь,
глубина изменений 5.

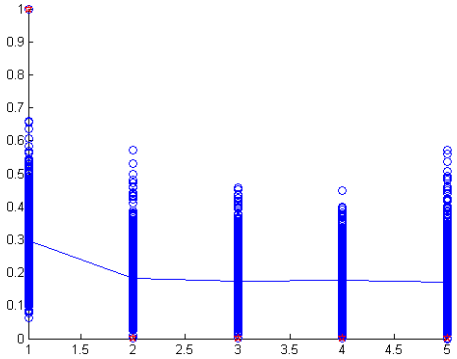


Рисунок 7. Цепь, глубина изменений 20.

Другая интересная структура сети – «звезда». Отвечает ситуации, когда человек только прислушивается. То есть у вершины, отвечающей ему, есть только исходящие ребра.

Для глубины =1 для «звезды» возможно всего несколько вариантов развития (рис. 8 – 10).

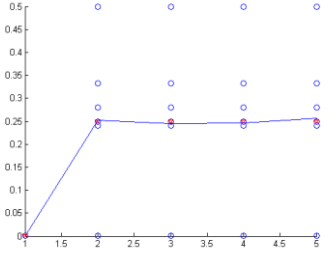


Рисунок 8. Звезда,
глубина изменений 1.

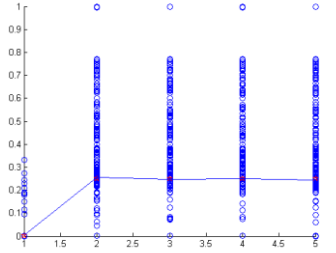


Рисунок 9. Звезда,
глубина изменений 5.

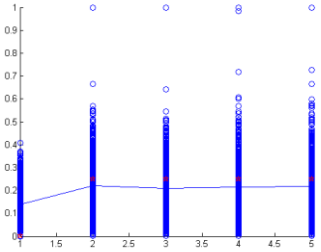


Рисунок 10. Звезда, глубина изменений 20.

Третья структура – «анти-звезда». Так можно моделировать СМИ. Есть вершина, у которой все ребра входящие.

Результаты симметричны результатам для звезды, что следует из симметричности структур.

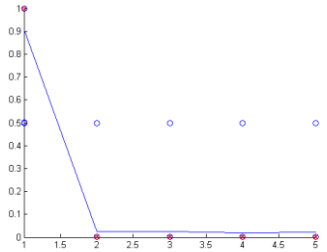


Рисунок 11. «Анти-звезда», глубина изменений 1.

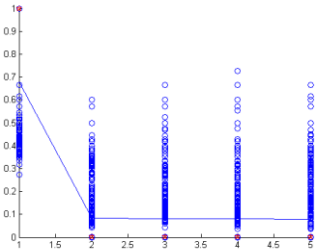


Рисунок 12. «Анти-звезда, глубина изменений 5.

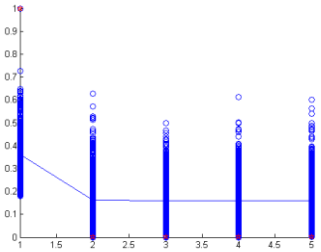


Рисунок 13. «Анти-звезда», глубина изменений 20.

Глубина 5 увеличивает дисперсию значений, а среднее значение сходится (рис. 12).

Глубина 20 еще сильнее сглаживает распределение в смысле среднего (рис. 13).

Наблюдаем зависимость размаха среднего престижа от глубины. Изучим это зависимость более точно. На нижеприведенных рисунках: по оси абсцисс – глубина, по оси ординат – размах среднего престижа.

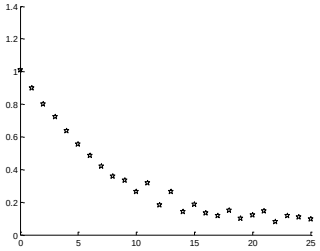


Рисунок 14. Цепь: зависимость среднего престижа от глубины изменений

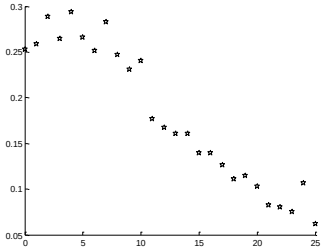


Рисунок 15. Звезда: зависимость среднего престижа от глубины изменений

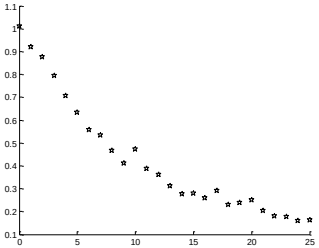


Рисунок 16. «Анти-звезда»: зависимость среднего престижа от глубины изменений.

Наблюдаем, что самая неустойчивая структура – у звезды, так как трансформации могут увеличить размах среднего значения. Но у всех структур одинаковое асимптотическое поведение размаха среднего престижа по отношению к глубине изменений.

Заключение

Протестированы два подхода к изменению модели ДеГрута. Оба показали, что изменение процесса передачи информации замедляет

сходимость, но итоговое распределение становится ближе к равномерному.

При этом, все расширения можно рассматривать как обобщения модель ДеГрута, полагая соответствующие вероятности равными нулю, можно получить решения в терминах классической модели.

Список источников

1. M.H. DeGroot: Reaching a Consensus, Journal of the American Statistical Association, Vol. 69, No. 345, 1974, pp. 118-121.

2. Roger L. Berger: A Necessary and Sufficient Condition for Reaching a Consensus Using DeGroot's Method, Journal of the American Statistical Association, Vol. 76, No. 374, 1981, pp. 415-418.

3. Matthew O. Jackson: Social and Economic Networks, Princeton University Press, 2008, 520 p.

4. Ulrik Brandes, Jurgen Lerner, Tom A. B. Snijders: Networks Evolving Step by Step: Statistical Analysis of Dyadic Event Data, Advances in Social Network Analysis and Mining. In proc. International Conference on Advances in Social Network Analysis and Mining (ASONAM 2009), IEEE Computer Society, 2009, pp. 200-205.

Особенности создания поискового индекса к фотографиям в цифровом историческом альбоме

Талбонен А. Н.

perhetal@onego.ru

Петрозаводский Государственный Университет

Аннотация. В данной статье описывается метод построению поисковой системы для узкоспециализированной текстовой коллекции, допускающей наличие текстовых ошибок, на примере коллекции фотографий со строительства Беломорско-Балтийского канала. Предложенные методы можно использовать для других проектов, связанных с работой над коллекциями исторических документов.

Ключевые слова: фильтрация изображений, распознавание текста, семантический анализ, морфологический анализ, поиск по шаблону, поисковый индекс.

Введение

В настоящее время существует большое количество оцифрованных коллекций исторических документов, поэтому задача организации поиска по ним является весьма актуальной. С учетом современного развития вычислительной техники и программных систем оцифровке подвергаются все более сложные архивные коллекции. В случае, когда оригинальные документы содержат текст любого стандартного шрифта, можно оцифровать коллекцию не только в формат изображения, но также в текстовый формат за счет сканирующих средств с высоким разрешени-

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16-18 марта, 2012.

ем. Однако в случае, если коллекция уже была однажды оцифрована с низким качеством, а выполнить повторную оцифровку высококачественным оборудованием не представляется возможным, необходимо использовать другие методы.

Другим способом организации поиска по документам является непосредственный анализ цифровых изображений с последующим извлечением текстовых меток. Найденные на изображениях объекты и соответствующие им метки в дальнейшем можно использовать при формировании полнотекстового индекса.

Практическая цель настоящего исследования заключается в формирования полнотекстового индекса для оцифрованной коллекции низкого качества за счет распознавания текста в документах коллекции и поиска объектов на изображениях. В данной статье описываются текущие результаты работы, в частности описывается метод формирования индекса на основе извлеченной текстовой информации, а также направления и промежуточные результаты текущего исследования в области поиска объектов на изображениях.

Исходные данные

В качестве исходного материала при поведении исследования использовалась коллекция снимков строительства Беломорско-Балтийского канала, сделанных в 30-е годы прошлого века. Данная коллекция состоит из 8-ми альбомов в среднем по 800 снимков в каждой, что в общей сложности составляет почти 6,5 тыс. изображений и находится в Карельском государственном краеведческом музее. Типичный пример изображения можно увидеть на рис. 1. Каждое изображение данной коллекции представляет собой сфотографированный лист, на который была наклеена оригинальная фотография, а также подпись к фотографии в виде небольшой бумажной полоски, с машинописным текстом [1].

Рассматриваемые изображения обладают следующими свойствами:

1. Оригинальные снимки и подписи к ним были сделаны в 1930-х гг.
2. Цифровые изображения были получены методом цифрового фотоаппарата с разрешением 75 точек на дюйм. Полученные снимки хранятся в формате JPEG.
3. Все изображения – черно-белые.
4. Общая зашумленность изображений, в частности областей, содержащих подпись

Рис. 1. Пример изображения коллекции, посвященной строительству ББК.

Постановка задачи

По данной коллекции требуется построить поисковую систему. В рамках исследования основная цель работы предполагает решение 2-х задач:

1. Извлечение текстовой информации из подписей к фотографиям и формирование на ее основе поисковой системы
2. Поиск любых растровых объектов на изображениях с последующим включением информации о найденных объектах в поисковую систему. К подобным объектам можно отнести лица людей, силуэты, а также объекты, обладающие определенным контуром или текстурой.

Обзор аналогичных работ

Общее процесс исследования можно разделить на несколько независимых этапов исследования по следующим направлениям: методы повышения качества изображений, работа современных средств оптического распознавания текста, анализ текстов, содержащих ошибки и сло-

ва, не встречающиеся в словарях, а также методы поиска объектов на изображениях.

В настоящее время существует большое количество различных методов обработки изображений. Наиболее популярные методы можно встретить в специальных изданиях, посвященных компьютерному зрению [2, 3, 4], а также в статьях Википедии. Данные методы широко применяются для визуального улучшения изображений, а также для предварительной обработки перед расчетом пространственных признаков в задачах классификации растровых объектов. При наличии такого количества альтернативных методов возникает необходимость сравнивать итоговые результаты их применения между собой. К сожалению, обнаружить работы, описывающие подобные исследования, не удалось.

На данный момент на рынке систем оптического распознавания существует несколько коммерчески успешных продуктов и их аналогов с открытым исходным кодом, например, «FineReader», «Google Tesseract» и «CuneiForm». При исследовании альтернативных систем также возникает необходимость в сравнении результатов распознавания. Подобные исследования можно, к примеру, обнаружить в работах [5]. В указанной статье описывается методика сравнения качества сегментации для наиболее популярных систем оптического распознавания символов, как коммерческих, так и систем и открытыми исходными кодами. Основным практическим результатом является перечень рекомендаций по выбору OCR при работе с документами определенного типа. Однако настоящее исследование предполагает распознавание только простого текста, поэтому необходимости в оценке качества сегментации нет.

Проводятся исследования в области приблизительного (нечеткого) поиска по словарю [6, 7, 8]. Например, существует специально разработанный модуль PHP [8], позволяющий выполнять нечеткий поиск на основе расстояния Левенштейна и словаря. Особенностью найденных разработок является то, что во всех случаях поиск осуществляется по нечеткому запросу среди данных фиксированного словаря. Настоящее исследование отличается тем, что словари являются неполными и обновляются в процессе формирования поискового индекса за счет найденных слов.

Методам поиска объектов на изображениях, включая методы извлечения текстовых метрик, посвящено множество исследований. Известный семинар РОМИП собирает множество различных работ по поиску изображений. К примеру, в статьях [9, 10] описаны методы решения однотипных задач поиска изображений по содержанию: поиска нечетких дубликатов, поиска изображений по образцу, которые рассматривались на семинаре РОМИП-2010. Методы поиска нечетких дубликатов и поиска изображений по образцу, описанные в данных статьях обладают наибольшими оценками полноты и точности, среди других методов,

представленных на семинаре. Кроме того, в статье [9] описан метод построения текстовых меток для изображений, основанный на поиске нечетких дубликатов для данного изображения в заранее аннотированной коллекции большого объема.

Описание полученных результатов

На данный момент полностью сформирован метод построения поисковой системы, основанный на информации, извлеченной из подписей. Кроме того, ведутся исследования, связанные с распознаванием лиц на изображениях и распознаванием различных текстур.

Метод построения поисковой системы фактически позволяет построить поисковой индекс, который далее можно использовать с любым существующим поисковым движком. Основными этапами построения индекса являются:

1. Формирование текстовой коллекции
   1.1. Выделение областей, содержащих подписи.
   1.2. Предварительная обработка выделенных областей различными методами повышения резкости изображений, получение конкурирующих изображений подписей.
   1.3. Распознавание с помощью OCR.
   1.4. Анализ полученных текстовых файлов с целью отбора наиболее качественных из них.
2. Анализ текстовой коллекции
   2.1. Морфологический анализ слов
   2.2. Итеративное добавление новых слов с автоматической коррекцией ошибок
3. Формирование индекса и его уточнение с помощью синтаксических правил

Кроме того, метод предусматривает анализ поисковых запросов с целью повышения точности и полноты результатов поиска.

Формирование текстовой коллекции

Из-за низкого качества результатов прямого распознавания в процесс формирования текстовой коллекции были введены дополнительные методы обработки изображений и текста.

С целью исключить влияние областей изображения, не содержащих текст, был разработан специальный эвристический алгоритм выделения подписей. Данный алгоритм выполняет поиск прямоугольной области изображения, содержащий текст, контрастирующий с цветом фона, с точностью около 92%. Ошибки в работе данного алгоритма возникли в результате наличия изображений, на которых присутствовали детали,

помещавшие распознаванию границ областей подписей. Например, к таким деталям относятся выступающие нижние края листа бумаги, не закрытые подписью целиком.

Для повышения резкости изображений и устранения шума были использованы следующие методы [1, 2] (всего в данной работе было рассмотрено 12 методов):

1. Метод порогового отсечения.
2. Методы пространственной фильтрации, основанные на выборе ядра.
   2.1. Методы, применяющие Лапласиан
   2.2. Методы выделения границ
   2.3. Методы сглаживания изображения

Однако вместо того, чтобы использовать только один из них для обработки всей коллекции изображений, было решено обработать коллекцию несколькими наиболее качественными методами параллельно. Данный подход предполагает, что на основе коллекции изображений в результате распознавания будут созданы несколько альтернативных коллекций текстовых файлов, которые затем можно проанализировать и отобрать для каждого исходного изображения по одному текстовому файлу среди всех альтернативных, который будет обладать наибольшей оценкой. Более того, вместо сравнения целых файлов можно выполнить разбиение каждого файла на отдельные элементы и сравнивать между собой соответствующие элементы альтернативных текстовых файлов.

Разбиение текстового файла на элементы осуществляется на основе эвристических правил, которые определяются содержанием подписей. К примеру, к отдельным элементам можно отнести дату, номер фотографии и остальной текст.

Результат сравнения альтернативных элементов или коллекций в данном подходе зависит от метода оценивания отдельного элемента/коллекции. В данной работе были разработаны оценки отдельно для всей коллекции и для отдельного элемента текстового файла. В обоих случаях оценка рассчитывается на основе весов каждого слова, составляющего элемент.

Расчет весов и оценок текстовых элементов

Вес каждого слова зависит от его схожести с одним или несколькими словами из словаря, которые будем называть словами-кандидатами. Рассчитывается вес по следующей формуле: $w = \frac{n - L}{n}$, где n - длина слова, L – расстояние Левенштейна до слова-кандидата. В случае, когда

$L > n$ вес считается равным 0. Вес слова, найденного в словаре, будет равен 1.

Обозначим через F_i текстовый файл (в данном контексте будет называть его просто файл), соответствующий одному и тому же исходному изображению и полученный с помощью обработки этого изображения методом i ($i = \overline{1,n}$, где n - число альтернативных методов). Тогда можно представить файл как множество составляющих его элементов следующим образом: $F_i = \{t_{ij} \mid j = \overline{1,m}\}$, где m - количество элементов файла. Предполагается, что у альтернативных файлов число элементов совпадает, а элементы с одинаковым номером соответствуют одной и той же области текста подписи исходного изображения. Кроме того, необходимо учитывать, что файл является результатом распознавания, и данный результат зависит от множества факторов, в том числе от возможности распознать то или иное слово. Поэтому не исключены ситуации, когда у разных альтернативных файлов (или элементов) количество слов будет разным. Очевидно, что элемент с наибольшим абсолютным весом может оказаться статистически ошибочным, тогда как следует выбрать элемент с наиболее часто встречающимся количеством слов. Следовательно, итоговая оценка для элемента должна зависеть не только от веса всех составляющих его слов и их количества, но также и от количества слов других альтернативных элементов.

Пусть W_{ij} - общий вес, а N_{ij} - количество слов элемента t_{ij}. Тогда оценка элемента t_{ij} будет рассчитываться следующим образом:

$$R_{ij} = S_{ij} \cdot D_{ij},$$

где $S_{ij} = \begin{cases} \dfrac{W_{ij}}{N_{ij}}, N_{ij} > 0 \\ 0, N_{ij} = 0 \end{cases}$, $D_{ij} = \begin{cases} \sqrt{1 - \left(\dfrac{N_{ij} - \overline{N_j}}{\overline{N_j}} \right)^2}, N_{ij} \le 2 \cdot \overline{N_j} \\ 0, N_{ij} > 2 \cdot \overline{N_j} \end{cases}$ и

$$\overline{N_j} = \frac{\sum_{k=1}^{n} N_{kj}}{n}.$$

В приведенной выше формуле S_{ij} играет роль относительной оценки элемента, определяющей долю правильно распознанных символов в

нем. Фактор D_{ij} в данном случае определяет отклонение количества слов элемента t_{ij} от среднего, т.е. от того количества слов, которое следовало бы ожидать после распознавания.

Сравнение оценок отдельных элементов позволяет составлять результирующий текстовый файл из элементов альтернативных текстовых файлов, обладающих наибольшей оценкой:

$$F^* = \{t_j^* \mid t_j^* = t_{i^* j}, i^* = \arg\max_i (R_{ij}), i = \overline{1,n},\ j = \overline{1,m}\}.$$

Примеры сравнения коллекций, соответствующих выбранным методам обработки изображений, и элементов одного текстового файла представлены в таблицах табл. 6 и табл. 7 (серым цветом выделены элементы с наибольшей оценкой). Названия коллекций, представленных в таблицах табл. 1 и табл. 2, соответствуют названиям методов обработки изображений и описаны в [1].

Табл. 1. Пример сравнения альтернативных коллекций.

Имя коллекции	Средняя оценка коллекции	Максимальная оценка файла коллекции
Cut	0.67	0.88
EAELaplace	0.66	0.93
AScharr	0.53	0.80
Smooth	0.55	0.86
Original	0.52	0.77

Табл. 2. Пример сравнения альтернативных элементов одного и того же файла.

Имя коллекции	Оценка текста	Оценки атрибутов	
		Дата	Номер
Cut	0.8	1	0.6
EAELaplace	0.82	0.9	0.9

Анализ текстовой коллекции

Данный этап опирается на результаты первичного анализа текстовой коллекции, который проводится с целью оценки веса каждого слова, при этом определяется слово, наиболее близкое к данному слову, и соответствующее расстояние Левенштейна [11]. Для каждого распознанного слова на данном этапе извлекается тематическая информация на основе имеющихся тематических словарей, а также с помощью морфологического анализа, в результате чего определяется тематика слова. В данной работе используются следующие тематические словари:

1. Имена
2. Географические названия
3. Строительная техника
4. Сооружения
5. Общий словарь (приведен здесь, поскольку является словарем по умолчанию)

В качестве морфологического анализатора используется Mystem (Яндекс) [12], выявляющий атрибут имени или географического названия.

Процесс добавления новых слов требует участия пользователя. При добавлении нового слова кроме самого слова указывается также дополнительная информация:

1. Часть речи
2. Нормальная форма слова
3. Словоформы (наиболее часто встречающиеся)
4. Тематика

Процесс коррекции ошибок может выполняться полностью автоматически, в случае определения необходимых пороговых значений. В результате добавления нового слова часть слов, которые до этого были признаны нераспознанными или ошибочными могут изменить свой статус, поэтому на следующем шаге требуется пересчет весов всех слов, имеющих вес меньше 1. В случае, когда очередное рассматриваемое слово будет достаточно близко к новому слову можно выполнить коррекцию слова без участия пользователя. Таким образом, необходимо задать порог близости, заключающийся в абсолютном или относительном значении расстоянии Левенштейна.

Общая схема анализа текстовой коллекции представлена на рис. 2.

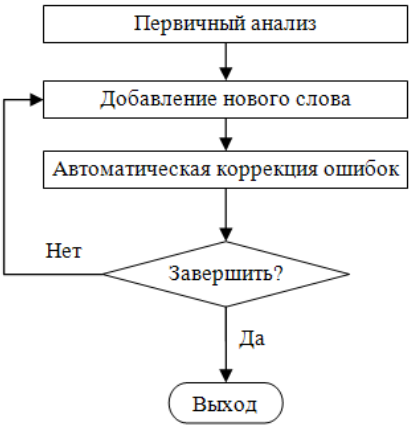


Рис. 2. Общая схема анализа текстовой коллекции.

Кроме решений о коррекции система должна также принимать решение о близости к тому или иному слову в случае, когда есть несколько слов-кандидатов с одинаковым расстоянием Левенштейна. Основным решением этой проблемы является определение тематики каждого кандидата и отнесение рассматриваемого слова к одной из них, однако в данной работе указанный метод на данный момент не реализован.

Формирование индекса

Построение полнотекстового индекса выполняется с помощью приведения всех слов к нормальной форме. Для этого используется как морфологический анализ (Mystem), так и дополнительные словари, заполненные пользователем в процессе добавления новых слов. Формирование индекса можно осуществить с помощью любой СУБД, поддерживающей полнотекстовый поиск. Суть данной операции заключается в заполнении специально подготовленной таблицы, включающей в себя информацию о файлах коллекции, атрибуты, найденные в текстах подписей, а также поле для хранения нормальных форм слов основного текста. Последнее поле индексируется для полнотекстового поиска средствами СУБД. В данной работе использовалась СУБД MS SQL Server, реализующий полнотекстовый поиск, основанный основан на алгоритме ранжирования BM25 [13].

Кроме того, с помощью специальных синтаксических правил поиска и подстановки можно заменять отдельные словосочетания в индексе на уникальные ключевые слова, тем самым повышая точность поиска. Данный метод называется контекстным поиском. Каждое правило мож-

но представить в виде ориентированного набор элементов определенного типа. В данной работе были выделены следующие типы элементов правил:

1. Строковая константа
2. Лексема (токен), например, слово, состоящее из букв алфавита.
3. Лексическая группа (часть речи)
4. Тематическая группа (слово относится к одному из тематических словарей)
5. Онтологическая группа (слово принадлежит определенному таксономическому узлу)

Данные правила позволяют выполнять поиск ключевых словосочетаний не только в текстовой коллекции, но и в поисковом запросе, что в свою очередь позволяет уточнять также и результат поиска. Например, такие часто встречаемые словосочетания, как «шлюз 17» можно заменить одним словом «шлюз_17», тем самым сделав его ключевым для данной коллекции. Также возможно осуществлять контекстный поиск различных сокращений, которые в большинстве случаев никак не распознаются. Например, словосочетание «ж. д.» можно заменить словом «железнодорожный».

Уточнение и расширение поискового запроса

Уточнение поискового запроса возможно с помощью описанного выше метода контекстного поиска, при котором отдельные словосочетания, содержащиеся в запросе, заменяются ключевыми словами. Например, для словосочетания «шлюз 17» использование ключевого слова «шлюз_17» позволяет искать только те файлы, которые содержат упоминания шлюз именно с таким номером, тогда как, файлы, в которых встречаются словосочетания «шлюз 15» останутся неучтенными, что в результате повысит точность поиска.

В отличие от точности, повышению полноты поиска сопутствует расширение поискового запроса дополнительным набором ключевых слов. С помощью онтологии/тезауруса можно расширить запрос за счет включения понятий онтологии/тезауруса, связанных с ключевыми словами запроса различными отношениями. В зависимости от типов включаемых отношений можно расширить запрос:

1. С помощью синонимов
2. В глубину за счет отношений «род-вид» и «часть-целое».
3. В любую сторону на 1 шаг, рассматривая и другие отношения, например ассоциативные.
4. С помощью комбинации различных отношений

На рис. 13 представлен пример семантической сети, поддерживающей различные отношения. Для удобства восприятия были введены

следующие обозначения: В – вид, Ч – часть, С – синоним, БР – частное ассоциативное отношение, означающее близкое расположение соответствующих объектов в пространстве. Прямоугольником на данной схеме изображены понятия, сглаженным прямоугольником – синонимы, овалами – частные объекты.

С помощью представленного на рисунке рис. 3 примера рассмотрим механизм расширения запроса за счет онтологии/тезауруса. Пусть поисковый запрос содержит ключевое слово «шлюз_17». В случае, когда расширение в глубину направлено от листьев к корню, к запросу могут добавиться слова «канал» или «шлюз» (за счет отношений «род-вид» и «часть-целое» соответственно). В случае включения в механизм расширения ассоциативных связей, например связь «Близкое расположение» («БР»), в запрос может добавиться слово «шлюз_15». Если в запросе указано ключевое слово «плотина» и включена связь «синонимы», то в запрос будет добавлено слово «дамба».

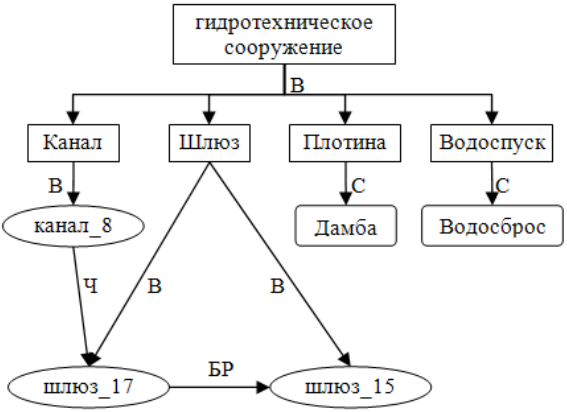


Рис. 3. Пример семантической сети с различными поддерживаемыми отношениями.

Описание метода уточнения и расширения запроса

Метод контекстного преобразования запроса позволяет с помощью определенного набора команд задавать операции, которые должны быть выполнены над множеством слов. При одновременном использовании контекстного поиска можно выполнять преобразование над цепочкой совпадений, являющейся результатом поиска по контекстному правилу. В данном случае цепочкой совпадений может быть либо подмножество слов одного файла в виде одной последовательности, в котором каждое слово соответствует элементу контекстного правила, либо все слова

файла, содержащего совпавшую последовательность. В данной работе при определении контекстного правила поиска можно определять контекстные преобразования с помощью специального синтаксиса. Введенная пользователем инструкция трансформируется в набор простейших команд, кодируемых числовым идентификатором, и исходных данных, в результате чего инструкцию можно хранить в базе данных, быстро извлекать при выполнении запроса и выполнять преобразование. Для реализации данного метода использовались видоизмененные комбинации шаблонов проектирования «Команда» и «Интерпретатор».

Рассмотрим пример преобразования запроса с целью повышения точности поиска путем замены словосочетаний. Будем выполнять преобразование над всеми запросами, в которых встречается словосочетание «шлюз <число>«. Данное словосочетание будем заменять одним словом: «шлюз_<число>«. При этом будем считать, что в индексе были проведены подобные преобразования и в нем определены вхождения слов «шлюз_15» и «шлюз_17» вместо вхождений отдельных слов «шлюз», «15» и «17». Для выполнения поставленной задачи зададим правило контекстного поиска:

const «шлюз», simple number

и укажем следующее преобразование:

replace from {0}, to {1} with value concat {item {0}, «_», item {1}}.

Приведенная выше команда определяет, что слова в совпавшей последовательности с индексами 0 и 1, которые будут соответствовать элементам контекстного правила «шлюз» и «число», будут заменены на конкатенацию этих двух слов с добавлением символа «_» между ними.

Поскольку преобразования выполняются в процессе обработки поискового запроса, необходимо обеспечить высокую скорость выполнения преобразований. Для этого можно использовать специальный индекс, ключами которого будут являться строки-константы, являющиеся элементами контекстных правил, тогда как соответствующими значениями могут быть наборы <правило поиска, преобразование, номер вхождения>. Третий элемент набора необходим для ориентирования поискового правила относительно элемента-константы. В этом случае для каждого слова запроса можно проверять в данном индексе наличие соответствующего правила и в случае, когда найдено вхождение, применять соответствующее преобразование.

Дальнейшее развитие работы

Результат работы вышеописанного метода формирования поискового индекса напрямую зависит от качества используемых словарей и, в

особенности, от мощности тематических словарей. Кроме того, использование контекстного поиска предполагает наличие тезауруса, формирование которого еще не закончено. Поэтому первоочередной нерешенной задачей на данный момент является формирование тематических словарей и тезауруса по заданной тематике. Для этого предполагается использовать существующие в сети Интернет лингвистические и морфологические ресурсы такие, как «Викисловарь» и другие, с целью извлечения информации автоматическим либо автоматизированным методом. Подобный метод описан в [14].

Другим направлением развития работы является анализ изображений с целью обнаружения лиц, контуров и текстур. На данный момент выполняются исследования возможностей библиотеки OpenCV, в частности, функции обнаружения объектов detectMultiScale [15]. Данная функция использует заранее подготовленные классификаторы для поиска различных объектов таких, как лица, силуэты, глаза и др. Основной проблемой на данный момент является привязка найденных изображений лиц к поисковому индексу, а также сравнение изображений лиц одного и того же человека, обнаруженных на нескольких изображениях и обладающих разными пространственными параметрами.

Кроме того, проводятся исследования, связанные с анализом текстур методом моментов. В качестве отправной точки было использована статья [16], в которой описан математический аппарат метода моментов, предложена формула расчета характеристик, а также приведены примеры сегментации текстур с использованием данного метода. К сожалению, воспроизвести результаты описанных в статье экспериментов полностью не удалось, тем не менее, удалось выполнить сегментирование изображение, состоящее из 2-х текстур, с долей правильно сегментированных пикселей в 97%.

Предполагается применить данный метод для поиска объектов с определенной текстурой, например, камней или воды. Кроме того, планируется применить один или несколько методов поиска изображений и методов извлечения текстовых метрик, предложенных участниками семинаров РОМИП.

Выводы

Предложенный метод формирования текстовой коллекции позволяет существенно повысить качество распознанного текста за счет использования различных методов обработки изображений и последующего сравнения результатов обработки изображений между собой.

Результаты сравнения результатов обработки изображений показали, что наилучшим образом работают методы повышения резкости на основе фильтров Лапласа с большим размером окна в сочетании с операцией выравнивания по гистограмме с исходным изображением.

Из всех рассмотренных средств оптического распознавания символов наилучшие результаты показала система «FineReader».

Предложенный в данной статье метод анализа текстовой коллекции позволяет повысить качество полнотекстового индекса.

Предложенные методы уточнения и расширения запроса за счет контекстных преобразований позволяют повысить точность и полноту поиска.

Список источников

1. Талбонен А. Н., Рогов А. А. Анализ машинописных подписей к фотографиям в цифровом альбоме / Электронные библиотеки: перспективные методы и технологии, электронные коллекции: Труды XII Всероссийской научной конференции RCDL'2010.- Казань: Казан. ун-т, 2010. С. 422-429.

2. Гонсалес Р., Вудс Р. Цифровая обработка изображений / Р. Гонесалес, Р. Вудс. - М.: Техносфера, 2005. - 1072 с.

3. Методы компьютерной обработки изображений / Под ред. В. А. Сойфера – 2-е изд., испр. – М.: ФИЗМАТЛИТ, 2003. – 744 с.

4. Форсайт Д., Понт Ж. Компьютерное зрение. Современный подход.: Пер. с. англ. – М.: Издательский дом «Вильямс», 2004. – 928 с.: ил. – Парал. тит. Англ.

5. Кулешов С. В., Смирнов С. В. Методы сегментации OCR-систем в задачах ав-томатической обработки архивных документов / Труды СПИИРАН. Научный, научно-образовательный, междисциплинарный журнал с базовой специализа-цией в области информатики, автоматизации и прикладной математики. Вы-пуск № 1(16), 2011. – СПб. Наука, 276 с. С. 110-122.

6. Baeza-Yates R, Navarro G. «Fast Approximate String Matching in a Dictionary». Proc. SPIRE'98. IEEE CS Press. pp. 14–22.

7. Mihov S., Shultz K. U. Fast Approximate Search in Large Dictionaries.

8. Approximate/fuzzy string search in PHP [Электронный ресурс]. URL: http://elonen.iki.fi/code/misc-notes/appr-search-php/ (дата обращения: 16.02.2011).

9. Пименов В. Ю. Простые методы поиска изображений по содержанию / Россий-ский семинар по Оценке Методов Информационного Поиска. Труды РОМИП 2010. – Казань: Казан. ун-т, 2010. – 210. с., с 69 – 79.

10. Слесарев А. В., Мучник И. Б., Михалев Д. К. Яндекс на РОМИП 2010: Поиск похожих изображений и дубликатов / Российский семинар по Оценке Методов Информационного Поиска. Труды РОМИП 2010. – Казань: Казан. ун-т, 2010. – 210. с., с 148 – 153.

11. Гасфилд Д. Строки, деревья и последовательности в алгоритмах: Информатика и вычислительная биология / Пер. с англ. И. В. Романовского. — СПб.: Невский Диалект; БХВ-Петербург, 2003. — 654 с.

12. О программе mystem [Электронный ресурс]. URL: http://company.yandex.ru/technology/mystem/ (дата обращения: 16.02.2011).

13. Hugo Zaragoza, Nick Craswell, Michael Taylor, Suchi Saria, and Stephen Robertson. Microsoft Cambridge at TREC-13: Web and HARD tracks [Электронный ресурс]. URL: http://trec.nist.gov/pubs/trec13/papers/microsoft-cambridge.web.hard.pdf (дата обращения: 16.02.2011).

14. Крижановский А. А. Построение машинно-читаемого словаря на основе русского викисловаря / Труды СПИИРАН. Выпуск 11, 2009. с. 228-234.

15. Cascade Classification — opencv v2.1 documentation [Электронный ресурс]. URL: http://opencv.willowgarage.com/documentation/cpp/objdetect_cascade_classification.html?highlight=detect (дата обращения: 16.02.2011).

16. Tuceryan M. Moment Based Texture Segmentation [Электронный ресурс]. URL: http://masters.donntu.edu.ua/2003/kita/korsakova/library/moment-paper.pdf (дата обращения: 16.02.2011).

Применение онтологии при синтезе изображения по тексту

Д. Усталов[1], А. Кудрявцев[2]

[1] dmitry@eveel.ru, [2] vt@dpt.ustu.ru

Уральский федеральный университете имени первого Президента России Б. Н. Ельцина, Екатеринбург, Россия

Аннотация. В статье представлен способ применения онтологии в процессе формирования описателей в системах синтеза изображения по тексту. Использование онтологии позволяет реализовать слабую связность компонентов системы, унифицировать представление акторов и их поведения, а также предоставить возможность верификации информационных ресурсов системы.

Ключевые слова: синтез изображения по тексту; онтология; тезаурус; фрейм; семантическое представление текста; Web Ontology Language; Resource Description Framework..

Введение

Одна картинка может заменить тысячу слов. Актуальность задачи синтеза графических изображений по текстам на естественном языке обусловлена существованием многочисленных предметных областей, в которых главную роль играет наглядность представления текстовой информации: изучение иностранных языков [1], воспроизведение сцен дорожно-транспортных происшествий по описанию [2], реабилитация людей с черепно-мозговыми травмами [3] и т. д.

Игнатов Д. И., Яворский Р. Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16–18 марта, 2012

Разрабатываемая TTP–система[1] Utkus [4] общего назначения ориентирована на работу с небольшими текстами из 1–3 предложений: отрывками из детской литературы, записями в микроблогах, аннотациями к новостям, комментариями на Web-сайтах. Такие тексты легко подвергаются как автоматической обработке, так и последующей визуализации [5].

В TTP-системах процесс синтеза изображения по тексту проходит в три этапа [6]:

1) этап *лингвистического анализа* исходного текста — разбиение текста на составляющие, его морфологическая и синтаксическая разметка, а также получение семантического представления текста;
2) этап *формирования описателей* текста — генерация набора описателей, соответствующих семантическому представлению исходного текста;
3) этап *синтеза изображения* — построение векторного или растрового изображения на основе множества графических примитивов, расположенных в соответствии с описателями исходного текста.

Постановка задачи

Этап формирования описателей текста состоит из нескольких шагов:

1) получение списка семантических элементов (объектов и действий) из результатов лингвистического анализа исходного текста;
2) интерпретация семантического представления, то есть получение ответов на вопросы «кто?», «что?», «когда?», «где?», «как?» в случае, если актор, объект, время, местоположение, и вид действия не были перечислены в тексте;
3) назначение описателей для каждого семантического элемента;
4) разрешение неявных и конфликтующих ограничений описателей;
5) последовательное применение описателей для формирования данных об итоговой сцене.

Аналогичные работы

Несмотря на наличие большого количества полнофункциональных аналогов [2–3,5–8], решение задачи формирования описателей текста обнаружено только в работе [6], посвящённой системе WordsEye, выполняющей построение трёхмерных графических сцен по описанию на упрощённом подмножестве английского языка. В системе WordsEye приняты следующие меры:

[1] TTP–система (от англ. *Text–to–Picture*) — система синтеза изображения по тексту.

1) тезаурус WordNet применяется для выявления семантических отношений между отдельными словами;
2) при обработке текста выполняется отображение заранее подготовленных фреймов на обнаруженные в тексте синтаксические группы с целью получения дополнительной информации об акторах: цвет, размер, и т. д.;
3) поведение, реализуемое в виде известных действий (глаголов), описано в виде правил изображения[2], определённых в программе на языке Lisp в декларативном стиле;
4) в качестве инструмента для визуализации применяется коммерческая система трёхмерной анимации Izware Mirai, работающая с моделями из библиотеки Viewpoint Model Library.

Можно отметить два значительных недостатка такого решения:

1) несмотря на богатые возможности языка программирования Lisp, его применение затрудняет пополнение фреймовой базы из-за высоких требований к предварительной подготовке разработчиков;
2) работа в трёхмерном пространстве требует значительных усилий и ресурсов, что далеко не всегда оправдывается качеством результата: зачастую достаточно обойтись двумерными изображениями [1].

Предлагаемое решение

Аналогично работе [6], мы рассматриваем акторов в рамках объектной парадигмы:

1) акторы имеют свойства: координаты, угол поворота, и т. д.;
2) акторы имеют методы: функции, определённым образом изменяющие свойства акторов: падать, лежать, и т. д.

Мы предлагаем формализовать в виде онтологии все возможные сведения об акторах: их характеристики и отношения.

Мы также предлагаем разделить онтологию, тезаурусные ресурсы, фреймы, и галереи графических примитивов, обеспечив слабую связность этих компонентов TTP–системы (рис. 1, 2):

— слова и семантические отношения между ними представлены в тезаурусе;
— к каждому слову в тезаурусе может быть привязано несколько графических примитивов из галереи: несмотря на то, что слова *кот*

[2]Правило изображения (от англ. *depiction rule*) — правило преобразования семантического элемента в графический описатель.

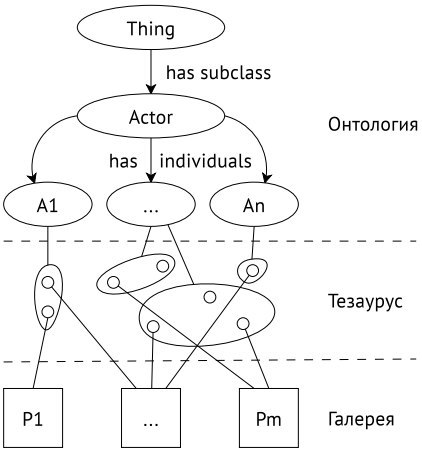


Рис. 1. Связь онтологии, тезауруса и галереи графических примитивов

и *кошка* являются антонимами по признаку пола, они являются являются гипонимами по отношению к слову *зверь*;

- онтология содержит в себе класс *Actor*, экземпляры которого привязываются к синсетам[3] тезауруса. Таким образом, для каждого набора синсетов может быть задан экземпляр класса *Actor* со специфичными для него свойствами;
- для экземпляров класса *Actor* описаны свойства–объекты, привязанные к глагольным синсетам в тезаурусе, выражающие все возможные отношения между акторами (например, `fall(actor)` и `fallTo(actor1 actor2)`);
- для экземпляров класса *Actor* описаны свойства–значения, выражающие параметры этих акторов (например, `positionX` и `rotation`);
- фреймы, описывающие поведение каждого свойства–объекта (рис. 2) определяются в отдельном документе на языке XML.

Привязка элементов онтологии к синсетам в тезаурусе выполняется при помощи механизма аннотаций языка OWL. Важно отметить, что к одному элементу может быть привязано несколько синсетов. Эти синсеты могут относиться к тезаурусам разных языков, благодаря встроенному в OWL механизму интернационализации.

Примеры. Класс *Actor* является прямым потомком класса *Thing*:

```
<owl:Class rdf:about="World.owl#Actor"/>
```

[3]Синсет (от англ. *Synset*) — множество синонимов.

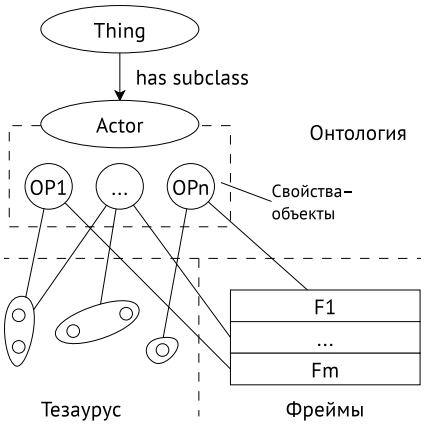


Рис. 2. Связь онтологии, тезауруса и хранилища фреймов

Экземпляры класса *Actor* (рис. 3) могут быть привязаны к синсетам в тезаурусе при помощи аннотаций:

```
<owl:NamedIndividual
  rdf:about="World.owl#Man">
    <rdf:type
      rdf:resource="World.owl#Actor"/>
    <synset xml:lang="ru">2039</synset>
    <synset xml:lang="ru">2040</synset>
    <synset xml:lang="ru">238</synset>
    <synset xml:lang="ru">6939</synset>
    <synset xml:lang="ru">75</synset>
</owl:NamedIndividual>
```

Свойства–объекты также определяются для класса *Actor* и предикаты, которыми оперирует система. Как уже говорилось выше, свойства–объекты связаны с фреймами, описывающими их поведение.

Между свойствами–объектами может существовать отношение эквивалентности. В нашем подходе, SPO–тройки[4] `(упасть человек)` и `(упасть человек стул)` будут отнесены к разным свойствам-объектам: `fall` и `fallTo`:

```
<owl:ObjectProperty
  rdf:about="World.owl#fall">
```

[4]SPO (от англ. *Subject-Predicate-Object*) — кортеж вида <Субъект, Предикат, Объект>.

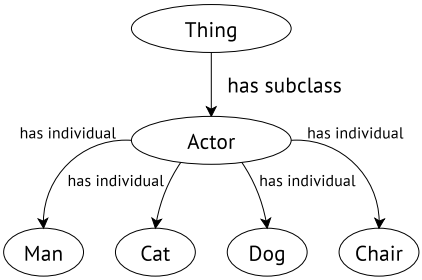


Рис. 3. Фрагмент онтологии с экземплярами класса *Actor*

```
    <synset xml:lang="ru">106</synset>
    <synset xml:lang="ru">397</synset>
    <synset xml:lang="ru">406</synset>
    <rdfs:domain
      rdf:resource="World.owl#Actor"/>
    <owl:equivalentProperty
      rdf:resource="World.owl#fallTo"/>
</owl:ObjectProperty>

<owl:ObjectProperty
  rdf:about="World.owl#fallTo">
    <synset xml:lang="ru">106</synset>
    <synset xml:lang="ru">397</synset>
    <synset xml:lang="ru">406</synset>
    <rdfs:domain
      rdf:resource="World.owl#Actor"/>
    <rdfs:range
      rdf:resource="World.owl#Actor"/>
</owl:ObjectProperty>
```

Для того, чтобы обнаруженные системой свойства–объекты были изображены на результирующем изображении, необходимо присвоить каждому заданному свойству–объекту специфичное поведение, чтобы ТТР–система могла выполнить различные преобразования акторов и их характеристик. Такое поведение описывается в фреймовом виде в отдельном XML–документе. Для свойства–объекта `fall`, фрейм примет вид:

```
<frame rdf:about="World.owl#fall">
    <transforms
```

```
      property="rotation"
      delta="60">
        <yield id="subject" />
    </transforms>
</frame>
```

В данном примере, при реализации свойства–объекта `fall`, угол наклона субъекта этого предиката будет изменён на 60° по часовой стрелке.

Реализация

В экспериментальной реализации системы Utkus, программа преобразования семантического представления написана на языке Ruby:

1) благодаря своей доступности, используется синтаксический анализатор АОТ [9];
2) из дерева зависимостей каждого предложения, полученного от анализатора АОТ, извлекаются только глагольные группы и входящие в них именные группы. Выделенные синтаксические группы отображаются в SPO–тройки;
3) онтология описана в формате RDF/OWL при помощи специализированного редактора Protegé;
4) используется словарь русского языка, хранящий исключительно синсеты [10];
5) применяются графические примитивы из коллекции The Noun Project [11], предварительно откадрированные, преобразованные в растровый вид (PNG) и привязанные к синсетам, содержащим имена существительные, при помощи встроенных тегов и категорий;
6) вывод графики осуществляется при помощи библиотеки GD2 в виде растровых PNG–изображений размером 320×240. При рендеринге, позиционирование объектов выполняется в рамках сетки 8×8.

В качестве примера, системой Utkus сгенерировано три изображения[5] (рис. 4), соответствующих текстам:

1) Человек и закон.
2) Человек пошёл на кухню и налил чай.
3) Пёс упал на стул.

[5] В целях экономии места, изображения были кадрированы.

(a) Человек и закон (b) Человек пошёл на кухню и налил чай (c) Пёс упал на стул

Рис. 4. Графическое представление текстов

Заключение

Представлен способ организации информационных ресурсов ТТР–системы на этапе преобразования семантического представления текста, предполагающий слабую связность онтологии, тезауруса, фреймов и графических примитивов.

Основными преимуществами данного способа можно отметить:

1) удобство развития и модификации всех информационных ресурсов, используемых ТТР–системой:
   — для редактирования онтологии можно использовать любой доступный редактор онтологий (например, Protegé);
   — для модификации фреймов можно использовать любой текстовый редактор, или же специализированный XML-редактор;
   — способ редактирования тезауруса и графических примитивов зависит от используемых инструментов (в текущей реализации тезаурус представлен набором CSV–файлов).

2) применение онтологии в открытом формате RDF/XML позволяет повторно использовать данные ресурсы в других системах и предметных областях;

3) инструменты верификации (например, системы логического вывода над OWL–онтологиями) для использованных форматов представления данных позволяют контролировать качество информационных ресурсов.

Данный способ апробирован в разрабатываемой системе Utkus, благодаря чему были получены изображения на рис. 4(a), 4(b), 4(c).

Направления дальнейшей работы. Имеется несколько направлений для дальнейших исследований:

1) подключить полноценный тезаурус для унификации тезаурусных ресурсов (например, Russian WordNet [12]);
2) отказаться от синтаксического анализатора АОТ в пользу парсеров на основе машинного обучения;
3) решить проблему многозначности предикатов [13] при формировании семантического представления;
4) провести эксперименты над прототипом системы Utkus и, при необходимости, внести уточнения в онтологию и фреймы.

Список источников

1. Yoshii M., Flaitz J. Second language incidental vocabulary retention: The effect of text and picture annotation types // CALICO journal. — 2002. — V. 20, № 1. — P. 33–58.
2. Åkerberg O., Svensson H., Schulz B., Nugues P. CarSim: an automatic 3D text-to-scene conversion system applied to road accident reports // In proc. of The 10th Conf. on European chapter of the Association for Computational Linguistics 2003 — V. 2 — P. 191–194.
3. Goldberg A., Rosin J., Zhu X. Dyer C. Toward Text-to-Picture Synthesis // In NIPS 2009 Mini-Symposia on Assistive Machine Learning for People with Disabilities.
4. `http://utkus.eveel.ru` — Utkus: a Text-to-Picture Synthesis System — 2012.
5. Zhu X., Goldberg A., Eldawy M., Dyer C. A text-to-picture synthesis system for augmenting communication // In proc. of The National Conf. of the Artifical Intelligence 2007 — V. 22 — P. 1590–1595.
6. Coyne B., Sproat R. WordsEye: an automatic text-to-scene conversion system // In proc. of The 28th ACM Annual Conf. on Computer graphics and interactive techniques 2001 — P. 487–496.
7. Li H., Tang J., Li G., Chua T. Word2Image: Towards Visual Interpretation of Words // In proc. of The 16th ACM Int. Conf. on Multimedia 2008 — P. 813–816.
8. Yamada A., Yamamoto T., Ikeda H., Nishida T., Doshita S. Reconstructing spatial image from natural language texts // In proc. of The 14th conference on Computational linguistics 1992 — V. 4 — P. 1279–1283.
9. `http://aot.ru/docs/synan.html` — АОТ: Технологии: Синтаксический анализ — 2012.

10. http://speakrus.ru/dict/index.htm — Словари русского языка для скачивания — 2012.
11. http://thenounproject.com — NounProject — 2012.
12. http://www.wordnet.ru — Russian Wordnet — Русский WordNet — 2012.
13. Fomichov, V. A comprehensive mathematical framework for bridging a gap between two approaches to creating a meaning-understanding web // International Journal of Intelligent Computing and Cybernetics. — 2008. — V. 1, № 1. — P. 143–163.

Определение компетенций участников конкурса

А. Воробьев

alvorobyev88@gmail.com

Мех.-мат. ф-т МГУ им. М. В. Ломоносова, Москва, Россия

Аннотация. В современной жизни в самых разных ситуациях появляется необходимость сравнения работ (результатов работы) разных людей. Например, это может быть объявленный компанией тендер, распределение грантов научной организацией или литературный конкурс. Причем иногда целесообразно поручить это сравнение этим же людям. Такая ситуация возможна, когда участники являются экспертами в данной области. Также это имеет смысл, если число участников большое, в силу чего их усредненная оценка проекта будет не менее объективной, чем оценка комиссии из небольшого числа экспертов, работу которой, к тому же, сложно организовать в виду большого числа проектов.

Тогда перед нами встает ряд вопросов: как на основе поставленных оценок сформировать итоговую оценку для каждой работы? а не сможем ли мы еще и оценить компетентность каждого участника как оценщика? а не будет ли связана эта компетентность с итоговыми оценками его работ? Попытке ответить на эти вопросы посвящена данная работа.

Ключевые слова: экспертные оценки; компетентность экспертов; характеристики участников.

Игнатов Д. И., Яворский Р. Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16–18 марта, 2012

Общая постановка задачи

Рассмотрим некоторый конкурс проектов. В рамках этого конкурса его участники предлагают проекты и оценивают качество проектов, предложенных другими участниками. Важным условием является возможность участника представить на конкурс любое количество проектов.

На основе всех оценок, выставленных участниками проектам, нам необходимо оценить качество проектов и характеристики участников как оценщиков проектов и как авторов (создателей) проектов.

Для этого вводится следующая модель, формализующая действия участников.

Описание модели

Предполагается, что каждый участник наделен двумя характеристиками, принимающими численные значения: C (Creator) - его способность к созданию проектов, E (Evaluator) - его способность к оцениванию качества проектов. Считаем, что сначала по закону распеделения, соответствующему характеристике участника C, генерируется случайное число проектов N, которые представит этот участник. Далее в этом количестве, независимо друг от друга, по некоторому закону, определяемому также характеристикой C этого участника, случайно задаются уровни качества L (Level) проектов, которые он подаст. Далее оценка e, которую участник выставляет какому-либо проекту, является случайной величиной, распределение которой определяется характеристикой E этого участника и уровнем качества L этого проекта.

Мы считаем, что число представленных участником проектов не зависит от его характеристик C и E, поэтому знание распределения этого числа нам никак не поможет их оценить. Значит, и наблюдения этого числа в рассмотрение не принимаем.

Упомянутые распределения уровня качества проекта и оценки при фиксированных значениях C и E, L соответственно могут быть полностью заданы, но могут быть и заданы с точностью до параметров. Поэтому в общем виде первое распределение определяется неизвестным вектором параметров P, второе — вектором Q.

Таким образом, стоящая перед нами задача может быть сформулирована следующим образом:

Нам дан трехмерный массив оценок качества проектов

$$(e_{i,j,k})_{i=1..n,j=1..n,k=1..m_j},$$

где $e_{i,j,k}$ — оценка качества, данная участником i k-ому проекту, представленному участником j (нумерация внутри его проектов);

n — количество участников конкурса;

m_j — количество проектов, представленных участником j.

Необходимо оценить векторы:

$(C_j)_{j=1..n}$ — вектор характеристик участников как создателей проектов;

$(E_j)_{j=1..n}$ — вектор характеристик участников как оценщиков проектов;

$(L_{j,k})_{j=1..n,k=1..m_j}$ — вектор уровней качества проектов;

P — вектор параметров распределения уровня качества проекта;

Q — вектор параметров распределения оценки качества проекта.

Метод решения задачи

Предлагается произвести оценку параметров модели методом наибольшего правдоподобия. То есть оценкой параметров будет служить совокупность их значений, при условии которой максимальна условная вероятность достижения наблюдаемых значений оценок качества проектов.

Для дальнейшего изложения метода нам нужно определиться, имеем мы дело с дискретным или непрерывным случаем в плане областей изменения величин C, E, P, Q, L, e.

Естественным является введение дискретных распределений для величин e, L. Действительно, что касается параметров e, считается (см. [1]), что, глядя на перечень объектов, люди могут сравнивать их друг с другом по различным параметрам, но выставление численных оценок параметров объектов для них является неестественным. Ограниченную дискретную шкалу с небольшим числом делений можно почти однозначно сопоставить упорядочиванию объектов, тогда как непрерывная или бесконечная дискретная всегда содержит больше информации, чем упорядочивание.

Далее, очевидно, удобно взять для этих величин одну и ту же шкалу. Тогда можно считать, что участник старается оценить именно уровень качества проекта L, когда выставляет свою оценку e. Итак, распределения случайных величин e, L заданы на одном конечном дискретном множестве.

Договоримся, что шкалы для оцениваемых параметров C, E, P, Q также лежат в конечном дискретном множестве. Для параметров C, E это сделает результат более легким для интерпретации. А дискретность параметров P, Q примем для полной дискретности всей области изменения переменных. Конечность шкал введем для ускорения поиска оптимальных значений. Такая необходимость появляется из-за большого количества участников и проектов.

Итак, оценим наши параметры модели методом наибольшего правдоподобия. Заметим, что при таком подходе, если мы будем подбирать оптимальные значения величин $(L_{j,k})_{j=1..n,k=1..m_j}$ наряду с остальными, то сделаем их параметрами модели, никак не учтя, что они являются случайными величинами, о распределении которых у нас есть некоторая информация. И влияние параметров C, P на наблюдаемые оценки полностью пропадет. Поэтому оценку параметров модели необходимо разбить на два шага. Предложим 5 вариантов сделать это.

Для удобства в записи массива значений $(A_{i,j,k})_{i=1..n,j=1..n,k=1..m_j}$ будем опускать индексы, пробегающие все возможные значения, и писать просто $(A_{i,j,k})_{k=1..m_j}$, а также $(A_{i,j})$ вместо $(A_{i,j})_{i=1..n,j=1..n}$. Также договоримся, что во всех формулах операторы умножения и суммирования пробегают все возможные значения указанных под ними переменных.

1) Сначала будем подбирать параметры C, E, P, Q, считая, что величины L генерируются в соответствии со своим законом. То есть распределение $(e_{i,j})$ фактически будет зависеть от параметров C, E, P и Q.
Итак, на первом шаге оценки параметров C,E,P,Q задаются следующим образом:

$$\begin{aligned}&((\widetilde{C_j}),(\widetilde{E_j}),\widetilde{P},\widetilde{Q}) = \\ &= \arg(\max_{\substack{(C_j),(E_j),\\P,Q}} P((e_{i,j,k})_{k=1..m_j}|(C_j),(E_j),P,Q)) = \\ &= \arg(\max_{(C_j),(E_j),P,Q} \prod_{i,j,k}[\sum_{L_{j,k}}(P(e_{i,j,k}|E_j,Q,L_{j,k})P(L_{j,k}|C_i,P))]).\end{aligned} \tag{1}$$

Теперь мы располагаем двумя способами оценить L тем же методом максимального правдоподобия.

а) Зафиксируем параметры E и Q равными их первой оценке и оценим величины L:

$$\begin{aligned}&(\widetilde{L_{j,k}})_{k=1..m_j} = \\ &= \arg(\max_{(L_{j,k})_{k=1..m_j}} P((e_{i,j,k})_{k=1..m_j}|(L_{j,k})_{k=1..m_j},(\widetilde{E_j}),\widetilde{Q})) = \\ &= \arg(\max_{(L_{j,k})_{k=1..m_j}} \prod_{i,j,k}(P(e_{i,j,k}|\widetilde{E_j},\widetilde{Q},L_{j,k}))).\end{aligned} \tag{2}$$

b) Будем варьировать параметры E и Q заново, вместе с параметрами L:

$$\begin{aligned}&(\widetilde{L_{j,k}})_{k=1..m_j} = \\ &= \arg(\max_{\substack{(L_{j,k})_{k=1..m_j} \\ (E_j),Q}} P((e_{i,j,k})_{k=1..m_j}|(L_{j,k})_{k=1..m_j},(E_j),Q)) = \\ &= \arg(\max_{(L_{j,k})_{k=1..m_j},(E_j),Q} \prod_{i,j,k}(P(e_{i,j,k}|E_j,Q,L_{j,k}))).\end{aligned}$$

2) На первом шаге оценим параметры E, Q, L, решая ту же задачу максимизации, что и в п. 1b):

$$\begin{aligned}&((\widetilde{E_j}),(\widetilde{L_{j,k}})_{k=1..m_j},\widetilde{Q}) = \\ &= \arg(\max_{\substack{(L_{j,k})_{k=1..m_j}, \\ (E_j),Q}} P((e_{i,j,k})_{k=1..m_j}|(L_{j,k})_{k=1..m_j},(E_j),Q)) = \\ &= \arg(\max_{(L_{j,k})_{k=1..m_j},(E_j),Q} \prod_{i,j,k}(P(e_{i,j,k}|E_j,Q,L_{j,k}))).\end{aligned} \tag{3}$$

Далее у нас есть три возможности получить оценку параметров C и P.

a) Зафиксируем полученные на первом шаге оценки параметров L (тогда параметры E, Q нам уже не важны: распределение наблюдаемых величин L зависит только от параметров L):

$$\begin{aligned}&((\widetilde{C_j}),\widetilde{P}) = \\ &= \arg(\max_{(C_j),P} P((\widetilde{L_{j,k}})_{k=1..m_j}|(C_j),P)) = \\ &= \arg(\max_{(C_j),P} \prod_{j,k}(P(\widetilde{L_{j,k}}|C_j,P))).\end{aligned} \tag{4}$$

b) Будем считать, что величины L распределены по своему закону и будем варьировать параметры E, Q (то есть решаем

ту же задачу максимизации, что и на первом шаге в п. 1):

$$
\begin{aligned}
&((\widetilde{C_j}), \widetilde{P}) = \\
&= \arg(\max_{\substack{(C_j),(E_j),\\ P,Q}} P((e_{i,j,k})_{k=1..m_j}|(C_j),(E_j),P,Q)) = \\
&= \arg(\max_{\substack{(C_j),(E_j),\\ P,Q}} \prod_{i,j,k} [\sum_{L_{j,k}} (P(e_{i,j,k}|E_j,Q,L_{j,k})P(L_{j,k}|C_i,P))]).
\end{aligned}
\tag{5}
$$

Заметим, что в сценариях 1b) и 2b) мы в разном порядке решаем одну и ту же пару задач максимизации. Одна из них дает оценки параметров C, E, P, Q, другая - параметров L, E, Q. И единственное отличие итоговых оценок в этих сценариях состоит в том, что оценки параметров E, Q мы берем из разных задач.

c) Будем считать, что величины L распределены по своему закону, а параметры E, Q зафиксируем равными их оценке, полученной на первом шаге:

$$
\begin{aligned}
&((\widetilde{C_j}), \widetilde{P}) = \\
&= \arg(\max_{(C_j),P} P((e_{i,j,k})_{k=1..m_j}|(C_j),(\widetilde{E_j}),P,\widetilde{Q})) = \\
&= \arg(\max_{(C_j),P} \prod_{i,j,k} [\sum_{L_{j,k}} (P(e_{i,j,k}|\widetilde{E_j},\widetilde{Q},L_{j,k})P(L_{j,k}|C_i,P))]).
\end{aligned}
\tag{6}
$$

Задачу поиска вектора значений, на котором функция правдоподобия достигает экстремума, будем решать методом градиентного спуска — итеративной процедурой, стартующей с задаваемого нами значения аргумента и приближающейся к некоторой точке локального максимума. Процедура работает для функций, заданных на непрерывной области изменения аргументов, поэтому будем рассматривать значения каждого аргумента лежащими на отрезке от минимального до максимального значения его дискретной шкалы, а значения функции в нецелых точках доопределим линейно через значения в целых. Тогда функция получится кусочно-линейной. Поскольку функция правдоподобия может иметь несколько локальных максимумов, будем запускать нашу процедуру много раз из различных начальных точек, равномерно распределенных по пространству, которое пробегает вектор аргументов. Для каждого запуска запоминаем итоговое значение вектора аргументов, округленное

до целых чисел, и соответствующее значение функции правдоподобия. Сравнивая эти значения функции правдоподобия, выберем единственное значение вектора аргументов, для которого та максимальна.

Уточнение модели

Теперь решим вопрос о виде распределения уровня качества проекта $L_{j,k}$ и оценки качества проекта $e_{i,j,k}$. Мы помним, что они распределены на одном и том же конечном дискретном множестве. Для определенности зафиксируем, что обе величины принимают целые значения из отрезка $[-3, 3]$.

Далее, нам представляется разумным предположение, что распределение оценки качества проекта $e_{i,j,k}$, выставляемой участником, должно иметь два параметра: среднее, зависящее от уровня качества этого проекта $L_{j,k}$ (а значит, просто равное этому уровню), и дисперсия, зависящая от уровня этого участника как оценщика. То есть если даже самому неумелому оценщику дать много проектов одного уровня на оценивание, то усреднение его оценок будет близко к истинному уровню этих проектов. Но чем более умелый оценщик, тем меньше будут колебаться его оценки для такого пула, какого бы (но одинакового!) уровня идеи в нем ни содержались.

Общепринятое нормальное распределение как раз задается средним и дисперсией, но мы ищем дискретное распределение на конечном отрезке. Поэтому возьмем функцию вероятности (вероятность попадания в различные целые точки интервала $[-3, 3]$) равной плотности нормального распределения в этих точках, поделенной на нормирующую константу. Тогда функция вероятности будет иметь форму плотности нормального распределения - форму колокола.

Осталось задать зависимость дисперсии от величины E_i. Ничто не мешает нам отождествить E_i со средним квадратичным отклонением (СКО — корень из дисперсии) с точностью до умножения на константу. Действительно, мы еще не задали шкалу для E_i. Других требований у нас к ней нет, поэтому пусть эти величины и означают СКО. Но мы не можем просто приравнять их СКО, т. к. они лежат в заранее заданном нами конечном отрезке, а насколько большим может быть СКО, мы не знаем. Поэтому пусть СКО равно qE_i, где q — константа, E_i принимает целые значения на интервале $[0, 3]$. То есть вектор Q в данном случае состоит из одной константы q.

Итак, величина $e_{i,j,k}$ распределена по закону:

$$P(e_{i,j,k} = a|L_{j,k}, E_i) = \frac{e^{-\frac{(a-L_{j,k})^2}{2q^2E_i^2}}}{\sum\limits_{b=-3}^{3} e^{-\frac{(b-L_{j,k})^2}{2q^2E_i^2}}},\ a = -3..3 \tag{7}$$

Что касается распределения уровня качества проекта, введем для него такое же дискретизированное нормальное распределение, у которого среднее зависит от уровня автора как создателя проектов, а СКО равно неизвестной константе p. То есть вектор P в данном случае состоит из одной константы p.

То есть чем выше уровень участника как создателя проектов, тем выше в среднем уровень у создаваемых им проектов. А про зависимость разброса уровня его проектов от его уровня как создателя мы ничего сказать не можем, отчего и полагаем их независимыми.

Поскольку серьезных требований к шкале величины C_j у нас нет, то пусть эта величина совпадает со средним распределения $L_{j,k}$, то есть уровень участника как создателя принимает целые значения из интервала $[-3, 3]$ и означает средний уровень проектов, создаваемых участником.

Итак, величина $L_{j,k}$ распределена по закону:

$$P(L_{j,k} = a|C_j) = \frac{e^{-\frac{(a-C_j)^2}{2p^2}}}{\sum\limits_{b=-3}^{3} e^{-\frac{(b-C_j)^2}{2p^2}}},\ a = -3..3 \tag{8}$$

Связь с другими задачами

Стандартной является задача (см. [4]), которую в терминах данной работы можно сформулировать следующим образом: на основе оценок, данных экспертами (не участниками) проектам, представленным на конкурс, нужно оценить характеристики экспертов как оценщиков (компетентность экспертов) и уровень качества проектов.

Решается эта задача итеративной процедурой, где на каждом шаге измеряется близость вектора оценок разных проектов каждого оценщика к вектору средневзвешенных оценок этих проектов. При расчете средневзвешенной оценки проекта оценки экспертов учитываются с весами, соответствующими их компетентности как оценщиков (изначально эти веса одинаковы для всех экспертов). На основе этой близости перерассчитываются веса экспертов и т. д.

То условие, что в нашей задаче оценки выставляют сами авторы проектов, пока не дает существенного отличия нашей задачи от стандарт-

ной, т. к. мы не устанавливаем никаких связей между характеристиками участника как оценщика и как создателя проектов. Но в дальнейшем установление таких связей представляет интерес, т. к. в реальности эта связь (положительная корреляция компетентностей) существует, а значит, она может помочь оценить параметры модели более точно.

Отличаются же задачи в том, что в стандартной задаче не принимается во внимание автор каждого проекта. В нашей же задаче проекты одного автора будут вероятнее иметь один уровень качества, чем проекты разных авторов.

Выводы

В данной статье предложен новый метод определения компетентности участников конкурса проектов как оценщиков проектов и как создателей проектов и качества предложенных ими проектов на основе оценок, выставленных проектам этими же участниками. Предложены 5 различных вариаций этого метода.

В ближайшее время планируется применить метод на реальных данных и сравнить 5 предложенных вариаций данного метода, а также сравнить предложенный метод с методом, упомянутым в разделе 5 и описанным в [4], в части определения компетентности участников как оценщиков и качества проектов.

Список источников

1. Орлов А. И. Экспертные оценки. — Москва: 2002. — 31 с. (http://www.orlovs.pp.ru.)
2. Лемешко Б. Ю., Денисов В. И., Постовалов С. Н. Прикладная статистика. Правила проверки согласия опытного распределения с теоретическим. Методические рекомендации. Часть I. Критерии типа Хи-квадрат. — Новосибирск: Изд-во НГТУ, 1998. — 126 с. (http://ami.nstu.ru/~headrd/seminar/xi_square/start1.htm.)
3. Орлов А. И. О критериях согласия с параметрическим семейством // Журнал “Заводская лаборатория”. — 1997. — Т. 63, № 5. — С. 49–50. (http://www.orlovs.pp.ru.)
4. Видяпин В. И. Бакалавр экономики. — Т.2. — М.: Триада, 1999. — 696 с. (http://lib.vvsu.ru/books/Bakalavr02.)

Формирование критериев эффективного трудоустройства выпускников ВУЗа на основе методов Data Mining

Ю. И. Ахмайзянова

yuliyanova@yandex.ru

Оренбургский Государственный Университет, Оренбург, Россия

Аннотация. Данная работа посвящена выявлению основных факторов, влияющих на трудоустройство выпускников ВУЗа, путем применения методов Data Mining. Реализованы два подхода к выявлению скрытых закономерностей в данных: 1) подход, решающий проблему обработки качественных данных и 2) подход, основанный на решении проблемы большого объема данных. В ходе исследования сформированы критерии эффективного трудоустройства молодых специалистов.

Ключевые слова: Data Mining; выявление закономерностей; критерии; эффективное трудоустройство выпускников.

Введение

Сегодня одной из серьезных социальных проблем является угроза безработицы для молодых специалистов, окончивших вузы. В настоящее время существует устойчивая тенденция повышенного внимания к проблеме трудоустройства выпускников вузов Российской Федерации. Однако уровень трудоустройства далек от желаемых показателей. По данным Федеральной службы по труду и занятости населения, доля выпускников вузов, оказавшихся безработными, в некоторых регионах достигает 30%. Трудоустройство выпускников ВУЗов является не толь-

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16-18 марта, 2012.

ко проблемой выпускников, затрагивающей и работодателей, но и проблемой самих высших учебных заведений. Каждый ВУЗ является субъектом двух рынков: рынка образовательных услуг и рынка труда специалистов, работа которых тесно взаимосвязана. Поэтому повышение гарантии трудоустройства после получения образования является важным конкурентным преимуществом ВУЗа на рынке образовательных услуг. В этой связи выявление факторов, влияющих на эффективное трудоустройство выпускников, является весьма актуальной задачей, и этой проблеме посвящена настоящая работа. Попробуем выяснить критерии, способствующие отысканию наилучшего пути преодоления порога «учеба - работа» на примере крупнейшего вуза региона – Оренбургского Государственного Университета (ОГУ).

В ОГУ с 2007 года функционирует отдел содействия трудоустройству выпускников и маркетинга образовательных услуг, в функции которого входит:

– изучение спроса на образовательные услуги в регионе;

– сотрудничество с предприятиями и организациями, выступающими в качестве работодателей для студентов и выпускников университета;

– организация временной занятости студентов университета;

– создание банка данных о выпускниках университета, а также банка вакансий, предлагаемых работодателями по соответствующим специальностям;

– сбор, обобщение, анализ и предоставление студентам информации о кадровых предпочтениях и требованиях, предъявляемых к соискателю рабочего места;

– мониторинг рынка труда и анализ трудоустройства выпускников.

Для выполнения поставленных задач Отдел содействия трудоустройству выпускников каждый год проводит анкетирование выпускников. Ежегодно в анкетировании принимают участие более 2000 выпускников ОГУ очной формы обучения. Анкеты содержат следующий ряд вопросов:

- средний балл выпускника;
- наличие дополнительного образования;
- изучаемый язык;
- уровень владения персональным компьютером (ПК);
- форма обучения в ВУЗе;
- нравится ли полученная специальность;
- как вы оцениваете уровень образования;
- подработка во внеучебное время;
- ожидаемый минимальный уровень заработной платы;

- как вы оцениваете перспективы своего трудоустройства и пр.

Традиционно для установления приоритетности выбранных критериев применяют различные статистические методы. Но в нашем случае, так как ответы на вопросы являются качественными переменными, обычные статистические методы неприменимы. Это и представляет весьма трудную задачу для обработки данных. Поэтому для выделения наиболее значимых факторов успешного трудоустройства, решаем проблему, используя метод обучения с учителем, а именно, алгоритм индукции решающих деревьев ID3. В ходе исследования реализовано программное средство, реализующее ID3, входными данными для которого как раз и выступили анкеты выпускников ОГУ (формат Excel) [1].

Алгоритм ID3

Основная идея ID3 (разработан Джоном Р. Куинланом) при построении дерева заключается в том, что в каждом узле среди ещё неиспользованных атрибутов определяется тот, который наиболее информативен для классификации образцов, соответствующих пути от корня дерева до этого узла. Наиболее информативный атрибут сопоставляется с этим узлом, и далее такой подход рекурсивно повторяется в каждом узле. Если в какой-то момент все атрибуты оказываются использованными, то создаётся лист, которому присваивается значение наиболее часто встречающейся среди оставшихся образцов категории. Информативность атрибута оценивается в ID3 функцией прироста (gain), которая использует теоретико-информативное понятие энтропии.

Алгоритм ID3 основан на процедуре рекурсивного характера:

1. Выбирается атрибут для корневого узла дерева, и формируются ветви для каждого из возможных значений этого атрибута.
2. Дерево используется для классификации обучающего множества. Если все примеры на некотором листе принадлежит одному классу, то этот лист помечается именем этого класса.
3. Если все листья помечены именами классов, алгоритм заканчивает работу, в противном случае узел помечается именем очередного атрибута и создаются ветви для каждого из возможных значений данного атрибута, после чего переход к шагу 2. [2]

Исходные данные для исследования - анкеты выпускников ОГУ 2009, 2010 и 2011 года выпуска. Общее число респондентов составило 2187, 2133 и 2215 человека соответственно, то есть общий объем выборки для анализа равен 6535. Мощность обучающего множества также

равна 6535. Множество предсказывающих атрибутов состоит из следующих факторов: «средний балл», «дополнительное образование», «изучаемый язык», «уровень владения ПК», «форма обучения в ВУЗе», «нравится ли полученная специальность», «уровень образования, полученный в ОГУ», «подработка во внеучебное время», «минимальный уровень заработной платы». Целевой атрибут – «как вы оцениваете перспективы своего трудоустройства после окончания ВУЗа». Этот атрибут может принимать значения «вероятно» и «менее вероятно». Задача состоит в том, чтобы, применив алгоритм ID3, предсказать вероятность трудоустройства, то есть построить решающие деревья для исходных данных, классифицировав имеющиеся атрибуты в порядке убывания значимости, тем самым, выявив наиболее важные факторы, влияющие на трудоустройство.

На первом шаге вычисляем для каждого атрибута информативность согласно алгоритму и находим среди них максимальную. Помечаем корневую вершину этим атрибутом и его значением информативности. Далее осуществляем такой подход рекурсивно для неиспользованных атрибутов в соответствии с вариантами уже помеченных атрибутов. При этом алгоритм выбирает атрибуты, дающие наибольшую информацию.

Таким образом, применив алгоритм ID3, построены три решающих дерева: согласно данным анкет 2009 года (см. рисунок 1), 2010 года и 2011 года. Причем, числа в вершинах деревьев показывают информативность признака данной вершины; пустые листья показывают, что значения, предписанные веткам пути от вершины дерева к данному пустому листу, не позволяют предсказать трудоустроится выпускник или нет (здесь возможно рассмотрение большего количества факторов, влияющих на трудоустройство).

Анализируя полученные деревья, можно сформировать перечень основных факторов, влияющих на перспективы трудоустройства.

На протяжении всех трех лет основным фактором, влияющим на трудоустройство, является «подработка во внеучебное время». Также немаловажным является «ожидаемый уровень з/п» выпускника. В 2009 г. к более приоритетным критериям относились «дополнительное образование» и «средний балл», к 2011 г. повышается требование к владению «иностранными языками». Помимо этого важным становится «оценка уровня образования» выпускника.

Алгоритм извлечения продукционных правил из большой базы данных

При решении проблемы трудоустройства, сталкиваемся с большим размером базы данных (за 3 года – это более 6,5 тысяч записей). «+» состоит в том, что больше шансов среди большого количества атрибу-

тов отыскать подходящее описание классов, а «-» - увеличение размеров пространства поиска, что может привести к комбинаторному взрыву. Основное отличие извлечения знаний из баз данных от традиционных методов машинного обучения – использование базы данных в качестве обучающего множества.

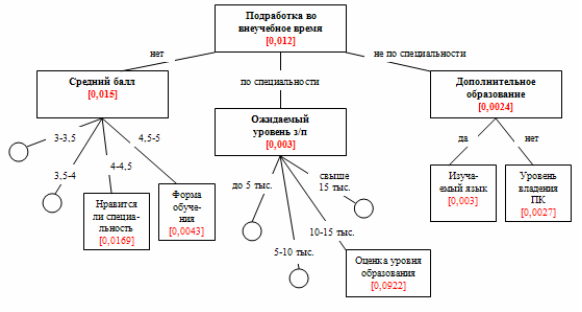


Рисунок 1 Дерево решений, построенное при помощи алгоритма ID3 на основе анкет 2009 года

Для этих целей используем алгоритм извлечения продукционных правил из большой базы данных (АИПП), предложенный Холшеймером и Керсеном [3]. Алгоритм должен найти правила, которые определяют принадлежность объекта к некоторому классу, задаваемому пользователем (в нашем случае, возможность трудоустройства). Пользователь проводит разделение базы данных (БД) на две части: первая содержит объекты, принадлежащие классу, и составляющие тем самым множество положительных примеров (вероятно трудоустройство), в то время как вторая содержит все остальные, образующие множество отрицательных примеров (маловероятно трудоустройство). Правило считается корректным, если оно покрывает все положительные примеры и не покрывает ни одного отрицательного. Целью алгоритма является извлечение правил с наибольшим показателем качества.

Для проведения исследования имеем исходные данные - выборку базы данных большого размера, представляющую собой ответы выпускников 2009, 2010 и 2011 года выпуска на вопросы анкеты, что составляет в целом более 6,5 тысяч записей.

Целью данного алгоритма является нахождение правил, определяющих принадлежность объекта к классу «Трудоустройство» = *вероятно*.

Работа начинается с разделения обучающей выборки на две части: первая образует множество положительных примеров и содержит объекты, для которых «Трудоустройство» = *вероятно*, а вторая образует множество отрицательных примеров и содержит объекты с другим зна-

чением атрибута, то есть «Трудоустройство» = *маловероятно*. Следующий шаг состоит в построении «тривиального» правила, приписывающего все объекты к классу «Трудоустройство» = *вероятно*. Оценка качества этого правила есть вероятность того, что произвольно выбранный объект базы данных принадлежит этому классу. Следующим шагом является расширение правила за счет добавления в его часть условия <атрибут> = <значение>, что приводит к новому правилу и так далее. При этом алгоритм выбирает расширения с наиболее высокой оценкой качества.

На выходе данный алгоритм дает некую совокупность продукционных правил. Например, для данных за 2009 год, она следующая:

если («наличие подработки» = *по специальности*) & («оценка уровня образования» = *отлично*), **то** («трудоустройство» = *вероятно*) в 73,5% случаев.

если («наличие подработки» = *по специальности*) &(«средний балл» = *4,5 - 5*), **то** («трудоустройство» = *вероятно*) в 73,5% случаев.

если («наличие подработки» = *по специальности*) &(«владение иностранными языками» = *английский*), **то** («трудоустройство» = *вероятно*) в 73,4% случаев.

если («уровень владения ПК» = *профессиональный*) &(«средний балл» = *3,5 - 4*), **то** («трудоустройство» = *вероятно*) в 77,7% случаев.

если («уровень владения ПК» = *профессиональный*) &(«наличие подработки» = *по специальности*), **то** («трудоустройство» = *вероятно*) в 73,3% случаев.

если («уровень владения ПК» = *профессиональный*) &(«оценка уровня образования» = *хорошо*), **то** («трудоустройство» = *вероятно*) в 72,1% случаев.

если («наличие дополнительного образования» = да) & («средний балл» = *3,5 - 4*), **то** («трудоустройство» = *вероятно*) в 74,5% случаев.

если («наличие дополнительного образования» = да) & («наличие подработки» = *по специальности*), **то** («трудоустройство» = *вероятно*) в 72,8% случаев.

если («наличие дополнительного образования» = да) & («оценка уровня образования» = *отлично*), **то** («трудоустройство» = *вероятно*) в 70,8% случаев.

Так как такая форма представления является громоздкой, то для демонстрации результатов остановимся на построении решающих деревьев.

Итак, применив алгоритм извлечения продукционных правил из большой базы данных, построены три решающих дерева: согласно данным анкет 2009 года (см. рисунок 2), 2010 года и 2011 года. Во всех деревьях корнем является значение целевого атрибута «Трудоустройство» = *вероятно* и соответствующая ему оценка качества. Вершины дерева

формируют атрибуты с наивысшей оценкой качества, она указана в квадратных скобках. Таким образом, чем выше оценка, тем больше вероятность трудоустройства выпускника. Число веток в каждом ярусе дерева равно 3, а глубина поиска, то есть число таких ярусов, равно 2.

Анализируя полученные деревья, можно сделать следующие выводы. Как и в 2009 году, в 2010 более приоритетные факторы «подработка по специальности», «профессиональный уровень владения ПК» и «наличие дополнительного образования». Наряду с этим повысились требования к «знанию английского языка». В 2011 году принимает важное значение атрибут «оценка уровня образования», помимо «подработки по специальности» и «профессионального уровня владения ПК». С отличной оценкой трудоустроены порядка 70% выпускников. Более значимым становится «средний балл». 70% выпускников трудоустроены со средним баллом выше среднего (выше 4,25).

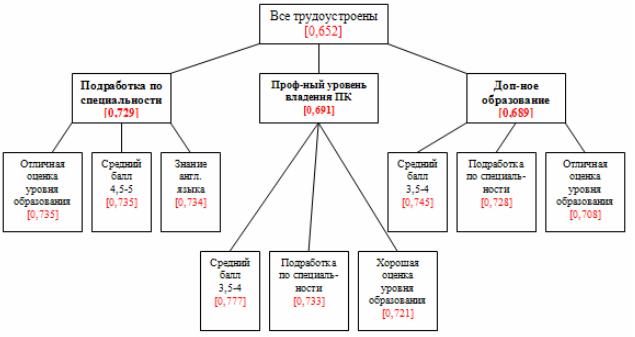


Рисунок 2. Дерево решений, построенное на основе данных анкет 2009г., применяя алгоритм извлечения продукционных правил

Сравнительный анализ алгоритма ID3 и алгоритма извлечения продукционных правил из большой базы данных

Алгоритм ID3 строит дерево решения, а алгоритм извлечения продукционных правил из большой БД - систему продукционных правил. Продукционные правила являются более удобными и гибкими, а дерево решений – более наглядным. Однако дерево решения может быть легко сведено к системе продукционных правил, и наоборот. В этом вопросе можно считать алгоритмы схожими.

Существенным моментом является длина полученных описаний. Дело в том, что алгоритм ID3 включает в полученное описание все доступные атрибуты. Алгоритм извлечения продукционных правил - на-

против, строит дерево минимальной длины (длина, т е глубина задается пользователем. В худшем случае она будет такой же, как в ID3). Выбор оптимальной длины описания представляет проблему. С одной стороны, небольшие описания легче воспринимаются человеком, в них быстрее осуществляется поиск. С другой стороны, минимизируя описание, мы можем отбросить атрибуты, в общем случае влияющие на классификацию.

Алгоритм ID3 имеет возможность обработки атрибутов, имеющих непрерывные области определения. Все значения таких атрибутов разбиваются на два диапазона, при этом выбирается порог, дающий наибольший прирост информации. Это позволяет построить всего две ветви для поддерева на данном этапе. В отличие от ID3, алгоритм извлечения продукционных правил из большой БД не имеет механизмов обработки непрерывных атрибутов, но можно расширить область применения последнего алгоритма, предварительно заменив непрерывный атрибут дискретным, разбив весь диапазон непрерывных значений на 2-5 отрезков.

Вычислительная сложность алгоритма извлечения продукционных правил из большой БД равна $O(k\ N^3)$. Вычислительная сложность алгоритма ID3 равна $O(k^2\ N^4)$, так как используется дополнительный цикл для определения ширины луча, т. е. количества веток при переходе к нижележащим листьям (число лучей определяется числом вариантов каждого атрибута, в то время как в предыдущем алгоритме мы сами задаем количество лучей). Таким образом, на больших обучающих множествах алгоритма извлечения продукционных правил будет выдавать результаты существенно быстрее, чем ID3.

АИПП является более гибким, т. к. позволяет выделить значимые атрибуты в конкретном интересующем классе, в то время как ID3 работает со всей выборкой.

АИПП дает более точные результаты при работе с большой выборкой.

Таким образом, на основании проведенного сравнительного анализа более предпочтительным для обработки имеющихся данных является алгоритм извлечения продукционных правил из большой базы данных.

Выводы

В ходе исследования, в котором респондентами выступили выпускники 2009, 2010 и 2011 года выпуска Оренбургского Государственного Университета, выделены основные факторы, влияющие на перспективы трудоустройства. Для установления приоритетности выбранных критериев использованы два подхода к решению стоящих задач: 1) подход, основанный на использовании алгоритма ID3, применяемый к решению

проблемы о наличии качественных переменных в выборке; 2) подход, использующий алгоритм извлечения продукционных правил из большой базы данных с целью решения проблемы объемной выборки.

В результате применения алгоритма ID3 выделены следующие значимые факторы при трудоустройстве: «подработка во внеучебное время», «ожидаемый уровень з/п» и «изучаемый язык», «наличие дополнительного образования». Алгоритм извлечения продукционных правил ранжировал атрибуты следующим образом: наличие «подработки во внеучебное время», «профессиональный уровень владения ПК» , наличие «дополнительного образования» и владение «иностранным языком». Отсюда можно сделать вывод, что наиболее приоритетными являются «подработка во внеучебное время», «дополнительное образование», «изучаемый язык» и «уровень владения ПК».

Таким образом, на основе проведенного исследования можно утверждать, что рынок труда выпускников динамичен и очень сложно предугадать как он станет развиваться в будущем. Но всё же можно предположить, что с течением времени важными при трудоустройстве останутся приведенные выше факторы. Для более точных результатов необходимо ежегодное исследование данной проблемы, что возможно позволит найти закономерность в выделении важнейших факторов, влияющих на перспективы трудоустройства. Также для уточнения приоритетности возможно использование алгоритма для исследования зашумленных данных, так как вопросы анкеты являются субъективными и при исследовании некоторые ответы объединены, что собой и представляет некий шум в выборке.

Список источников

1. Аналитические отчеты по данным маркетингового исследования «Выпускник ОГУ – 2009», «Выпускник ОГУ – 2010», «Выпускник ОГУ – 2011»

2. Паклин Н., Орешков В. Бизнес – аналитика: от данных к знаниям / Учебное пособие, 2-е издание, 2010

3. Вагин В.Н., Головина Е.Ю. Достоверный и правдоподобный вывод в интеллектуальных системах / ФИЗМАТЛИТ, 2008

Автоматизированный анализ мнений о товарах

Ермаков Сергей

Пермский государственный научный исследовательский университет

Аннотация. Проект направлен на автоматизированный анализ отзывов и комментариев из различных источников, в том числе из поисковой выдачи, и преобразование их в обобщенные категоризированные оценки на основе применения технологии Sentiment Analysis.

Ключевые слова: opinion mining, sentiment analysis, анализ мнений

Введение

В процессе выбора, многие руководствуется не описанием товара, а отзывами, написанными «независимыми экспертами».Однако чтобы получить полноценную информацию, нужно обработать большой объем информации. Мы покажем, каким образом это можно автоматизировать.

Сбор данных

Для того чтобы получить общую оценку какого-либо товара, необходимо получить множество текстов из различных источников:

- интернет-магазины
- форумы, порталы
- Yandex Market
- сайты с обзорами

Перечисленные выше ресурсы представляют собой заранее (до анализа) заданный пополняемый (вручную) список сайтов. Однако пере-

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16-18 марта, 2012.

числить все нужные ресурсы вручную невозможно, поэтому необходимо так же учитывать данные поисковой выдачи по данному товару. В качестве поискового сервера планируется (если позволит лицензионное соглашение) использовать Yandex.XML[1].

Сложность получения нужных текстов в том, что очень мало ресурсов предоставляют API для доступа к своим данным. Таким образом, необходимо анализировать HTML и выделять в нем те тексты, которые являются отзывами или комментариями. Данная задача является трудоемкой – необходим анализ текстов, а так же система правил, позволяющая выделить нужные части, плюс способность системы обучаться на основе Machine Learning. Для решения этой задачи планируется использовать систему Weka [2].

Ранжирование информации

На первом этапе мы получили набор текстов, которые необходимо проанализировать.

В большинстве интернет текстов нарушена орфография, поэтому перед непосредственным анализом нужно нормализовать орфографию текстов.

На основе метрики tf\idf и расширения запроса с помощью лингвистической онтологии (например, WordNet) определяются релевантные тексты.

Каждый текст разбивается на набор предложений – будем считать, что предложения содержат законченную мысль, которая характеризует товар с какой-нибудь стороны. Каждое предложение затем анализируется отдельно, но при этом должен учитываться локальный контекст[3], потому что он может содержать анафорические связи, а так же семантически связанные слова. Для анализа проводится лемматизация с помощью словаря Зализняка[4]. Предложению ставится в соответствие ранг 0-5, оценивающий его полезность, в соответствии с некоторыми эмпирическими признаками:

- объем текста
- наличие слов с высоким индексом tf\idf
- количество знаков восклицания
- длина предложения
- положение предложения

На данном этапе мы получаем ранжированные предложения из отзывов. Это значительно упрощает работу пользователя, уменьшая объем информации, который ему необходимо обработать.

Sentiment Analysis

Результат предыдущий этапа хотя и уменьшает объем информации, но все равно требует времени для чтения текста. Для того чтобы получить оценку самого товара в виде числа (или количества звезд), необходимо проанализировать каждое предложение – содержит оно нейтральную информацию, негативную или позитивную. Для этого используется технология Sentiment Analysis[5]. Пока на базовом этапе планируется простой анализ на основе списка эмоционально окрашенных слов. В дальнейшем для оценки окраски предложения потребуется использовать более сложные методики, с возможностью машинного обучения.

Получив эмоциональную оценку каждого предложения, мы можем вычислить суммарную оценку товара следующим образом: эмоциональная оценка дискретна: {-1, 0, 1},

$P = (p_1, p_2, \dots, p_n)$ набор весов, $E = (e_1, e_2, \dots, e_n)$ набор sentiment-оценок. Суммарная оценка получается как $S = \sum_{i=1}^{n} p_i * e_i$.

С помощью полученных суммарных оценок можно быстро сравнить много товаров между собой.

Для английского языка существуют системы, которые позволяют выполнить анализ эмоциональной окраски (TrustYou[6], TwitterSentiment[6], IDOL[7], AlchemyAPI[8], и др.), для русского языка подобные сервисы пока не развиты, поэтому оценку эмоциональной окраски необходимо реализовывать самостоятельно.

Категоризированная оценка

Зачастую, при выборе товара нам неважно насколько он хорош в целом, нам важны отдельные характеристики, поэтому общая оценка не дает нужной информации. Для того чтобы ее получить, нужно кластеризовать набор предложений в различные категории. Для того чтобы выделить категории необходим синтаксический анализ предложения с целью выявления объекта или предмета высказывания. Выделение категории можно проводить на основе шаблонов[9], по возможности используя технологию бутстрепа (вручную разметив базовые варианты, затем автоматически расширяя набор шаблонов за счет выявления новых вариантов на основе базовых). Таким образом, мы сможем для каждого предложения получить ту сторону или несколько сторон товара, которая оценена. Кластеризация на примере предметной области отзывов об автомобилях представлена в системе Pulse[10].

Для реализации полноценной кластеризации, необходимо решить несколько проблем анализа текста, таких как разрешение анафоры, снятие синонимии. Кроме того, для выделения нескольких понятий в одну категорию, необходима онтология. Так как мы пока не специфицируем

область, к которой принадлежит товар (автомобили, компьютерная техника, и т.д.), онтология должна быть общей – для этих целей может быть использован WordNet или другая онтология общего назначения.

Существует так же проблема эффективности проведения анализа эмоциональной окраски для произвольной предметной области: в различных сферах те или иные термины могут иметь различное значение: например, большой размер для монитора – это плюс, а для портативных устройств – минус. Общий классификатор всегда будет иметь меньшую эффективность, чем специализированный. Существует несколько вариантов решений данной проблемы:

1. Обучать классификатор на всех доступных наборах данных сразу – самый очевидный вариант. Данный метод показывает результаты хуже, чем классификатор для отдельного домена, он используется в качестве основы для других методов.
2. Разграничивать использование признаков для разных доменов. Другими словами, для каждого домена создается специализированный словарь. Таким образом, мы исключаем специфические высказывания для данной предметной области, но оставляем общие для всех областей эмоционально окрашенные тексты.
3. Использовать наборы классификаторов: разные классификаторы можно объединять в наборы [11]. При классификации в этом случае каждый из классификаторов участвует в итоговом решении с некоторым весом. Существуют различные варианты использования и обучения данных наборов, в том числе использование мета-классификатора [12] (который калибрует веса составляющих его классификаторов)

Использование

Представляемый проект будет онлайн-сервисом, который ориентирован на рядовых потребителей, их цель - купить максимально подходящий им продукт. Во многих случаях при выборе товара пользователи опираются на мнения других людей. Объективность оценки во многом связана с репрезентативностью выборки, которую вручную получать очень долго.

Рассматривается реализация API для коммерческого массового автоматизированного доступа к сервису для магазинов и производителей. Магазины заинтересованы в покупке наиболее продаваемых товаров, а производители смогут оценить отклик на их продукцию без проведения дорогостоящего анкетирования. Возможно, будет определенная сложность с коммерческим использованием системы, т.к. в рамках нее применяются несколько других систем, со своими лицензиями.

Выводы

Существует очень мало сервисов анализа эмоциальной окраски текста реализовано для русского языка

Для получения качественного результата до того, как выполнять Sentiment Analysis, необходимо решать много трудоемких задач

Необходима реализация эффективного кросс-доменного анализа эмоциональной окраски, что является не решенной на данный момент проблемой даже для английского языка

Для решения главной и сопуствующих задач необходимы средства работы с русским языком, которые сейчас намного более ограничены, нежели средства для работы с английским языком

Список источников

1. Информация доступна на сайте http://xml.yandex.ru/

2. Информация доступна на сайте http://www.cs.waikato.ac.nz/ml/weka/

3. Ermakova L., Mothe J., IRIT at INEX: Question answering task, INEX 2011 Workshop Pre-proceedings, 2011

4. Информация доступна на сайте http://zaliznyak-dict.narod.ru/index.htm

5. B. Pang. L. Lee, Opinion Mining and Sentiment Analysis, Foundations and Trends® in Information Retrieval, pp. 1-135, 2008

6. Информация доступна на сайте http://www.trustyou.com/

7. Информация доступна на сайте http://www.autonomy.com/content/Functionality/idol-functionality-sentiment/index.en.html /

8. Информация доступна на сайте http://www.alchemyapi.com/

9. Peter D. Turney, Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews, Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 417-424, 2002.

10. M. Gamon, A. Aue, S. Corston-Oliver, E. Ringger, Pulse: Mining customer opinions from free text, Advances in Intelligent Data Analysis VI, pp. 121–132, 2005

11. T. Dietterich, Macine learning research: Four current directions, AI Magazine, vol. 18, no. 4, pp. 97-136, 1997.

12. L. Todorovski and S. Dzeroski, Combining classifiers with meta decision trees, Machine Learning, vol. 50, no. 3, p. 223–249, 2003.

Географическая информационная система «Поездка на один бензобак»

Н.К. Габдрахманов, Е.Е. Михеева, М.В. Рожко

nz9nz@rambler.ru

ФГАОУВПО «Казанский (Приволжский) федеральный университет, Институт Экологии и Географии, кафедра социально-культурного сервиса и туризма, Казань, Республика Татарстан, Россия

Аннотация. Статья посвящена вопросу популяризации внутреннего туризма. Одним из решений данной задачи выступает географическая информационная система туристской направленности. Данная ГИС представлена как система сбора, хранения, анализа и графической визуализации пространственных данных и связанной с ними информации о туристских объектах, и позволяющая пользователям искать, анализировать и редактировать цифровые карты, а также дополнительную информацию об объектах. Приведен пример одного из туристских маршрутов, полученного с помощью данной геоинформационной системы.

Ключевые слова: географическая информационная система, внутренний туризм.

Введение

Туризм является одной из важнейших сфер деятельности современной экономики, нацеленной на удовлетворение потребностей людей и повышение качества жизни населения. При этом в отличие от многих других отраслей экономики туризм не приводит к истощению природных ресурсов. Будучи экспортноориентированной сферой, туризм про-

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16-18 марта, 2012.

является большую стабильность по сравнению с другими отраслями в условиях неустойчивой ситуации на мировых рынках.

Применение геоинформационных технологий является огромным подспорьем в деле организации и проведения туров, а также сопутствующего сервиса. На сегодняшний день Российский туристический бизнес не может похвастаться огромными успехами в области внедрения геоинформационных технологий, наибольшего успеха достигли центральные города: Москва и Санкт-Петербург.

В настоящее время Россия вызывает малый интерес у иностранных граждан с точки зрения туристической привлекательности. Статистические данные показывают, что происходит сокращение доли иностранных граждан, прибывающих на территорию РФ с туристическими целями в общем количестве иностранцев прибывающих в Россию. Так, в 2010 году нашу страну посетило 2133869 человек с целью туризма (9,6 % от общего числа въехавших иностранных граждан) (Рис.1).

Рис.1. Въезд иностранных граждан.

Что касается граждан РФ, доля лиц, посещающих иностранные государства с туристическими целями, увеличивается. В 2010 году 12605053 выехало за пределы РФ с целью туризма, что составляет 32 % от общего числа выехавших граждан (Рис.2).

Поэтому, исходя из приведенных данных, мы приходим к выводу о необходимости развития различных направлений внутреннего туризма на территории РФ в целом и Республики Татарстан в частности.

Одним из методов популяризации внутреннего туризма является тур выходного дня. Туры выходного дня становятся все более популярными среди туристов. Туры выходного дня были разработаны для

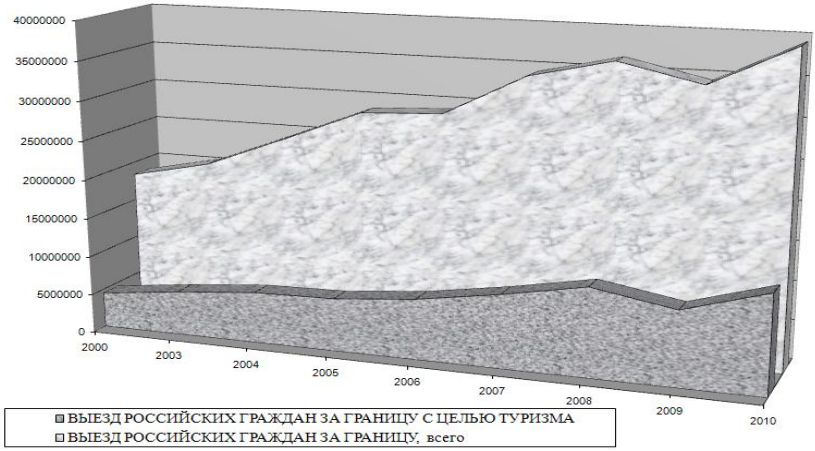


Рис.2. Выезд граждан Российской Федерации.

людей, которые не могут посвятить отдыху неделю или две, слишком плотный рабочий график которых, не позволяет этого сделать. При помощи подобных туров создается возможность разнообразить привычные выходные, сделать их более оригинальными и запоминающимися.

Довольно популярными турами являются автомобильные туры выходного дня, с помощью которых можно фактически в течение дня посетить достопримечательности желаемого города и затем снова вернуться к своим делам.

Что же геоинформационные технологии могут сделать для туризма? Пожалуй, главным их козырем является наиболее «естественное» представление как собственно пространственной информации, так и любой другой информации, имеющей отношение к объектам, расположенным в пространстве. Таким образом, геоинформационные системы (ГИС) могут помочь везде, где используется пространственная информация и информация об объектах, находящихся в определенных местах пространства. ГИС помогает сократить время получения ответов на запросы клиентов; выявлять территории подходящие для требуемых мероприятий; выявлять взаимосвязи между различными параметрами и т.д.
Проведя ряд исследований, мы пришли к выводу о целесообразности развития проекта геоинформационной системы «Поездка на один бензобак», суть которого заключается в том, что создается информационная система, которая позволяет получать информацию, производить поиск маршрутов в радиусе «транспортной доступности» (Рис.3), определять ориентировочную стоимость тура, а также получать краткую справку о культурно-исторических объектах.

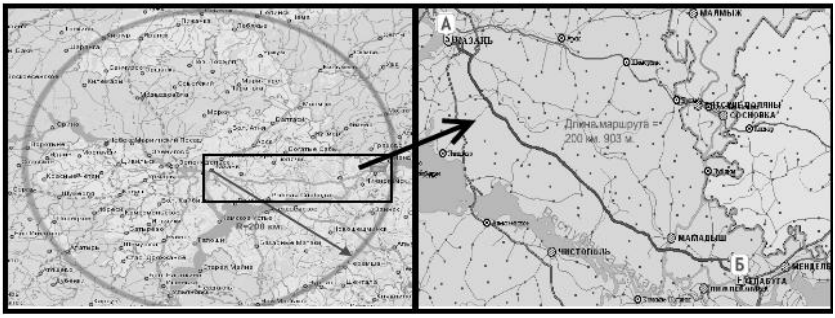


Рис.3. Радиус «транспортной доступности одного бензобака»

Приведем конкретный пример: 5 человек, 1 автомобиль, емкость бака которого составляет 40 л, расход топлива 10л/100 км. Маршрут выходного дня: Казань – Елабуга – Казань. Длина маршрута - 400 км, длительность - 2 дня. Начало – Суббота, окончание – Воскресенье.

Стоимость тура выходного дня складывается из следующих основных положений: Проезд, проживание, питание, а также стоимость экскурсионного обслуживания и посещения музеев:

Проезд. Вместительность стандартного бензобака – 40 литров. Стоимость 1 литра бензина 26,50 рублей. Общая стоимость проезда будет составлять 1060 руб. При максимальной вместительности автомобиля – 5 пассажиров, стоимость проезда на 1 пассажира будет составлять 212 рублей.

Проживание. 1 ночь в гостинице Елабуги. Стоимость проживания в Двухместном номере Отеля Alabuga City Hotel - 3250 руб. на двоих, стоимость проживания в Двухместном номере Гостиницы «Горка» 1800 на двоих, стоимость проживания в Двухместном номере Гостиницы «Тойма» 1500 на двоих. Для расчетов возьмем среднюю стоимость проживания на одного человека, которая составляет 800 рублей.

Питание: Средняя стоимость обеда на предприятиях общественного питания Елабуги составляет 170 руб. С учетом того, что завтраки включены в стоимость проживания, туристам необходимо пообедать 3 раза. Общая стоимость питания в таком случае будет составлять 510 руб.

Посещение музеев: В Елабуге 8 музеев, стоимость посещения которых колеблется от 50 до 100 руб. Стоимость посещения всех музеев Елабуги будет составлять 450 руб.

В итоге общая стоимость тура составляет 1972 рубля на 1 человека (Табл.1). Следует отметить, что данная стоимость не является окончательной и зависит от интересов, потребностей и финансовых возможностей туристов.

Табл.1 Пример расчета стоимости тура выходного дня

Позиция	Стоимость на 1 человека (руб.)
Проезд	212
Проживание	800
Питание	510
Посещение музеев	450
ИТОГО	1972

Выводы

Данный инновационный проект облегчит поиск информации по туристическим маршрутам и дестинациям, позволит не только модернизировать и оптимизировать существующие туристские маршруты, но и создать новые. Информационная система даст несомненный толчок в развитии внутреннего туризма в регионе, даст возможность привлечения дополнительных инвестиций в экономику малых городов и населенных пунктов республики. Немалое значение данный проект имеет в преддверии проведения Универсиады 2013 года, что позволит гостям республики ознакомиться с достопримечательностями не только Казани и ее окрестностей, но и соседних регионов.

В итоге мы приходим к тому, что сегодня геоинформационные технологии постепенно завоевывают Российский рынок. Создание земельного кадастра позволит на основе его карт строить другие, предметно ориентированные карты и дополнять их соответствующим атрибутивным наполнением. Для создания геоинформационной системы «поездка на один бензобак» потребуется объединение усилий всех заинтересованных сторон, это необходимо для создания информационного контента баз данных, постоянного поддержания его актуальности и соответствия действительности. Также необходима финансовая и законодательная поддержка со стороны государства.

Прототипы системы стереонаблюдения

В. Горшенин

Gorshenin.Vladimir@gmail.com

Челябинский государственный университет, кафедра КБиПА, Челябинск, Россия

Аннотация. В статье излагается задача стереонаблюдения, общие требования к системе стереонаблюдения. Приведены недостатки, обнаруженные в процессе эксплуатации первого созданного прототипа системы стереонаблюдения. Предлагается модификация прототипа с помощью алгоритма кодовой книги.

Ключевые слова: стереозрение; видеонаблюдение; алгоритм кодовой книги.

Введение

Термин «стереонаблюдение», используемый в рамках данной работы, сформулирован по аналогии с «видеонаблюдением» и означает применение систем стереозрения в задачах наблюдения [1] [2]. Стереонаблюдение качественно отличается от видеонаблюдения тем, что позволяет не только выполнять видеорегистрацию событий в рамках наблюдаемой сцены, но и определять трехмерные координаты тех участков сцены, в которых происходят события (движение). Переход на использование систем стереонаблюдения позволит автоматически классифицировать наблюдаемые события на «разрешенные» и «запрещенные», а также реагировать на появление определенным образом (видеорегистрация, отправка сигнала оператору и т. п.). Выбор метода стереозрения среди других методов технического зрения для решения поставленной задачи объясняется следующим обстоятельством. Системы стерео- и видеонаблюдения используют одинаковую аппаратную базу и поэтому существующие

Игнатов Д. И., Яворский Р. Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16–18 марта, 2012

системы видеонаблюдения могут быть улучшены установкой дополнительных камер и вычислительных модулей со встроенными алгоритмами стереозрения. Другие методы технического зрения требуют для своей работы дополнительной лазерной, инфракрасной, ультразвуковой и т. п. «подсветки», поэтому их применение серьезно усложнит существующие системы наблюдения.

Данная статья публикует промежуточные результаты разработки программно-аппаратного комплекса системы стереонаблюдения с функций автоматической классификации движения в рамках наблюдаемой сцены:

— общая схема функционирования системы стереонаблюдения;
— реализация прототипа системы стереонаблюдения.

Система стереонаблюдения

Для реализации основной функции система стереонаблюдения должна выполнять вспомогательные функции:

1) калибровка стереокамеры;
2) выделение движения;
3) выполнение сопоставления характерных точек для движущегося объекта;
4) восстановление трехмерных координат наблюдаемого объекта;
5) применение правил реагирования.

Основной идеей предложенного автором работы прототипа [4] является переход от отслеживания характерных точек к отслеживанию контуров перемещающегося объекта. Для выделения контуров движущегося объекта последовательно были реализованы и опробованы три способа:

— разность двух последовательных кадров;
— применение статистической модели;
— алгоритм кодовой книги (codebook).

Изначально был реализован первый подход как наиболее простой, тем не менее результаты его работы оказались неприемлемыми. С целью улучшения была применена статистическая модель наблюдаемой сцены, которая улучшила общее качество работы, но не решила отдельные проблемы. В качестве итогового способа используется алгоритм кодовой книги, изначально разработанный для сжатия видеоданных и адаптированный к задаче определения движения.

Алгоритм кодовой книги

В данном способе также создается модель наблюдаемой сцены. С каждым пикселем изображения соотносится набор допустимых интервалов яркости — кодовая книга. Формирование кодовой книги для каждого пикселя выполняется в течение предварительного периода времени согласно следующему алгоритму [3]. Если текущее значение яркости пикселя точно попадает в один из существующих интервалов, то продолжить далее. Если текущее значение яркости пикселя попадает на границу интервала с учетом определенного допуска, то интервал необходимо увеличить в сторону расширения данной границы интервала. Если предыдущие условия не выполнены, то необходимо сформировать новый интервал, «накрывающий» текущее значение яркости пикселя. Каждому интервалу назначается атрибут — время последнего изменения, это позволяет отбрасывать устаревающие записи и поддерживать модель в актуальном состоянии. После того как для каждого пикселя изображения сформирована кодовая книга система переходит в режим функционирования. Если текущее значение яркости пикселя попадает в один из интервалов кодой книги, то считается, что это элемент заднего фона. В противном случае, пиксель помечается как элемент движущегося объекта. Таким образом формируется двухцветная маска, помечающая движущиеся объекты. При периодическом обновлении кодовой книги во время работы алгоритма она приспосабливается к плавным изменениям наблюдаемой сцены. Применение данного метода требует большего объема памяти ЭВМ и вычислительных ресурсов процессора. Тем не менее, метод кодовой книги дает наилучшие результаты по выделению движущихся объектов.

В дополнении к методу кодовой книги в модифицированном прототипе системы стереонаблюдения используется обработка полученной маски с целью выделения компонент связности. Каждая выделенная компонента связности признается отдельным движущимся объектом. Для нее вычисляется точка центра масс — именно она используется для выполнения последующей триангуляции и восстановления трехмерных координат движущегося объекта.

Список источников

1. Горшенин В. В. Модифицированный алгоритм Лукаса—Канады в задаче стереонаблюдения // Казанская наука. — 2010. — № 5. — С. 4–7.
2. Горшенин В. В. Перспективы развития систем видеонаблюдения // Казанская наука. — 2010. — № 6. — С. 7–10.

3. Kim K., Chalidabhongse T. H., Hardwood D., Davis L. Real-time foreground-background segmentation using codebook model // Real-time Imaging. — 2005. — Т. 11, № 3. — С. 172–185.
4. Горшенин В. В. Прототип системы стереонаблюдения // Современные проблемы математики: тезисы 42-й Всероссийской молодежной школы-конференции, Город: Изд-во, 2011. — С. 283–286.

Оценивание параметров билинейных динамических систем с помехой в выходном сигнале

Д. В. Иванов[1], О. В. Усков[2]

[1]dvi85@list.ru, [2]Quentyn@mail.ru

СамГУПС, Самара, Россия

Аннотация. Предложен алгоритм, являющийся обобщением метода наименьших квадратов, который позволяет получать сильно состоятельные оценки параметров билинейных динамических систем при наличии помех наблюдения в выходном сигнале в условиях отсутствия информации о законе распределения помех.

Ключевые слова: оценивание параметров, модель выходной ошибки, метод наименьших квадратов, билинейные динамические системы.

Введение

Билинейные модели являются простейшим обобщением линейных динамических систем. Моделирование физических процессов с помощью билинейных систем находит применение во многих областях науки, таких как ядерная физика, электрические сети, химическая кинетика, гидродинамика и т.д. [1].

В настоящее время активно развиваются методы идентификации билинейных динамических систем с помехой в выходном сигнале, такие как инструментальные переменные [2], компенсирующий смещение метод наименьших квадратов [3], метод максимального правдоподобия [4] и методы на основе высших статистик [5]. Анализ существующих

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16-18 марта, 2012.

методов показал, что метод инструментальных переменных имеет малую точность, а остальные методы требуют априорную информацию, которая, обычно неизвестна: знания дисперсий помех, знания закона распределения и т.д.

В работе предложен метод, являющийся обобщением метода наименьших квадратов [6], который позволяет получать сильно состоятельные оценки параметров билинейных систем, и не требует, по сравнению с методом наименьших квадратов, дополнительной априорной информации.

Постановка задачи

Рассмотрим билинейную динамическую систему, описываемую следующими стохастическими уравнениями с дискретным временем:

$$z_i - \sum_{m=1}^{r} b_0^{(m)} z_{i-m} = \sum_{m=0}^{r_1} a_0^{(m)} x_{i-m} + \sum_{m=0}^{r_2} \sum_{k=1}^{r_3(m)} c_0^{(mk)} x_{i-m} z_{i-k}, \quad (1)$$

$$y_i = z_i + \xi(i), .$$

где z_i, y_i -ненаблюдаемая и наблюдаемая выходные переменные; x_i – наблюдаемая переменная; $\xi(i)$ – помеха наблюдения в выходном сигнале.

Предположим, что выполняются следующие условия:

1 . Множество $\tilde{B}$, которому априорно принадлежат истинные значения устойчивой, наблюдаемой и управляемой билинейной динамической системы является компактом.

2. Случайный процесс $\{\xi(i)\}$, удовлетворяют следующим условиям:

$E(\xi(i+1)/F_i) = 0$ п.н., $E(\xi^2(i+1)/F_i) = C(i+1) \le \pi < \infty$ п.н.,

$E(\xi^4(i)) \le \pi < \infty, E(\xi^2(i)) \le \pi$, где F_i $-\sigma$ - алгебра, индуцированная семейством случайных величин $\{\xi(t), t \in T_i\}, T_i = \{t; t \le i, t \in Z_c$ - множество целых чисел$\}$.

3. Входной сигнал x_i является случайным процессом с $E(x_i) = 0$, $E(x_i^2) = \overline{\sigma}_x^2 < \infty$ и удовлетворяет условию постоянного возбуждения порядка $r_1 + 1$, т.е. с вероятностью 1 предел существует

$$\lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} x_{r_1}(i) x_{r_1}^T(i) = H_{xx},$$

где $x_{r1}(i) = (x_i, .. x_{i-r1})^{\mathrm{T}} \in R_{r1} + 1$ и матрица H_{xx} положительно определена.

4. $\{x_i\}$ статистически не зависит от $\{\xi(i)\}$.

Заметим, что из предположений 1- 4, непосредственно следует, что для случайного процесса $\{z_i\}$ с начальными условиями $\left(z(0)=\ldots=z(1-r)=0\right)$ с вероятностью 1 существует предел

$$\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N}\begin{pmatrix}\phi_z(i)\\ \hdashline \phi_x(i)\\ \hdashline \phi_{xz}(i)\end{pmatrix}\left(\phi_z^T(i)\;\vdots\;\phi_x^T(i)\;\vdots\;\phi_{xz}^T(i)\right)=$$

$$=\lim_{N\to\infty}\frac{1}{N}\left(\begin{array}{c:c:c}\sum_{i=1}^{N}\phi_z(i)\phi_z^T(i) & \sum_{i=1}^{N}\phi_z(i)\phi_x^T(i) & \sum_{i=1}^{N}\phi_z(i)\phi_{xz}^T(i)\\ \hdashline \sum_{i=1}^{N}\phi_x(i)\phi_z^T(i) & \sum_{i=1}^{N}\phi_x(i)\phi_x^T(i) & \sum_{i=1}^{N}\phi_x(i)\phi_{xz}^T(i)\\ \hdashline \sum_{i=1}^{N}\phi_{xz}(i)\phi_z^T(i) & \sum_{i=1}^{N}\phi_{xz}(i)\phi_x^T(i) & \sum_{i=1}^{N}\phi_{xz}(i)\phi_{xz}^T(i)\end{array}\right)=$$

$$=\lim_{N\to\infty}\frac{1}{N}\left(\begin{array}{c:c:c}H_{zz} & H_{zx} & H_{z\cdot xz}\\ \hdashline (H_{zx})^T & H_{xx} & H_{x\cdot xz}\\ \hdashline (H_{z\cdot xz})^T & (H_{x\cdot xz})^T & H_{xz\cdot xz}\end{array}\right)=H^*,$$

$$\varphi_y(i)=\left(y_{i-1},\ldots y_{i-r}\right)^T\in R^r,\;\;\varphi_x(i)=\left(x_i,\ldots,x_{i-r_1}\right)^T\in R^{r_1+1},$$

$$\varphi_{xz}(i)=\left(x_i z_{i-1},\ldots,x_i z_{i-r_3(0)}\;\vdots\;x_{i-1}z_{i-1},\ldots,x_{i-1}z_{i-r_3(1)}\;\vdots\;\ldots\right.$$

$$\left.\ldots\;\vdots\;x_{i-r_2}z_{i-1},\ldots,x_{i-r_2}z_{i-r_3(r_2)}\right)^T\in R^{r_3(0)+\ldots+r_3(r_2)+r_2+1},$$

причем H^* существует, ограничена и положительно определена.

Требуется определять оценки неизвестных коэффициентов динамической системы описываемой уравнением (1) по наблюдаемым последовательностям y_i, x_i, при известных порядках, r, r_1, r_2 и r_3 определить оценки истинных значений параметров.

Критерий для оценивания параметров

Система может быть записана как линейная регрессия

$$y_i=\varphi_i^T\theta+\varepsilon_i,\;(2)$$

где $\varphi_i=\left(\varphi_y^T(i)\;\vdots\;\varphi_x^T(i)\;\vdots\;\varphi_{xy}^T(i)\right)^T\in R^{r+r_1+r_3(0)+\ldots+r_3(r_2)+r_2+2}$,

,

$$\varphi_{xy}(i)=\left(x_i y_{i-1},\ldots,x_i y_{i-r_3(0)} \;\vdots\; x_{i-1}y_{i-1},\ldots,x_{i-1}y_{i-r_3(1)} \;\vdots\; \ldots \right.$$
$$\left. \ldots \;\vdots\; x_{i-r_2}y_{i-1},\ldots,x_{i-r_2}y_{i-r_3(r_2)}\right)^T \in R^{r_3(0)+\ldots+r_3(r_2)+r_2+1},$$

$$\theta_0=\left(b_0^T \;\vdots\; a_0^T \;\vdots\; c_0^T\right)^T \in R^{r+r_1+r_3(0)+\ldots+r_3(r_2)+r_2+2},$$

$$b_0=\left(b_0^{(1)}..b_0^{(r)}\right)^T \in R^r,\quad a_0=\left(a_0^{(1)}..a_0^{(r_1)}\right)^T \in R^{r_1+1},$$

$$c_0=\left(c_0^{(01)}\ldots c_0^{(0r_3(0))} \;\vdots\; c_0^{(11)}\ldots c_0^{(1r_3(2))} \;\vdots\; \ldots \right.$$
$$\left. \ldots \;\vdots\; c_0^{(r_2 1)}\ldots c_0^{(r_2 r_3(r_2))}\right)^T \in R^{r_3(0)+\ldots+r_3(r_2)+r_2+1},$$

$$\varepsilon_i=\xi(i)-\sum_{m=1}^{r} b_0^{(m)}\xi(i-m)-\sum_{m=0}^{r_2}\sum_{k=1}^{r_3(m)} c_0^{(mk)}x_{i-m}\xi(i-k).$$

Из предположения (2) следует, что обобщенная ошибка имеет нулевое среднее значение, а из предположения (3) - что ее локальная дисперсия с вероятностью 1 будет равна:

$$\bar{\sigma}_\varepsilon^2=\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N}E\left(\left(\varepsilon_i(b_0,c_0,i)\right)^2\right)=\bar{\sigma}_\xi^2+\bar{\sigma}_\xi^2 b_0^T b_0+\bar{\sigma}_\xi^2\bar{\sigma}_x^2 c_0^T c_0=$$
$$=\bar{\sigma}_\xi^2(1+b_0^T b_0+\bar{\sigma}_x^2 c_0^T c_0)=\bar{\sigma}_\xi^2\omega(b_0,c_0).$$

Тогда определим оценку $\hat{\theta}(N)$ неизвестных параметров θ из условия минимума суммы взвешенных квадратов обобщённых ошибок $\left(\varepsilon_i(b_0,c_0,i)\right)^2$ с весом $\omega(b,c)$, т.е.

$$\min_{\theta\in B}\sum_{i=1}^{N}\frac{\left(y_i-\varphi_i^T\theta\right)^2}{\left(1+b^T b+\bar{\sigma}_x^2 c^T c\right)}=\min_{\theta\in B}\frac{U_N(b,a,c)}{\omega(b,c)}. \qquad (3)$$

Имеет место, следующая теорема:

Теорема. Пусть некоторый случайный процесс $\{y_i, i=\ldots-1,0,1,\ldots\}$ *описывается уравнением (1) с начальными нулевыми условиями и выполняются предположения 1- 4. Тогда оценка* $\hat{\theta}(N)$*, определяемая выражением (3) с вероятностью 1 при* $N\to\infty$*, существует, единственная и является сильно состоятельной оценкой, т.е.*

$$\hat{\theta}(N)\xrightarrow[N\to\infty]{П.Н.}\theta_0.$$

В формуле (3) используется дисперсия входного сигнала, которая обычно неизвестна. Согласно теореме Манна-Вольда [7]: если случай-

ная величина $\hat{\bar{\sigma}}_x^2$ сходится почти наверное соответственно к постоянной $\bar{\sigma}_x^2$, то любая непрерывная функция $J(\hat{\bar{\sigma}}_x^2)$ сходится почти наверное к постоянной $J(\bar{\sigma}_x^2)$:

$$\hat{\bar{\sigma}}_x^2 \xrightarrow{п.н.} \bar{\sigma}_x^2,\ J(\hat{\bar{\sigma}}_x^2) \xrightarrow{п.н.} J(\bar{\sigma}_x^2), \qquad (14)$$

Следовательно, если заменить в (3) $\bar{\sigma}_x^2$ оценками $\hat{\bar{\sigma}}_x^2$ оценки параметров $\hat{\theta}$ останутся сильно состоятельными.

Состоятельная и несмещенная оценка дисперсии $\hat{\bar{\sigma}}_x^2$ может быть получена как

$$\hat{\bar{\sigma}}_x^2 = (N-1)^{-1}\sum_{i=1}^{N}(x_i - \bar{x})^2,\ \ \bar{x} = N^{-1}\sum_{i=1}^{N} x_i.$$

Результаты моделирования

Предложенный алгоритм (3) был реализован в Matlab и сравнен с рекуррентным алгоритмом наименьших квадратов. Динамическая система описывается уравнениями:

$$z_i - 0.7z_{i-1} + 0.4z_{i-2} = 0.3x_i + 0.7x_{i-1} + 0.2x_{i-2} + 0.2x_i z_{i-1},$$
$$y_i = z_i + \xi(i). \qquad (18)$$

На вход подавался сигнал: $x_i + 0.5 \cdot x_{i-1} = \zeta_i + 0.8 \cdot \zeta_{i-1} + 0.6 \cdot \zeta_{i-2}$.

Предложенный в статье нелинейный метод наименьших квадратов (НМНК), сравнивался с методом наименьших квадратов (МНК) и расширенным методом инструментальных переменных (РИП) [2]. Алгоритмы сравнивались по следующим характеристикам:

относительной погрешностью оценивания параметров:

$$\delta\theta = \left\|\hat{\theta} - \theta_0\right\| \Big/ \left\|(\theta_0)\right\| \cdot 100\,\%,$$

относительной погрешностью моделирования:

$$\delta z = \left\|\hat{z} - z\right\| \Big/ \left\|z\right\| \cdot 100\,\%,$$

где $z = \left|z_i, \ldots z_N\right|^T$ – вектор выходной ненаблюдаемой переменной,

$\hat{z} = \left|\hat{z}_i, \ldots \hat{z}_N\right|^T$ – оценка вектора выходной ненаблюдаемой переменной, полученная с помощью модели.

Количество наблюдений $N = 1000$. В таблице 1 приведены средние значения и среднеквадратические отклонения относительных погрешностей, рассчитанные по 50 процедурам оценивания.

Табл. 8. Относительные погрешности оценивания параметров и моделирования

	σ_ξ / σ_x	МНК	РИП	НМНК
$\delta\theta$,%	0.2	19.88±2.26	10.40±9.54	2.61±1.01
δz , %		9.78±0.99	15.47±15.59	2.10±0.56
$\delta\theta$,%	0.5	56.62±3.76	28.09±24.50	8.06±4.06
δz , %		22.71±0.97	75.97±180.4	5.85±2.02
$\delta\theta$,%	0.75	70.50±3.48	58.83±50.04	12.12±5.86
δz , %		27.20±1.11	-	9.41±2.90

Как видно из таблицы предложенный алгоритм дает наименьшие относительные погрешности, как для оценивания параметров, так и для моделирования. Инструментальные переменные оценивают параметры точнее, чем классический МНК, однако дают большую погрешность моделирования, а при высоком уровне помех оценивание параметров с помощью РИП, может приводить к неустойчивости динамической системы.

Выводы

В работе предложен алгоритм для оценивания параметров билинейных динамических систем с помехой наблюдения. В среде MATLAB создано программное обеспечение, результаты моделирования подтверждают эффективность работы алгоритма. Дальнейшее направление исследований может быть направлено на обобщение предложенного алгоритма на случай более сложных моделей шума.

Список источников

1. R.R. Mohler, Bilinear Control Processes: With Applications to Engineering, Ecology, and Medicine. New York: Academic Press, 1973.

2. M. S. Ahmed, Parameter estimation in bilinear systems by instrumental variable methods. International Journal of Control, Vol. 44(4), pp.1177-1183, 1986.

3. M. Ekman, Modeling and control of bilinear systems: application to the activated sludge process. PhD thesis 2005.

4. M. M. Gabr, T. Subba Rao, On the identification of bilinear systems from operating records. International Journal of Control, Vol. 40(1), pp.121-128, 1984.

5. V. Tsoulkas, P. Koukoulas, N. Kalouptsidis. Identification of input-output bilinear systems using cumulants. In Proceedings of the 6th IEEE International Conference on Electronics, Circuits and Systems, Pafos, Greece, pp. 1105-1108, 1999.

6. Кацюба О.А. Теория идентификации стохастических динамических систем в условиях неопределенности: монография. – Самара: СамГУПС, 2008. – 119с. – ISBN 978-5-98941-079-8.

7. http://en.wikipedia.org/wiki/Continuous_mapping_theorem# (дата обращения: 20.04.2009).

Geospatial Semantic Web — расширение семантической паутины для описания и обработки пространственных данных

С. Кузьмин

to.stepan.kuzmin@gmail.com

Уральский государственный горный университет, Екатеринбург, Россия

Аннотация. В данной работе приводится краткое введение в проблему представления и обработки пространственных данных в семантической паутине, описываются существующие технологии и стандарты.

Ключевые слова: пространственные данные; семантическая паутина; Geospatial Semantic Web; GeoRSS; GeoRDF; GeoSPARQL.

Введение

Стремительный рост Всемирной паутины привёл к пониманию того, что существующие методы поиска и обработки пространственной информации малоэффективны. Отчасти, это связанно с тем, что широко используемая в данный момент геореляционная модель представления данных накладывает определённые ограничения на обработку распределённой геоинформации. Недостатком геореляционной модели данных является неприспособленность реляционных таблиц для семантического анализа и полнотекстового поиска в распределённых базах пространственных данных [1]. Однако эти задачи довольно легко решаются с помощью использования модели пространственных данных, основанной на технологиях стандарта семантической паутины (Semantic Web). Появление семантической паутины предполагает качественное улучшение

Игнатов Д. И., Яворский Р. Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16–18 марта, 2012

методов извлечения информации за счёт использования семантики данных в процессе поиска [11]. Такое развитие технологий требует особого подхода к организации пространственных данных — необходимо чтобы, семантика такого вида данных описывалась надлежащим образом [7].

Краткий обзор технологий и стандартов

На данный момент существует несколько, ещё не стандартизированных окончательно, технологий для описания и обработки пространственных данных в семантической паутине. Существует четыре способа представления семантики пространственных данных [3]:

- Естественный язык с минимумом разметки. При этом поисковые системы могут обрабатывать только небольшие подмножества естественного языка (например, основные инструкции поиска) и поддерживают небольшой набор техник полнотекстовой индексации естественного языка.
- Простые метаданные, в которых семантику описывает некоторый набор ключевых слов (например языки основанные на XML). Несмотря на то, что пользователям необходимо взаимодействовать с метаданными во время извлечения информации, семантика метаданных определяется документами или, например, поисковыми системами.
- Модели представления данных. Самая популярная модель представления данных — Resource Description Framework (RDF). RDF представляет утверждения о ресурсах в виде, пригодном для машинной обработки. Концептуальная структура приводится в описании терминов сущностей, отношений и атрибутов. Семантика модели представления данных также определяется интерфейсами, документами и поисковыми системами.
- Логическая семантика обеспечивает соответствие между терминами и объектами реального мира, позволяя производить автоматические рассуждения. Также как метаданные и модели представления данных, логическая семантика принципиально определяется интерфейсами, документами и поисковыми системами.

Базовым инструментом описания семантики пространственной информации являются метаданные [2]. В настоящее время существует ряд стандартов на метаданные электронных карт и геоинформации в международном [4] и национальном масштабах, ведутся работы по дальнейшему развитию системы международных стандартов и их взаимодействию с национальными стандартами, в которых активное участие принимает и наша страна [2].

Прежде всего, необходимо описать онтологию для представления пространственных данных [9]. Работа над этим ведётся в рамках разработки основной онтологии и словаря OWL для представления пространственных свойств ресурсов Всемирной паутины. В рамках разработки этого стандарта было решено использовать синтаксис похожий на модель объектов в GeoRSS (например, описание прямоугольников, точек, линий, и полигонов). В стандарт также включается описание отношений пространственных объектов (идентичность, пересекаемость, непересекаемость и прочее). Подробнее об этом можно узнать из отчёта W3C Incubator Group от 23 октября 2007 года [8].

GeoSPARQL — это расширение SPARQL, представляющее собой язык запросов к пространственным данным, представленным по модели RDF [10]. Создан на основе существующих стандартов W3C (RDF, OWL, SPARQL) и OGC (Simple Features, Spatial Relations). Расширяет точки доступа SPARQL (SPARQL-endpoints) возможностью манипулировать пространственными данными. Предоставляет основу для пространственных рассуждений (spatial reasoning). Цель стандарта OGC GeoSPARQL состоит в поддержке представления и обработки пространственных данных в семантической паутине. Стандарт определяет словарь представления пространственных данных в RDF и расширение SPARQL для обработки пространственных данных. Например, так выглядит GeoSPARQL запрос на выборку всех аэропортов расположенных вблизи Лондона:

```
PREFIX co: <http://www.geonames.org/countries/#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
SELECT ?link ?name ?lat ?lon
WHERE  {
 ?link gs:within(51.139725 -0.895386 51.833232 0.645447) .
 ?link gn:name ?name .
 ?link gn:featureCode gn:S.AIRP .
 ?link geo:lat ?lat .
 ?link geo:long ?lon
}
```

Этот запрос можно выполнить на сайте GeoSPARQL [5], созданного и поддерживаемого компанией KONA на основе популярного фреймворка Apache Jena и собственной разработки [6]. Результат будет примерно таким[1]:

```
-------------------------------------
| link
=====================================
| <http://sws.geonames.org/2647216/>
```

[1] Ответ можно получить в Plain Text, XML или KML.

```
...
| <http://sws.geonames.org/6691396/>
-------------------------------------

      ------------------------------
      | name                       |
      ==============================
      | "Heathrow"                 |

      | "Heathrow Terminal 5"      |
      ------------------------------

            ---------------------------------------------
            | lat                 | lon                 |
            =============================================
            | "51.4711455584879" | "-0.456490516662598" |

            | "51.4729365901483" | "-0.450611114501953" |
            ---------------------------------------------
```

Заключение

Исследования в области Geospatial Semantic Web нацелены на повышение эффективности и точности обработки пространственных данных в сети. Представление пространственных данных в сети, реализованное на основе методов и стандартов Semantic Web, таких, как GeoRSS и GeoRDF, позволяет описывать данные на языке, приближенном к естественному. Такая модель хорошо представляет разнородные пространственные данные и обеспечивает значительно большую скорость выполнения транзакций. Технология Geospatial Semantic Web предоставляет возможность создания логически объединённых децентрализованных пространственных баз данных и позволяет осуществлять более эффективный поиск требуемой пространственной информации.

Список источников

1. Калантаев П. А. Семантическая организация пространственных данных.
2. Лунева Н. В. Инструменты текстового отображения семантики геопространственной информации.
3. Egenhofer M. J. Toward the semantic geospatial web.

4. ISO 6709 Standard representation of geographic point location by coordinates. ISO/TC211. Обзор по стандартам ISO/TC 211 «Географическая ифнормация/Геоматика».
5. http://geosparql.org/.
6. Geospatial Reasoning for the Semantic Web http://code.google.com/p/geospatialweb/.
7. Hebeler J., Fisher M., Blace R., Perez-Lopez A. Semantic Web Programming.
8. W3C Incubator Group Report 23 October 2007 http://www.w3.org/2005/Incubator/geo/XGR-geo-20071023/.
9. Wang Y., Dai J., Sheng J., Zhou K., Gong J. Geo-ontology design and its logic reasoning.
10. Battle R., Kolas D. Enabling the Geospatial Semantic Web with Parliament and GeoSPARQL.
11. Wallgrub J.O., Bhatt M. An Architecture for Semantic Analysis in Geospatial Dynamics.

Сравнение методов извлечения ключевых слов из текстов на естественных языках

Д. Р. Недумов

НИУ ВШЭ, Москва, Россия

Аннотация. В связи с быстрым ростом количества текстовой информации крайне осложняется проведение ее анализа без привлечения автоматических средств. Проблемы анализа текстовой информации рассматриваются в рамках обработки естественного языка (Natural Language Processing, NLP). Тема данного исследования - выделение ключевых слов (одна из подзадач NLP.), т.е. слов, определяющих содержание текста. К настоящему времени разработано много методов выделения ключевых слов, в связи с чем существует необходимость их систематизации, сравнительного анализа. Цель исследования - рассмотреть существующие методы выделения ключевых слов, применить их для анализа массивов реальных данных, провести сравнительный анализ рассматриваемых методов и моделей.

Ключевые слова: выделение ключевых слов; анализ текстов на естественных языках; статистические меры значимости ключевых слов.

Введение

Проблема: в связи с необычайно быстрым ростом количества текстовой информации, генерируемой пользователем, крайне осложняется проведение ее анализа без привлечения автоматических средств обработки естественного языка. Проблемы анализа текстовой информации рассматриваются в рамках обработки естественного языка (Natural

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16-18 марта, 2012.

Language Processing, NLP) [1] – одного из направлений компьютерных наук (computer science) и математической лингвистики. В задачах анализа и классификации текстов, с которыми сталкивается NLP, возникает подзадача выделения ключевых слов (keywords extraction), т.е. слов, определяющих содержание текста. К настоящему времени разработано довольно много подходов к проблеме выделения ключевых слов, в связи с чем существует необходимость их систематизации, проведения экспериментов по сравнительному анализу их эффективности, определения границ применимости методов для оптимального их использования при решении прикладных задач.

Задача: Рассмотреть существующие методы выделения ключевых слов, применить их для анализа массивов реальных данных (коллекции текстовых документов), провести сравнительный анализ рассматриваемых методов и моделей.

Методы извлечения ключевых слов

На практике задача извлечения ключевых слов может решаться с помощью статистических методов (конечно, еще существуют методы, использующие лингвистику). Одно из основных преимуществ статистических методов — возможность их успешного применения к корпусам[1] текстов на разных языках без существенных изменений применяемых алгоритмов и конкретных реализаций. При выделении ключевых слов статистическими методами используются различные числовые меры, которые позволяют определить «вес» каждого слова, его значимость. Далее мы рассмотрим наиболее известные из таких мер. В трудах, посвященных данной проблеме в силу многообразия целей и подходов использование метрик рассматривается в разных терминах, в частности: Uchyigit и Clarky в статье [2] с помощью различных числовых мер исследуют зависимость между наличием в некотором документе[2] заданного слова и принадлежностью этого документа к некоторой категории[3]. Такой подход удобен для классификации текстов. Manning и Schütze в книге [3] используют числовые меры для выявления т.н. коллокаций (устойчивых словосочетаний), то есть проблема рассматривается в терминах совместной встречаемости слов.

---

[1] Корпус (коллекция) – собрание текстов, объединённых некоторым общим признаком. Например, корпус текстов из газет политической тематики.

[2] Под документом (текстом) здесь и далее будем понимать набор слов и символов. В данной задаче предполагается, что документ принадлежит корпусу и может принадлежать категориям корпуса.

[3] Категория – набор документов, содержащихся в данной коллекции и обладающих некоторым объединяющим их признаком.

Статистические методы для выявления коллокаций использовались также в статье Е. В. Ягуновой и Л. М. Пивоваровой [4].

Так как наше исследование ориентировано именно на работу с ключевыми словами, будем использовать более удобные термины для описания числовых мер. Будем рассматривать наличие некоторого слова w (про это слово необходимо выяснить, является ли оно ключевым) в тексте t и наличие у текста t некоторой особенности p (т.е. текст t обладает признаком p). Под признаком будет пониматься, например, принадлежность текста к некоторой категории или факт того, что текст является/не является спамом (задача фильтрации спама).

Для удобства восприятия представим используемые далее переменные в таблице (переменные в ячейках таблицы обозначают количество текстов в коллекции, удовлетворяющих условиям, записанным в первой строке и первом столбце таблицы):

	Текст t обладает признаком p	Текст t не обладает признаком p	
Слово w принадлежит тексту t	E_{11}	E_{12}	
Слово w не принадлежит тексту t	E_{21}	E_{22}	
			N - общее число текстов в корпусе

Критерий Хи-квадрат (Chi-Squared Statistic или Chi-Squared criterion).

Данная величина показывает зависимость между фактом содержания слова w в тексте t и наличием у текста t признака p. (чем больше ее значение, тем сильнее зависимость).

Рассчитывается по формуле [5]:

$$\chi^2 = \sum_{i,j} \frac{\left(E_{ij} - T_{ij}\right)^2}{T_{ij}}$$

где: E_{ij} – эмпирическая (наблюдаемая) частота (см. таб.)

$T_{ij} = \frac{E_{\cdot j}}{N} \cdot \frac{E_{i\cdot}}{N} \cdot N$ – теоретическая (ожидаемая) частота

$E_{\cdot j} = \sum_i E_{ij}$

$E_{i\cdot} = \sum_j E_{ij}$

Указанная формула может быть применена к таблицам произвольного размера, поскольку мы определили расчетную формулу, основываясь на таблице и расположении элементов в ней (индексы E соответст-

вуют номерам строк и столбцов таблицы соответственно), но мы остановимся на рассмотрении случая с таблицей 2-на-2, для которого получена [5] упрощенная формула:

$$\chi^2 = \frac{N \cdot (E_{11} \cdot E_{22} - E_{12} \cdot E_{21})^2}{(E_{11} + E_{12}) \cdot (E_{11} + E_{21}) \cdot (E_{12} + E_{22}) \cdot (E_{21} + E_{22})}$$

Для критерия Хи-квадрат замечено ошибочное выявление редких слов, т.е. таких слов, которые редко употребляются в языке в целом. Считается, что роль таких слов в задачах анализа текстов невелика, и их не стоит использовать в качестве ключевых. Данное явление описано исследователями [5] и объясняется, например, возможностью существования в исследуемом корпусе одного редкого слова, содержащегося в одном из документов. Это слово будет ошибочно выделено как ключевое для этого текста (т. к. в таком случае зависимость признака текста с редким словом и наличия слова в тексте будет велика), хотя оно, быть может, не несет совершенно никакой информации о нем.

NGL показатель.

$$NGL = \frac{\sqrt{N} \cdot (E_{11} \cdot E_{22} - E_{12} \cdot E_{21})}{\sqrt{(E_{11} + E_{12}) \cdot (E_{11} + E_{21}) \cdot (E_{12} + E_{22}) \cdot (E_{21} + E_{22})}}$$

NGL – мера, основанная на статистике Хи-квадрат, имеет некоторые отмечаемые исследователями [2] преимущества по сравнению со статистикой Хи-квадрат. А именно: NGL в отличие от статистики Хи-квадрат принимает как положительные (это означает, что есть связь между словом и наличием признака у текста), так и отрицательные (это означает что есть связь между словом и отсутствием признака у текста) значения. Таким образом при необходимости можно выделять слова, связанные с наличием или отсутствием признака у текста, тогда как статистика Хи-квадрат не дает такой возможности.

GSS показатель.

$$GSS = E_{11} \cdot E_{22} - E_{12} \cdot E_{21}$$

Существенно упрощенная версия Хи-квадрат статистики [2].

Знаменатель опущен, т.к. было замечено, что он завышает вес рассматриваемого слова в случаях, когда оно является редким, или, когда проверяется наличие у текста редкого признака (например, принадлежности редкой категории, т.е. категории с малым количеством текстов).

Взаимная информация (Mutual Information, MI).

Мера взаимной информации измеряет насколько много информации (в смысле теории информации) о документе содержит рассматриваемое

слово. Существует несколько вариантов расчетной формулы. G. Uchyigit и K. Clark предложили [2] следующий вариант формулы:

$$MI_{w,p} = \frac{E_{11} \cdot N}{(E_{11} + E_{21}) \cdot (E_{11} + E_{12})}$$

В своей статье они характеризуют взаимную информацию как достаточно эффективную меру. Тем не менее, был выделен недостаток: редкие слова получают большой вес. Эта проблема решается удалением редких слов из текста перед работой с мерой.

Другой вариант формулы рассматривается в книге Christopher D. Manning и др. [5]. Этот вариант точнее выражает взаимосвязь между словом и признаком у текста, т.к. формула включает слагаемые отражающие все комбинации наличия/отсутствия слова/признака в тексте (то есть, использованы значения из всех клеток в таб.). Вопрос о введении в формулу логарифма рассмотрен в книге М. Вернера [6].

$$MI_{w,p} = \sum_{i,j} \frac{E_{ij}}{N} \cdot log_2 \left(\frac{N \cdot E_{ij}}{E_{i\cdot} \cdot E_{\cdot j}} \right)$$

Проблема ложного выделения редких слов выявлена и этими исследователями, но замечено, что количество таких ошибок для статистики Хи-квадрат больше, чем в случае использования меры взаимной информации.

TF-IDF.

Эта мера измеряет «уникальность» рассматриваемого слова w для данного документа t. Слова, получающие большие значения по этой мере, частотны в рассматриваемом тексте, но редки в других текстах коллекции. Мера рассчитывается как произведение двух множителей: TF (term frequency) – частота слова и IDF – (inverse document frequency) – обратная частота документа [7].

$$TFIDF = TF \cdot IDF = (n(w)/n) \cdot log(N/N(w))$$

где: $TF = n(w)/n$,
$n(w)$ – число вхождений слова w в документ t, n – количество слов в документе t.
$IDF = log(N/N(w))$,
N – количество документов в корпусе, $N(w)$ – количество документов корпуса, содержащих слово w

Экспериментальное сравнение

В качестве данных для экспериментального сравнения статистических мер используем небольшую коллекцию документов из Брауновского корпуса[1][8]. В выбранной коллекции содержатся тексты из категорий «government» (правительственные документы, буклеты), «humor» (юмористические тексты), «news» (новостные тексты), «religion» (религиозные тексты). Ключевые слова будем выделять из текста категории «government». Технически задачу будем решать средствами библиотеки NLTK языка Python [9]. Исключим из рассмотрения стоп-слова, пользуясь корпусом стоп-слов английского языка, а также числовые данные, поскольку числа мы не рассматриваем как ключевые слова. Таблицу для топ-15 результатов приводим ниже:

Хи-квадрат	MI	TF-IDF	GSS	NGL
year	solutions	**Business**	individual	year
businesses	**Owners**	**SBA**	development	property
efficiency	stipulate	Small	Owners	**business**
Government's	Products	**Business**	stipulate	production
educational	Proposed	Small	Facilities	purposes
measures	Facilities	**Loans**	encourages	assistance
sponsored	locating	Concerns	One-day	available
large	encourages	**Administration**	semi-processed	use
many	One-day	Property	undertook	**management**
economic	semi-processed	**Credit**	Participation	may
indicate	**Interior**	**Government**	unavailable	Section
assure	Interest	Available	Eligibility	**SBA**
specified	undertook	Farm	**Developing**	services
exploration	Participation	**D.C.**	exploitation	**Federal**
locating	unavailable	assistance	supplements	financing

Жирным в таблице выделены слова, потенциально релевантные для рассматриваемого текста (исходя из того, что он содержится в тематической категории «government»). Наибольшее число таких слов были выделены мерами Хи-квадрат и TF-IDF, поэтому в данном случае их работа наиболее эффективна.

[1] Брауновский корпус (Brown Corpus) – компьютерный лингвистический корпус английского языка. Создан в 60-е годы в Университете Брауна, насчитывает около 1 млн. слов. Тексты корпуса разделены по тематическим категориям.

Выводы

Проведен обзор методов извлечения ключевых слов из текстов на естественных языках. Экспериментальное сравнение показывает, что на рассмотренных данных лучше сработали меры Хи-квадрат и TF-IDF.

В дальнейшем планируется провести эксперименты на данных полицейских отчетов в рамках проекта «Cordiet»[1]*[10] и написать полноценный обзор методов извлечения ключевых слов из текстов на естественных языках.*

Список источников

1. Jurafsky D., Martin J.H. Speech and language processing: An introduction to natural language processing, speech recognition, and computational linguistics. 2d ed. Upper Saddle River, NJ: Pearson Prentice Hall, 2009.

2. Uchyigit G., Clark K. An Experimental Study of Feature Selection Methods for Text Classication // Personalization Techniques And Recommender Systems, World Scientific Publishing, pp. 303-320, 2008.

3. Manning C., Schütze H. Foundations of Statistical Natural Language Processing, MIT Press. Cambridge, MA: May 1999.

4. Ягунова Е. В., Пивоварова Л. М. От коллокаций к конструкциям // Русский язык: конструкционные и лексико-семантические подходы / Отв. ред. С.С.Сай. СПб, 2011.

5. Manning C. D., Raghavan P., Schütze H. Introduction to Information Retrieval, Cambridge University Press. 2008.

6. Вернер М. Основы кодирования. Учебник для ВУЗов. Москва: Техносфера, 2004. - 288с.

7. Uzun Y. Keyword Extraction Using Naive Bayes http://www.cs.bilkent.edu.tr/~guvenir/courses/cs550/Workshop/Yasin_Uzun.pdf

8. http://khnt.hit.uib.no/icame/manuals/brown/INDEX.HTM Brown Corpus Manual 2012

[1] Cordiet (Concept Relation Discovery and Innovation Enabling Technology) – проект по созданию инструментария для получения данных из текстовых массивов. Разработка проекта совместно ведется Католическим университетом Лёвена и НИУ ВШЭ (Москва).

9. Bird S., Klein E., Loper E. Natural Language Processing with Python, O'Reilly Media, 2009. 512 p.

10. Poelmans J., Elzinga P., Neznanov A., Viaene S., Kuznetsov S., Ignatov D., Dedene G. Concept Relation Discovery and Innovation Enabling Technology (CORDIET) // In CEUR Workshop proceedings Vol-757, CDUD'11 – Concept Discovery in Unstructured Data, pp. 53-62, 2011.

Об одной задаче семантической классификации цифровых изображений

Паначёв М.А., Парфененков Б.В.

Институт математики и компьютерных наук

Уральский федеральный университет имени первого

Президента России Б.Н.Ельцина

Аннотация. Рассматривается задача автоматической классификации цифровых изображений с целью определения набора семантических категорий качественной оценки эмоционального восприятия их содержимого. Известно, что решение данной задачи проходит в пять этапов: выбор структуры пространства признаков, корреляционный анализ, выбор алгоритма машинного обучения, обучение классификатора, ROC-анализ. Данная работа представляет собой описание структуры и результат экспериментов по оценке информативности вектора признаков, используемого авторами для решения поставленной задачи.

Ключевые слова: классификация, обработка изображений, аннотация изображений, цветовые пространства, пространство признаков, текстурный анализ, машинное обучение.

Введение

Задача автоматической аннотации цифровых изображений представляет собой одну из классических проблем машинного обучения и компьютерного зрения. В современном мире потребность в решении данной задачи заметно увеличилась (растёт с каждым днём). В число актуальных прикладных задач, при решении которых возникает необходимость автоматической аннотации цифровых изображений входят та-

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16-18 марта, 2012.

кие проблемы, как качественный и релевантный поиск изображений в сети Интернет, анализ и фильтрация Интернет-трафика, аннотация цифровых изображений на персональных носителях информации и интернет ресурсах, предназначенных для хранения и публикации пользовательских фотоснимков.

В настоящее время задача классификации, в которой объект может находиться только в одной из заданных категорий, решается с помощью множества хорошо изученных методов теории машинного обучения [4], например: метод k ближайших соседей (kNN), логистическая регрессия, радиально-базисная нейронная сеть, многослойный персептрон [8], байесовский классификатор, машина опорных векторов (SVM) [7] и т.д. При этом решение задачи классификации, в которой объект может принадлежать как одному, так и нескольким классам одновременно, сводится к решению описанной выше задачи лишь при условии, что классы попарно не пересекаются.

Данная работа посвящена задаче автоматической аннотации цифровых изображений, в которой семантические категории представляют собой понятия, определяющие качественные характеристики восприятия содержимого изображений. К ним, например, относятся эпитеты, метафоры и сравнения.

Постановка задачи

Рассмотрим задачу автоматической аннотации цифровых изображений следующим образом.

Имеется множество цифровых[1] изображений $I = I_1 \cup I_2$, где $I_1 = \{img_1, img_2, \dots, img_k\}, I_2 = \{img_{k+1}, img_{k+2}, \dots, img_n\}$, и множество $S = \{s_1, s_2, \dots, s_m\}$ текстовых меток – тегов. Каждому изображению $img_i \in I_1$ поставлен в соответствие *индикатор* – вектор вещественных чисел $v_i = (v_{i1}, v_{i2}, \dots, v_{im})^T$, определяющий отображение-аннотацию $A^*: I \to 2^S$ согласно формуле F[2].

Требуется построить алгоритм классификации $a: I \to 2^S$, аппроксимирующий отображение A^* на всём множестве I.

Метки S, указывающие на определённые визуальные характеристики цифровых изображений, образуют систему семантических категорий.

---

[1] Цифровое изображение $-$ матрица $\left(p_{xy}^{(i)}\right)_{H \times W}$, где $p_{xy}^{(i)} \in [a, b] \cap \mathbb{Z}$

[2] Как правило, в качестве F используется пороговая фильтрация с параметром t:

$$F(t) \equiv v_{ij} > t \ \to \ s_i \in A$$

Если $v_{ij} \in [0,1]\ \forall i, j$, то v_{ij} можно интерпретировать как степень уверенности в том, что $s_j \in S_i$.

Очевидно, что если рассматриваемая система семантических категорий удовлетворяет гипотезе компактности, то решение поставленной задачи находится среди метрических алгоритмов классификации [1].

Поиск решения

В машинном обучении классическая схема поиска решения задачи классификации сложных[1] объектов выглядит следующим образом:

I. Изучение свойств (признаков) анализируемых объектов, определяющих особенности, влияющие на отнесение объекта к той или иной категории.
II. Корреляционный анализ. Выбор структуры пространства признаков.
III. Определение функции расстояния в пространстве объектов через функцию расстояния в пространстве признаков.
IV. Выбор алгоритма машинного обучения.
V. Обучение классификатора.
VI. ROC-анализ.

Данная работа содержит описание структуры пространства признаков, используемого авторами для решения поставленной задачи.

Пространство признаков

Алгоритм определения сходства между изображениями состоит из двух этапов:

1. Каждому изображению ставим в соответствие вектор признаков, определяющий ключевые особенности данного изображения, некоторым образом связанные с элементами множества S:
$$f: I \to \mathbb{R}^M$$
2. Степень сходства между двумя изображениями определим как евклидово расстояние между их образами в пространстве признаков:
$$\rho(img_i, img_j) = \sqrt{\sum_{k=1}^{M} (f_{ik} - f_{jk})^2}$$

При этом вид оператора f зависит от природы множества S, а отношение ρ в общем случае не является метрикой.

[1] С точки зрения их описания и представления.

В ходе данной работы был рассмотрен следующий набор признаков цифровых изображений.

Цветовые гистограммы. Группа признаков, которые характеризуют, насколько много того или иного цвета в изображении. Все цветовое пространство разбивается на M частей. Обычно выбираются гиперкубы (размерность зависит от размерности цветового пространства), которые полностью покрывают всё пространство, и пересечение любой пары – пусто. Далее для каждой части считаем отношение числа пикселей, которые попали в данную часть к общему числу пикселей изображения.

$$H_i = \frac{\sum_{x=1}^{Height} \sum_{y=1}^{Width} p_i(x,y)}{Height * Width}, \text{где}$$

$$p_i(x,y) = \begin{cases} 1, \text{если пиксель с координатами } (x,y) \in M_i \\ 0, \text{иначе} \end{cases}$$

Height и Width в формуле обозначают высоту и ширину изображения, соответственно.

Естественно, здесь идет борьба между потерей качества и большим объемом используемой информации. Чтобы полностью не потерять качество на обычном 24-битном RGB изображении, потребуется использовать 2^{24} части. Безусловно, хранить такой объем информации достаточно сложно, поэтому обычно используют квантование цветового пространства.

Центры масс цветов. Группа композиционных признаков, которые характеризуют, где находится наибольшее скопление определенного цвета. В геометрическом смысле являются центрами масс. В качестве основных цветов были выбраны классические цвета – черный, белый, красный, оранжевый, желтый, зеленый, голубой, синий и фиолетовый.

Чтобы определить для каждого пикселя изображения, к какому цвету из данного набора, он наиболее подходит, необходимо провести цветовую сегментацию изображения (каждый пиксель исходного изображения заменяется пикселем с цветом из заданного набора). Для цветовой сегментации использовался алгоритм классификации K-Means в цветовом пространстве CIE Lab.

Когда в изображении присутствуют только пиксели определенных цветов, достаточно взять сумму пикселей конкретного цвета по каждой координате и нормализовать ее относительно общего числа пикселей заданного цвета.

Нетрудно понять, что центры масс изображения это точка с 2мя координатами (по высоте и ширине). Если обозначить центры масс цветов, как M_i, то формула будет похожа на формулу для цветовых гистограмм

$$M_i.x = \frac{\sum_{x=1}^{Height}\sum_{y=1}^{Width} x * p_i(x,y)}{Count_i},$$

$$M_i.y = \frac{\sum_{x=1}^{Height}\sum_{y=1}^{Width} y * p_i(x,y)}{Count_i},$$

где $Count_i$ – количество пикселей i-го цвета.

Статистические признаки. Группа признаков, характеризующая распределение яркости по изображению. Включает в себя такие признаки, как средняя яркость, среднеквадратичное отклонение от средней яркости, коэффициент асимметрии и медиану. Необходимо вычислить первые три момента для распределения яркости изображения.

Для вычисления математического ожидания достаточно сложить яркости каждого пикселя и нормализовать по количеству пикселей в изображении.

$$EX = \sum_{x=1}^{Height}\sum_{y=1}^{Width} Y(x,y),$$

где Y(x,y) – яркость пикселя с координатами (x,y). Здесь в качестве яркости пикселя в пространстве RGB мы используем формулу

$$Y = 0.3 * R + 0.59 * G + 0.11 * B,$$

где R, G и B – компоненты красного, зеленого и синего цвета соответственно.

Для дисперсии и коэффициента асимметрии можно воспользоваться формулой $\mu_k = E[(Y - EY)^k]$, подставив k = 1 и k = 2 соответственно, где EY – математическое ожидание, а Y – распределение яркости. Чтобы вычислить медиану, необходимо отсортировать яркости всех пикселей и выбрать среднюю. Если Y_i – яркость i-го пикселя, отсортированных по возрастанию ($i \in \{1, 2, \dots, Width * Height\}$), то медиана будет равна $Y_{\frac{Width*Height}{2}}$.

Текстурные признаки. В 1978 года Тамура и Мори [5] сформировали 6 признаков, соответствующих человеческому восприятию: грубость, контраст, направленность, линейность, непрерывность и шероховатость. В ходе исследований было показано, что первые 3 из них являются наиболее информативными, то есть они коррелируют с человеческим восприятием.

Грубость. Данный признак дает представление о размере текстура элемента. Чем выше значение грубости, тем текстура является более выраженной. Грубость изображения вычисляется следующим образом

1. Для каждой точки изображения (n_0, n_1) считается среднее значение яркости по всем окрестностям. Размер окрестности $k * k$ выбирается, как степень двойки, то есть 1x1, 2x2, 4x4, …, 32x32.

$$A_k(n_0, n_1) = \frac{1}{2^k} \sum_{i=1}^{2^{2k}} \sum_{j=1}^{2^{2k}} Y\left(\mathrm{n_0} - 2^{\mathrm{k}-1} + \mathrm{i}, \mathrm{n_1} - 2^{\mathrm{k}-1} + \mathrm{j}\right).$$

2. Далее для каждой точки (n_0, n_1) считается разность E между неперекрывающимися окрестностями по разные стороны от точки в горизонтальном и вертикальном направлении:

$$E_k^h(n_0, n_1) = \left|\mathrm{A_k}(\mathrm{n_0} + 2^{\mathrm{k}-1}, \mathrm{n_1}) - \mathrm{A_k}(\mathrm{n_0} - 2^{\mathrm{k}-1}, \mathrm{n_1})\right|$$
$$E_k^v(n_0, n_1) = \left|\mathrm{A_k}(\mathrm{n_0}, \mathrm{n_1} + 2^{\mathrm{k}-1}) - \mathrm{A_k}(\mathrm{n_0}, \mathrm{n_1} - 2^{\mathrm{k}-1})\right|$$

3. Теперь необходимо для каждой точки (n_0, n_1) выбрать размер S, ведущий к наибольшей разности E:

$$S(n_0, n_1) = \underset{k \in 1..5}{\operatorname{argmax}} \max_{d \in \{h,v\}} E_k^d(n_0, n_1).$$

4. В результате необходимо взять среднее по 2^S, как меру грубости изображения:

$$F_{crs} = \frac{1}{\mathrm{Height} * \mathrm{Width}} \sum_{n0=1}^{Height} \sum_{n1=1}^{\mathrm{Width}} 2^{S(n_0, n_1)}$$

Контраст. В узком смысле слова, контраст отвечает за качество изображения. Если взглянуть более глобально, то данный признак является композицией влияний следующих факторов: разброс яркости; отдаленность белого и черного цвета на яркостной гистограмме; резкость; период повтора некоторого шаблона. Контраст может быть вычислен с помощью следующей формулы:

$$F_{con} = \frac{\sigma}{\alpha_4^z}, \text{а } \alpha_4 = \frac{\mu_4}{\sigma^4},$$

$$\mu_4 = \frac{1}{Height * Width} \sum_{n0=1}^{\mathrm{Height}} \sum_{n1=1}^{Width} (Y(n_0, n_1) - \mu)^4,$$

где μ и σ^2 – математическое ожидание и дисперсия яркости соответственно, а z – экспериментально-полученная величина равная ¼.

Статистические признаки. Моменты высших порядков

Класс инвариантных статистических моментов содержит набор признаков, который инвариантен по отношение к стандартным преобразованиям, таким как растяжение (сжатие), перенос и поворот. Далее кратко дадим описание анализа на основе статистических моментов.

Для непрерывной двумерной функции $f(x,y)$, момент порядка $(p+q)$ определяется, как

$$m_{p,q} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} x^p y^q dxdy, \text{ для } p,q = 0,1,2,\ldots$$

Соответственно центральные моменты вычисляется по формуле

$$\mu_{p,q} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} (x-\bar{x})^p (y-\bar{y})^q f(x,y)dxdy,$$

где $\bar{x} = \frac{m_{1,0}}{m_{0,0}}$ и $\bar{y} = \frac{m_{0,1}}{m_{0,0}}$

В нашем случае $f(x,y)$ это цифровое изображение, поэтому формула выше преобразуется в вид

$$\mu_{p,q} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q f(x,y).$$

Центральные моменты можно нормализовать. Нормализованные центральные моменты обозначаются как:

$$\eta_{p,q} = \frac{\mu_{p,q}}{\mu_{0,0}^{\gamma}}, \text{где } \gamma = \frac{p+q}{2} + 1 \text{ при } p+q = 2,3,\ldots$$

В [15] впервые были рассмотрены нормализованные центральные моменты второго и третьего порядка. С помощью различных алгебраических операций между ними была выведена целая группа инвариантных статистических признаков. Данная группа содержит в себе ортогональные признаки, которые можно использовать для идентификации изображений, которые были преобразованы с помощью растяжения (сжатия), переноса или поворота:

$$\phi_1 = \eta_{2,0} + \eta_{0,2},$$

$$\phi_2 = (\eta_{2,0} - \eta_{0,2})^2 + 4\eta_{11}^2,$$

$$\phi_3 = (\eta_{3,0} - 3\eta_{1,2})^2 + (3\eta_{2,1} - 3\eta_{0,3})^2,$$

$$\phi_4 = (\eta_{3,0} + \eta_{1,2})^2 + (\eta_{2,1} + \eta_{0,3})^2,$$

$$\phi_5 = (\eta_{3,0} - 3\eta_{1,2})(\eta_{3,0} + \eta_{1,2})[(\eta_{3,0} + \eta_{1,2})^2 - 3(\eta_{2,1} + \eta_{0,3})^2],$$

$$\phi_6 = (\eta_{2,0} - \eta_{0,2})(\eta_{3,0} + \eta_{1,2})\left[(\eta_{3,0} + \eta_{1,2})^2 - (\eta_{2,1} + \eta_{0,3})^2\right] + 4\eta_{1,1}(\eta_{3,0} + \eta_{1,2})(\eta_{2,1} + \eta_{0,3}),$$

$$\phi_7 = (3\eta_{2,1} - \eta_{0,3})(\eta_{3,0} + \eta_{1,2})\left[(\eta_{3,0} + \eta_{1,2})^2 - 3(\eta_{2,1} + \eta_{0,3})^2\right] + (3\eta_{1,2} - \eta_{3,0})(\eta_{2,1} + \eta_{0,3})[3(\eta_{3,0} + \eta_{1,2})^2 - (\eta_{2,1} + \eta_{0,3})^2].$$

Направленность. В качестве данного признака рассматривается не ориентация самой картинки, а ориентация текстуры на ней. Чтобы посчитать направленность необходимо посчитать производные Δ_H и Δ_V, которые вычисляются при помощи умножения на соответствующие операторы 3x3:

$$\begin{matrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{matrix} \quad \begin{matrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{matrix}$$

Далее для каждой точки (n_0, n_1) вычисляется следующая величина:

$$\theta(n_0, n_1) = \frac{\pi}{2} + tan^{-1}\frac{\Delta_V(n_0, n_1)}{\Delta_H(n_0, n_1)}.$$

В результате, строим гистограмму из 16 колонок по полученным θ. В качестве итогового признака можно использовать либо полученную гистограмму, либо преобразовать данную гистограмму в число, путем вычисления второго момента у получившегося распределения.

Чтобы использовать получившиеся текстурные признаки, как признаки для каждого пикселя изображения, необходимо произвести некоторые преобразования. Для грубости достаточно провести только пункты с 1 по 3, и возведя $2^{S(n_0,n_1)}$ получим грубость конкретного пикселя. Контраст нужно считать для области 13x13 вокруг данного пикселя. А для направленности можно использовать значение θ.

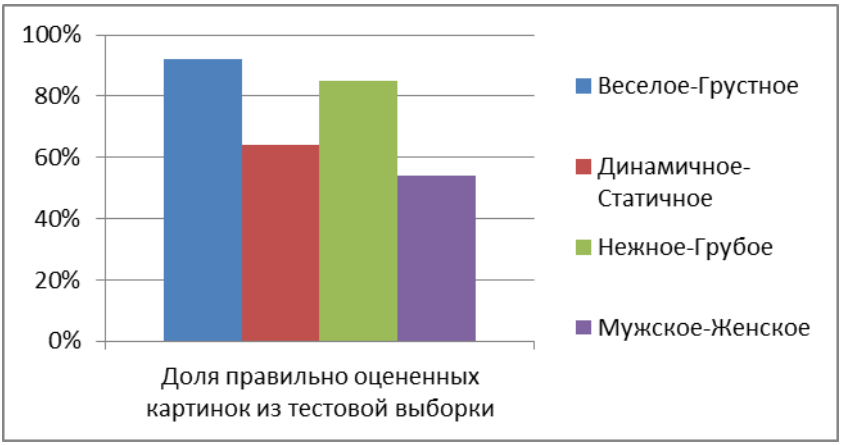
Локальные признаки. Это набор квадратных изображений маленького размера, полученных из исходного изображения. Известно, что локальные признаки могут дать хороший результат в задачах классификации [4].

Обычно локальные признаки это квадратные изображения небольшого размера (15x15, 17x17 или чуть больше). Из одного изображения мы извлекаем целый набор локальных признаков. Обычно их число колеблется от 100 до 1000. Какие именно области выделять обычно определяется дисперсией по яркости, то есть мы выделяем те локальные признаки, которые имеют наибольшую дисперсию (это связано с тем, что участки изображения с наибольшей дисперсией содержат наибольшую информативность).

Результаты

В ходе решения поставленной задачи был проведен анализ нескольких классов признаков: характеристики, связанные с анализом распределения отдельных цветов, яркости и композиции; низкоуровневые статистические признаки; моменты высоких порядков; текстурные и локальные признаки. В результате исследования корреляции между признаками были выделены группы признаков, наиболее тесно связанные с индикаторами, используемыми для качественной оценки восприятия заданного изображения.

На основе построенного пространства признаков был обучен нейронный классификатор, состоящий из карты Кохонена и радиально-базисной нейронной сети. Для тестирования алгоритма классификации использовалась база, составленная из 5540 аннотированных изображений, размещённых на ресурсе http://flickr.com.



Список источников

1. Воронцов К.В. Математические методы обучения по прецедентам // Электронный ресурс, режим доступа – URL: http://www.machinelearning.ru/wiki/images/6/6d/Voron-ML-1.pdf, дата обращения: 24.02.2012

2. Sangjin Lee, Jonghun Park Topic based photo set retrieval using user annotated tags // Multimedia Tools and Applications, 2011.

3. Neela Sawant, Jia Li, James Ze Wang Automatic image semantic interpretation using social action and tagging data // Multimedia Tools and Applications - MTA , vol. 51, no. 1, pp. 213-246, 2011.

4. Datta R, Li J, Wang J Z. Content-based Image Retrieval – Approaches and Trends of the New Age // ACM Intl. Workshop on Multimedia Information Retrieval, ACM Multimedia. Singapore; 2005.

5. H. Tamura, S. Mori, T. Yamawaki. Textural Features Corresponding to Visual Perception // IEEE Transaction on Systems, Man, and Cybernetcs, Vol. SMC-8, No. 6, June 1978. pages 460–472.

6. R. Paredes, J. Perez-Cortes, A. Juan, E. Vidal. Local Representations and a Direct Voting Scheme for Face Recognition // Workshop on Pattern Recognition in Information Systems, Set´ubal, Portugal, July 2001. pages 71–79.

7. Вапник В.Н., Червоненкис А.Я. Теория распознавания образов. – М.: «Наука», 1974.

8. Хайкин С. Нейронные сети. Полный курс. 2-ое издание. - М.: Издательский дом «Вильямс», 2006.

9. Гонсалес Р., Вудс Р. Цифровая обработка изображений. - М.: Издательство «Техносфера», 2005 г. - 1072 с.

10. Васильева Н., Новиков Б. Построение соответствий между низкоуровневыми характеристиками и семантикой статических изображений // Труды 7-ой Всероссийской научной конференции «Эллектронные библиотеки: перспективные методы и технологии, электронные коллекции» - RCDL’2005, – Ярославль, Россия, 2005.

11. Васильева Н., Марков И., Синтез цветовых и текстурных признаков при поиске изображений по содержанию // Санкт-Петербургский Государственный Университет на РОМИП’2008.

12. Zhang M.-L., Zhou Z.-H. A k-nearest neighbor based algorithm for multi-label classification // Proceedings of the 1st IEEE International Conference on Granular Computing (GrC'05). Beijing, China, 2005. pages 718-721.

13. Borgne H., Guerin-Dugue A., Antoniadis A. Representation of images for classification with independent features // Pattern Recognition Letters. vol. 25, 2004. pages 141-154.

14. J. Li and J. Z. Wang, Automatic linguistic indexing of pictures by a statistical modeling approach // IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 25, no. 9, 2003. pages 1075–1088.

15. M. K. Hu Visual pattern recognition by moment invariants, IRE Trans. on Information Theory, vol. 8, pp.179-187, 1962

Модель системы коллаборативного рейтингования событий

Е. Щербакова

Yekaterina_Shcherbakova@rambler.ru

Уральский федеральный университет имени первого Президента России Б. Н. Ельцина, Екатеринбург, Россия
Институт математики и компьютерных наук

Аннотация. В данном документе описывается формальная модель системы коллаборативного рейтингования событий. Вводится система рейтингов для пользователей и событий, описываются зависимости, использующиеся для пересчета рейтингов. Приводятся несколько случаев возможного применения данной модели, а также возможные динамики развития системы.

Ключевые слова: рекомендательные системы; коллаборативная фильтрация; рейтингование; фильтрация содержимого.

Введение

Рассмотрим неформальную постановку задачи на примере университской системы и задачи структуризации информационного (событийного) пространства студентов.

Пользователи нашей системы в этом примере — это студенты, преподаватели, другие сотрудники вуза, в том числе внештатные сотрудники, например, представители сторонних компаний, организующие культурные или научные мероприятия в университете. У каждого пользователя есть возможность создать событие и опубликовать информацию о нем в системе.

Игнатов Д. И., Яворский Р. Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16–18 марта, 2012

После публикации пользователи начинают голосовать за событие: за корректность информации — до проведения мероприятия, за качество события — после проведения. На основе результата голосования формируется набор рейтингов для данного события и пересчитывается соответствующий этому типу мероприятия рейтинг у пользователя. Вклады пользователей в итоговый рейтинг события будут не одинаковыми, а пропорциональными их собственным рейтингам, отражающим статусы полномочий пользователей, рейтинги опубликованных ими ранее событий и отклонения мнений пользователей от итоговых результатов голосований.

Цель: сделать информационную систему, которая бы умела составлять для каждого пользователя отранжированный список подходящих для него событий на основе имеющейся информации о событиях, их рейтингах и данных пользователя.

Математическая модель системы

Систему коллаборативного рейтингования событий мы будем моделировать с помощью двудольного мультиграфа

$$G = \{V, T; A, B, R; \pi, \theta\},$$

где

- V — множество пользователей сети, вершины графа, принадлежащие первой доле,
- T — множество событий, созданных в системе, вершины графа, принадлежащие второй доле,
- $A = \{(v_i, t_j) \mid v_i \in V,\ t_j \in T\}$ задает попарное отношение между событием и его автором,
- $B = \{(v_i, t_j) \mid v_i \in V,\ t_j \in T\}$ показывает, что пользователь и событие связаны попарным отношением “быть в целевой аудитории данного мероприятия”,
  Пример: Пусть в системе было опубликовано событие — лекция по общей алгебре для 1 курса Мат-Меха. Целевой аудиторией данного события, в рамках нашей системы, являются все пользователи из V, у которых в профиле указано, что они студенты 1 курса Мат-Меха.
- $r\colon V \times T \to R^n$ ставит в соответствие каждой паре (пользователь, событие) кортеж из n оценок, данных пользователем событию, согласно n критериям,
  Примечание: Если пользователь v_i не проголосовал за событие t_j, то $r(v_i, t_j) = (0, ..., 0)$.

- $R = \{(v_i, t_j) \mid v_i \in V,\ t_j \in T,\ r(v_i, t_j) \neq (0, ..., 0)\}$ задает попарное отношение между событием и оценившим его пользователем, рецензентом,
- $\pi\colon\ V \to \Pi$ — функция профилей пользователей, которая содержит персональную информацию об участниках сети,
- $\theta\colon\ T \to \Theta$ содержит параметры событий, такие как тип, название, целевая аудитория, место проведения и т. д.

Рейтинги

Важнейшей переменной частью профилей пользователей и информации о событиях является набор рейтингов. В примере, описанном во введении, и у пользователей и у событий имеется три типа рейтингов: рейтинги доверия, качества и эмоциональные рейтинги.

Рейтинги события формируются исходя из оценок, данных пользователями по трем критериям: корректность события, т. е. правдивость опубликованной информации, качество и эмоциональная оценка, данная после проведения мероприятия. Рейтинги событий — это часть информации о событии из T, которую отражает функция θ.

Рейтинги пользователей формируются в зависимости от рейтингов опубликованных ими событий и того, как их личные оценки других мероприятий отклонялись от средних значений. Рейтинги пользователей — это часть профилей пользователей из V, которую отражает функция π.

Классификация событий. Все события, опубликованные в системе, делятся на несколько классов в зависимости от типа мероприятия. Например, в университетской системе можно выделить следующие классы событий: пары, спецкурсы, внеплановые учебные мероприятия (семинары, конференции, научно-популярные лекции и пр.), культурные мероприятия (спектакли в студенческом театре, КВН, спортивные соревнования). Другой пример, если система коллаборативного рейтингования событий создается для какой-то компании, то возможно выделение следующих классов: совещания, деловые встречи, корпоративные праздники, мероприятия для повышения квалификации (семинары от ведущих специалистов, конференции и пр.).

Типизация мероприятий будет использоваться для классификации рейтингов в системе.

Уровень полномочий для пользователей. Уровень полномочий определяется только статусом пользователя. Вернемся к нашему примеру — университетская система, выделим следующие статусы пользователей: студент, преподаватель, представитель руководства, организатор и лаборант. Каждому из этих статусов будет соответствовать свой уровень

полномочий.

$$L_{auth} = F(s),$$

где s — статус пользователя.

Рейтинги доверия для пользователей. У каждого пользователя столько рейтингов доверия, сколько в системе выделено типов событий. Стартовым капиталом всех рейтингов доверия для пользователей является уровень полномочий. Высокий рейтинг доверия говорит о том, что пользователь публиковал в системе достоверную информацию о событиях определенного типа, низкий — недостоверную.

$$R^u_{conf_i} = F(\sigma_i, R^e_{conf_1}, ..., R^e_{conf_k}, L_{auth}),$$

где

σ_i — характеристика разброса, отклонения оценок правдивости событий i-ого типа, данных пользователем, от итоговых средних значений (дисперсия),

k — количество событий, опубликованных в системе данным пользователем, соответствующих i-ому типу,

$R^e_{conf_1}, ..., R^e_{conf_k}$ — рейтинги доверия опубликованных событий i-ого типа,

L_{auth} — уровень полномочий пользователя.

Качественный рейтинг для пользователей. Качественный рейтинг присутствует только у тех пользователей, которые участвовали в организации мероприятий, опубликованных в системе. Данный рейтинг будет высоким, если пользователь организовывал качественные, по мнению участников, мероприятия.

$$R^u_{qual} = F(R^e_{qual_1}, ..., R^e_{qual_p}),$$

где

p — количество мероприятий, организованных данным пользователем,

$R^e_{qual_1}, ..., R^e_{qual_p}$ — качественные рейтинги организованных мероприятий.

Эмоциональный рейтинг для пользователей. Аналогично качественному, эмоциональный рейтинг присутствует только у тех пользователей, которые участвовали в организации мероприятий, опубликованных в системе. Эмоциональный рейтинг пользователя будет низким, если участникам мероприятие не понравилось, высоким, если понравилось.

$$R^u_{emot} = F(R^e_{emot_1}, ..., R^e_{emot_p}),$$

где

p — количество мероприятий, организованных данным пользователем,
$R^e_{emot_1}, ..., R^e_{emot_p}$ — эмоциональные рейтинги, организованных мероприятий.

Рейтинг доверия для событий. Данная характеристика должна отражать правдивость опубликованной о событии информации.

$$R^e_{conf} = F(R^a_{conf}, (a, R^u_{conf}, L_{auth}, t_a)_1, ..., (a, R^u_{conf}, L_{auth}, t_a)_n),$$

где

R^a_{conf} — рейтинг доверия к автору события,
a — оценка, данная пользователем (“correct” или “incorrect”),
R^u_{conf} — рейтинг доверия пользователя, соответствующий данному типу мероприятия,
L_{auth} — уровень полномочий пользователя,
t_a — принадлежность к целевой аудитории проголосовавшего пользователя (1 — принадлежит, 0 — не принадлежит),
n — количество рецензентов, проголосовавших за событие.

Качественный рейтинг для событий. Качественный рейтинг события должен отражать уровень проведенного мероприятия.

$$R^e_{qual} = F((a, R^u_{conf}, L_{auth}, t_a)_1, ..., (a, R^u_{conf}, L_{auth}, t_a)_m),$$

где

a — качественная оценка, данная пользователем,
R^u_{conf} — рейтинг доверия пользователя,
L_{auth} — рейтинг полномочий пользователя,
t_a — принадлежность к целевой аудитории проголосовавшего пользователя (1 — принадлежит, 0 — не принадлежит),
m — количество рецензентов, оценивших качество мероприятия.

Участники системы оценивают качество мероприятия после проведения мероприятия.

Эмоциональный рейтинг для событий. Эмоциональный рейтинг строится на базе субъективных оценок пользователей («мне нравится» или «мне не нравится»).

$$R^e_{emot} = F((a, t_a)_1, ..., (a, t_a)_l),$$

где

a — эмоциональная оценка, данная пользователем,

t_a — принадлежность к целевой аудитории проголосовавшего пользователя (1 — принадлежит, 0 — не принадлежит),

l — количество рецензентов, давших эмоциональную оценку мероприятию.

Пользователи дают эмоциональную оценку событию как до, так и после проведения.

Начальные значения рейтингов. Стартовым капиталом доверия является уровень полномочий. Уровень полномочий пользователя L_{auth} определяется сразу после регистрации в системе, исходя из заполнения обязательного поля «ваш статус» и проверки достоверности введенной информации. Т. о. изначально рейтинг доверия пользователя зависит только от уровня полномочий для всех типов событий.

$$R^u_{conf} = F(L_{auth})$$

Качественный и эмоциональный рейтинги пользователей изначально равны нулю.

Для рейтингов событий — аналогично. Если за событие не проголосовал ни один пользователь, то его рейтинг доверия зависит только от рейтинга доверия автора.

$$R^e_{conf} = F(R^a_{conf})$$

Эмоциональный и качественный рейтинги для событий изначально также равны нулю.

Динамика в системах коллаборативного рейтингования событий

Отметим, что может меняться в системах коллаборативного рейтингования событий с течением времени, т. е. выделим основные направления динамики систем:

- добавление новых пользователей в V;
- публикация новых событий, т. е. расширение множества T;
- изменение целевой аудитории событий, модификация соответствующих пар в B;
- изменение рейтингов событий и пользователей на основе новых оценок этих событий, данных пользователями, изменение функций π и θ, соответственно, в описанной модели;
- изменение профилей участников сети, модификация функции π;
- изменение информации о событии, модификация функции θ.

Список источников

1. Бацын М. В., Калягин В. А. Об аксиоматических определениях общих индексов влияния в задаче голосования с квотой. М.: Издательский дом Государственного университета — Высшей школы экономики, 2009.
2. Николенко С. И., Сироткин А. В. Рейтинг-системы с точки зрения байесовского вывода. // Труды конференции “Интегрированные модели, мягкие вычисления, вероятностные системы и комплексы программ в искусственном интеллекте” (ИММВИИ-2009). В 2-х тт. Т. 2. М.: Физматлит, 2009. С. 29–48.
3. Мулен Э. Кооперативное принятие решений: Аксиомы и модели. М.: Мир, 1991.
4. Шварц Д. А. О вычислении индексов влияния, учитывающих предпочтения участников. // Автоматика и телемеханика 3, 2009. С. 152–159
5. Easley D., Kleinberg J. Networks, Crowds, and Markets: Reasoning About a Highly Connected World. Cambridge University Press, 2010.
6. Tsvetovat M., Kouznetsov A. Social Network Analysis for Startups. O’Reilly, 2011.
7. Yavorskiy R. Research Challenges of Dynamic Socio-Semantic Networks, 2011. On-line at: http://blog.witology.com/wp-content/uploads/2011/06/RYavorsky-Witology1.pdf.

Методики улучшения качества данных в онлайн исследованиях с помощью нематериальных стимулов мотивации участников access-панелей.

Е.А Соловьёва[1], И.А.Куприянов[2], Ю.С.Ермоленко[3]

e.a.solovjova@gmail.com[1], sociolog.k@gmail.com[2], ysermolenko@gmail.com[3]

НИУ ВШЭ, Москва, Россия

Аннотация. Сейчас перед провайдерами онлайн панелей стоит задача смещения с материального стимулирования активности участников на нематериальное. Основной целью в нашей работы - поиск наиболее подходящих методик нематериального стимулирования для разных групп респондентов по типам мотивации. На основе полученных нами результатов можно будет разработать методы улучшения менеджмента панелей, организации общения с панелистами, разработать новые нематериальные методы стимулирования их активности**.**

Ключевые слова: онлайн исследования, мотивация, нематериальное стимулирование, access-панели, рекрутирование участников панелей, качество данных онлайн исследований

Введение

Одним из основных дискуссионных вопросов остается качество данных, собираемых с помощью онлайн панелей (access panel)[1] респон-

[1] **Онлайн панели** представляют собой сообщества людей, давших согласие на регулярное участие в маркетинговых онлайн исследованиях. Каждый участник сознательно регистри-

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16-18 марта, 2012.

дентов. В 2005 году компания Virtual Surveys зафиксировала, что более чем 75% панелистов являются участниками более 3х панелей, а в 2006 году компания Comscore Networks на конференции CASRO привела результаты, что более 30% исследований в интернете проводятся на основе опроса всего 1% населения [1].

Перед организаторами онлайн панелей и онлайн исследователями, стоит задача улучшения качества данных. Один из способов сделать это - повышение заинтересованности и вовлеченности панелистов. Но с одной стороны проблема заключается в том, что активность участников в онлайн панелях в первую очередь поддерживается материальными стимулами, с другой в том, что личное присутствие и контроль со стороны интервьюера в онлайн среде отсутствует. В совокупности это приводит к образованию группы «профессиональных респондентов» и существенному снижению качества получаемых данных[1].

Если обратиться к результатам исследований относительно мотивации панелистов, мы увидим, что ситуация менее однозначна как может показаться на первый взгляд. Желание заработать на этом деньги вовсе не является главным мотивом для участия в панелях. Так, в опросе панели на Anketka.ru [2] в июле 2009 г. причину участия в опросах «Я хочу заработать деньги» указали всего 16% панелистов, а о своей заинтересованности сказали 28%. Таким образом, денежное стимулирование не всегда лучший способ повысить активность респондентов. А поэтому следует также обратиться к разработке иных нематериальных способов стимулирования.

Но дело еще и в том, что сами респонденты не представляют собой однородную массу. По результатам проведенных исследований среди панелистов присутствуют разные типы с отличающимися мотивационными схемами. В исследовании на Anketka.ru (см [2]) по результатам опроса были выделены три типа участников панели: 1) «любопытные»; 2) «ответственные»; 3) «прагматичные». Первые участвуют в опросе ради интереса или развлечения, вторые заинтересованы в практическом применении результатов исследования, третьи - участвуют ради определенных выгод. И это не единственная классификация. Так в 2010 году компанией Tiburon Research было проведено исследование мотивации

руется в панели на специальном Интернет-портале, предоставляя о себе различные социально-демографические и потребительские данные, а также получает компенсацию за участие в виде денежного вознаграждения или призов. http://www.omirussia.ru/ru/online_panels/

[1] 1 Основные проблемы связанные с недобросовестными, «прагматиками»: 1. Восприятие всех панелей, а также самой практики участия в опросах с точки зрения возможного заработка, 2. Множественные регистрации в панелях, 3. Некорректные данные, оставляемые при регистрации и заполнении анкет.1 Их основная задача – заработать больше при минимальных затратах.

панелистов из InternetOpros.ru [3]. Выявленные группы (профи, убить время, любопытные, азартные, небезразличные) были проанализированы и были даны рекомендации по улучшению качества данных, исходя из их частных особенностей.(см [3]). По итогам можно сделать заключение, что первым делом нужно «переманивать» остальные группы в «небезразличных», так как качество данных при этом оказывается самое высокое.

Таким образом, основной задачей в нашей работе будет исследование группы панелистов с целью их классификации и выявления «предпочтений» в вопросе участия в исследованиях. На практике с помощью результатов нашего исследования можно будет сократить отрицательные и «отталкивающие» недоработки в процессе рекрутирования и взаимодействия с панелистами. Так как необходимо не только «зацепить» их, но и удержать лучших, повысить лояльность – заинтересованность [4]. Для начала рассмотрим более подробно использующиеся в российских онлайн панелях методы стимулирования респондентов при рекрутировании.

Обзор практик рекрутирования панелистов в российских онлайн панелях. (См. Приложение 1)

Различные компании и web-сервисы вербуют панелистов чаще одним способом мотивирования – денежным вознаграждением. Помимо денежного перевода, также есть возможность участия в розыгрыше призов или зачисление накопленных средств на счет мобильного телефона («Комкон» или «GlobalTestMarket») . Однако некоторые компании предпочитают дополнять основной мотив участия в панелях другими «смыслами», которые также способствуют мотивации панелиста. Дополнительная мотивация – это возможность получить информацию о результатах исследований, или же опрос преподносится как способ выразить свое мнение и т.д. Возможность перевода средств на благотворительность предлагается панелями «Важное мнение», «Анкетка», «Avto opros».

Совершенно иной подход в мотивировании – это дать понять панелисту, что его мнение важно, что он будет «услышан», что он может «делиться своим мнением». Также опрос может быть представлен панелисту как способ поучаствовать в улучшении качества товаров и услуг. Такого рода мотив представлен на сайтах «Ask GfK», «Анкетка», «Profi Online Research», «Avto Opros» и т.д. Еще одна важная мотивация – это доступ (как правило, частичный) к результатам исследований. Возможность получить доступ к результатам предоставляют «Ask GfK» и «VoxRu».

Помимо перечисленных, также существуют некоторые специфические методики, но они крайне редки или находятся на стадии разрабо-

ток. Их особенность в том, что их внедрение основано на предварительных глубоких исследованиях, а также рассчитано на специфические группы. Например, для особой группы IT специалистов в панели - Omi (www.omirussia.ru) предлагается : «подписка на специализированные журналы, участие в тренингах или посещение IT конференций по всему миру, в зависимости от количества исследований,в которых они приняли участие.». На этапе ещё требующего разработки предложения пока осталась, например, интересная идея , озвученная на конференции CARSO 2009 [5] Колин Карлин и Шоном Эйдсоном о том, что можно увеличивать размер оплаты, в зависимости от количества потраченного на за заполнение анкеты времени и предварительно уведомлять об этом панелистов. Но в нашей стране пока что большинство панелей ограничиваются в основном материальными стимулами мотивации.

Классификация панелистов на основе пирамиды потребностей А.Маслоу.

Основной исследовательский вопрос в нашей работе: как повысить мотивацию участников панелей нематериальными стимулами. Более частный исследовательский вопрос, исходящий из основного и задающий направление дальнейшей работе над исследованием - каковы основные мотивы участия в опросах, у разных типов респондентов, иными словами, каким образом необходимо мотивировать их нематериально, исходя из их же потребностей.

Если рассматривать панелистов, как «наемных работников» (регулярно участвующих в исследованиях и получающих денежное или иное вознаграждение), чью мотивацию и качество работы необходимо повысить, то в разработке нематериальных стимулов можно опереться на результаты давно существующей и развитой области социологических исследований – социологии труда. Вопрос повышения трудовой мотивации хорошо разработан и в сфере социологии труда и в области менеджмента. Наиболее известные теории мотивации были предложены такими учеными, как А. Маслоу, Ф. Герцберг, Д. Мак-Грегор, К. Альдерфер, Д. Мак-Клелланд. Опустив некоторые моменты, можно сказать, что на сегодняшний день большинство исследователей сходятся на том, что потребности и мотивы лежат в основе действий человека, и для эффективного управления необходимо оказывать влияние на мотивы. Основная задача технологии мотивации - формировать «определенную мотивационную структуру кадрового потенциала, развивая и усиливая желательные для субъекта мотивирования мотивы и ослабляя те, которые мешают эффективному управлению.» [6] . Мы предполагаем, что все методы стимулирования также по-разному действуют на участников

панелей[1], в зависимости от их структуры мотивации. Основной целью в нашей работе мы ставим создание системы нематериальной мотивации для рекрутируемых участников онлайн панелей на основе иерархии потребностей А.Маслоу и классификации респондентов по мотивационным типам.

Предполагается провести исследование группы панелистов, как объекта исследования с целью их классификации и выявления «предпочтений» в вопросе участия в исследованиях. Для этого проведем количественный онлайн опрос среди участников панели Anketka.ru, с применением различных методик выявления отношения к онлайн опросам и методам стимулирования мотивации.(См.Приложение 1).

Основная гипотеза нашего исследования - существование различний среди панелистов по типам мотивации на основе иерархии потребностей А.Маслоу, и основной метод – кластерного анализа. Задействуем вопрос, диагностирующий предрасположенность к различным потребностям, по методике И.А. Акиндиновой7 (См. Приложение 1, вопрос 3)[2]. Чтобы получить показатель наиболее доминирующей потребности у респондента респонденту предлагается сравненить между собой 15ть утверждений (1) Я хочу добиться признания и уважения, 2) Я хочу иметь теплые отношения с людьми и т.д.), и выставлять «более важному» баллы при каждом сравнении. После подсчета выявляется, к какой шкале относятся набравшие наибольшее количество баллов выражения (I шкала - Материальное положение, II шкала - Потребность в безопасности, III шкала - Потребность в межличностных связях и т.д.) Градация идет по следующим числовым отрезкам: Полная удовлетворенность – 0-13 баллов, Частичная удовлетворенность 13-26 баллов, Полная неудовлетворенность – 26-39 баллов.

Так как кластерный анализ можно использовать только на интервальной или дихотомической шкале, надо будет перевести этот вопрос в дихотомическую шкалу на этапе анализа данных. Каждый вариант ответа будет представлен как отдельный вопрос с вариантами ответов «да» и «нет». Каждая шкала будет представлена отдельным вопросом, напри-

---

[1] Для того, чтобы продолжить тему мотивации, необходимо условиться об определениях основных понятий:

Потребность — испытываемая нужда в чем-то необходимом.

Мотив — то, как потребность субъективно преломляется в человеке, подталкивая его к действию.

Внутренняя мотивация — внутренняя структура индивидуальных мотивов, влияющая на действия человека.

Мотиватор — фактор удовлетворенности трудом, влияющий на его эффективность

[2] Мы полагаем, что при возможности необходимо использовать более глубокие методики психологического диагностирования мотивов. Но в нашем случае, мы такой возможностью не обладаем2, но, тем не менее, данная методика позволит получить показатель наиболее доминирующей потребности у респондента.

мер: «Вам важна потребность в безопасности? «. Число возможных баллов колеблется от 0 до 39, то есть всего 40 - делим пополам. Если при подсчете баллов «потребность в безопасности» наберет от 0 до 19 – это будет рассматриваться как «нет», если от 20 до 39 –то как «да».

Помимо ранжирования респондентов по потребностям, попросим их самих составить иерархию своих мотивов вступления в панель (см. приложение 1, вопрос 5): (1) Мне нравится выражать свое мнение, 2) Я хочу участвовать в благотворительности, 3) Я хочу заработать деньги и т.д.). Этот вопрос тоже будет переделан в дихотомическую шкалу, но в этом случае в качестве «нет» будут рассматриваться ответы «0» и «1», а в качестве «да» будут - ответы «2» и «3». После соответствующей подготовки данных эти два вопроса (показатель наиболее доминирующей потребности и доминирующий мотив участия в панели) будут положены в основу построения классификация респондентов с помощью кластерного анализа в программе анализа данных SPSS.

Так мы решим задачу построения типологии, и далее встанет вопрос подробного описания полученных типов респондентов. Для начала нужно исследовать наиболее важные мотивы участия в панелях и степени их удовлетворенности. Для этого используем результаты вопросов 6 и 7 (См. приложение 1) Вопросы 6 и 7 состоят из одинаковых утверждений (в 6ом вопросе нужно указать, степень их важности, в 7ом – степень удовлетворенности), этот список является одновременно и списком мотивов к участию и списком основных используемых при рекрутировании панелистов стимулов, (так как стимулы создавались рекрутерами панелей с расчетом удовлетворения определенных потребностей респондентов). Этот список был составлен нами в результате обзора сайтов 13ти российских онлайн панелей. Для наглядного представления о значимых и проблемных областях, построим график, со следующей структурой (см.рис.1).

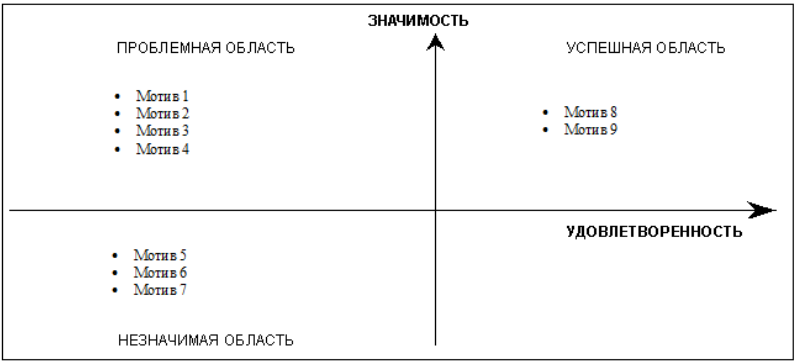


Рис. 1. График взаимозависимости значимости и удовлетворенностью мотивами участия в access-панелях.

Мотивы будут распределены по областям, в зависимости от значения соответствующего показателя (значимости и удовлетворенности) по каждому из них. Оба вопроса предполагают ответ со шкалой от 0 до 5 , что в нашем случае можно представить, как шкалу от -2 до 2 с сохранением обозначений («-2» = «0» , «-1»=1 и т.д.) Анализ проблемных областей будет положен в основу дальнейших рекомендаций по повышению мотивации и улучшению взаимодействия с разными типами панелистов.

Также хотелось бы изучить, как будут проявляться заинтересованность в самом опросе или в получаемых от участия выгодах в сравнении с качеством заполнения анкеты. Для этого вернемся к вопросу 6 (важность стимулов участия в панели). Имеем список утверждений (Мне важно: знать результаты исследований, накопление баллов, деньги, розыгрыш призов и т.д.). Их достаточное количество, чтобы провести факторный анализ, переведя их в дихотомическую шкалу. Ожидаем, что в результате анализа получим два типа факторов, и основываем наше предположение на теории Герцберга [8], о том, что существуют два типа факторов, влияющих на мотивацию к труду.[1] Для нашего исследования их можно будет интерпретировать как «интерес» (мотиваторы) и «удобство» (гигиенические факторы). Фактор «интерес» будет тем выше, чем больше человек действительно заинтересован в самом исследовании, а не сопутствующих выгодах. Создадим переменную для обоих факторов по степени выраженности (Rank Cases) и соотнесем с дополнительными данными. Так как мы будем располагать некоторыми дополнительными данными по социально – демографическим характеристикам (по полу, возрасту, уровню дохода (субъективного и на члена семьи), уровню образования, о семейном положении, городе проживания, наличию, например, автомобиля, и т.д.), и данными по качеству заполнения анкеты, которые соберем с помощью нескольких проверочных вопросов (1 и 10 – измеряют «настроение», насколько респонденту нравится деятельность по заполнению анкеты вообще, 2 и 9 – вопросы-ловушки, они представляют собой противоречивые утверждения, и расположены в разных концах анкеты, так что респонденту в случае недобросовестного ответа будет сложно себя проконтролировать). Эти переменные можно использовать для того, чтобы устанавливать связи для факторных значений.

[1] а) Мотиваторы, побуждающие к улучшению результатов деятельности (достижение целей, признание заслуг, предоставление самостоятельности, обогащение труда элементами творчества, возможности личной самореализации.) б) Гигиенические факторы - их отсутствие или ухудшение порождает неудовлетворенность (условия труда, вознаграждение, межличностные отношения, гарантия сохранения работы, стиль руководства). Бакурадзе А. Б. Факторы-мотиваторы. Что может почерпнуть администратор из теории Ф. Герцберга

Выводы

На основе полученных выводов сможем наметить, каковы черты наиболее активных и заинтересованных панелистов, которых и нужно больше привлекать. И также узнаем «проблемные» места менее активных и заинтересованных панелистов, чтобы их решить и в будущем расширить круг привлекаемых участников.

Наша работа только первый шаг к дальнейшей работе по данной проблеме – повышения мотивации нематериальными стимулами. Необходим более глубокий анализ потребностей с помощью психологических методик, необходимо иметь данные по качеству заполнения анкет участниками. Нужно разработать портрет «идеального» панелиста, и при первичном рекрутировании разделять респондентов на типы.

На данный момент при внедрении техник стимулирования мотивации мониторинг результатов проводится немногими отечественными провайдерами. Большинство провайдеров просто заимствуют западные разработки и способы нематериальной мотивации. Но данное заимствование должно быть проанализировано и осознанно с точки зрения применимости в нашей стране.

Но главное, отечественные исследователи и провайдеры должны придти к осознанию, того что панелисты – это особое «сообщество», со своими потребностями и «обязанностями», и требуют пристального и постоянного измерения и изучения. Необходимо выстраивать стратегии повышения и удержания лояльности [9] и основой взаимодействия должны быть долгосрочные и взаимовыгодные отношения между организаторами и участниками access-панелей.

Приложение 1

Обзор методов стимулирования в российских онлайн панелях

№	Панель	Сайт	Методы стимулирования
1	Ask GfK	http://ru.askgfk.com/	- Доступ к результатам исследований (быть в курсе); - Накопление баллов, перевод баллов на мобильный телефон, в денежный эквивалент на Web-money, Яндекс.Деньги; - Возможность делиться опы-

№	Панель	Сайт	Методы стимулирования
			том, высказывать мнение по продуктам.
2	GlobalTestMarket	https://www.globaltestmarket.com/	- Накопление баллов, перевод баллов в денежный эквивалент или на розыгрыш денежных призов лотереи
3	Важное мнение	http://www.vazhnoemnenie.ru/	- Денежное вознаграждение - Приобретение подарочного сертификата - Благотворительность
4	VoxRu	http://voxru.net	- Розыгрыш призов - Ознакомление с комментариями по пройденным опросам - Узнать распределение ответов респондентов по демографическим и социальным характеристикам
5	Комкон	http://www.interneto pros.ru/	- Денежное вознаграждение - Розыгрыш призов - Перевод денег на счет мобильного телефона
6	Анкетка	http://www.anketka.ru/	- Денежное вознаграждение - Высказать свое мнение - Участие в разработке новых продуктов ведущих мировых производителей - Знания о новых тенденциях огромного мира товаров и услуг - Благотворительность
7	Profi Online Research	http://profiresearch.net/ru/reg/	- Денежное вознаграждение - Розыгрыши призов - Участие в улучшении качества товаров и услуг
8	Oneline52	http://www.online52.ru/	- Денежное вознаграждение - Выбор опроса по интересу
9	Mysurvey	http://ru.mysurvey.com	- Накопление баллов, перевод в денежную валюту
10	Avto Opros	http://www.avtoopros.ru/	- Денежное вознаграждение - Возможность принимать

№	Панель	Сайт	Методы стимулирования
			участие в разработке новых автомобилей и авто-аксессуаров - Участвовать в опросе – значит быть в курсе новых тенденций на рынке. – Благотворительность
11	Planet Panel	http://www.planetpanel.net/	- Розыгрыш призов - Участие в улучшении качества товаров и услуг
12	Russian Information Network	http://www.rin.ru/panel/	- Денежное вознаграждение - Участие в улучшении качества товаров и услуг
13	Omi	http://www.omirussia.ru	Для специалистов - подписка на специализированные журналы, участие в тренингах или посещение IT конференций по всему миру, в зависимости от количества исследований, в которых они приняли участие.

Приложение 2

Анкета: Удовлетворенность участников социологических опросов качеством сервиса онлайн панелей.

1. Приветствие.

Добрый день! Вас приветствует исследовательская команда факультета социологии Национального Исследовательского Университета - Высшей школы экономики. Тема нашей работы «Удовлетворенность участников социологических опросов качеством сервиса онлайн панелей». Надеемся, что вам будет интересно. Опрос проводится анонимно и при желании вы сможете ознакомиться с результатами исследования.

Желаем удачи!

2.Вводная часть.

1) Какое у вас сейчас настроение?
   a) ужасное
   b) очень плохое
   c) плохое
   d) нормальное

e) хорошее
f) очень хорошее
g) замечательное

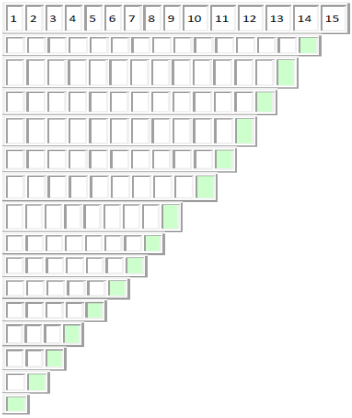
2) Когда я не вижу определенных выгод для себя, я не участвую в социологических опросах.
   a) Да
   b) Нет

3. Диагностика мотивации по А.Маслоу

«Иерархия потребностей» модификация И.А. Акиндиновой.

3) **«Я хочу...»**
   1. Добиться признания и уважения
   2. Иметь теплые отношения с людьми
   3. Обеспечить себе будущее
   4. Зарабатывать на жизнь
   5. Иметь хороших собеседников
   6. Упрочить свое положение
   7. Развивать свои силы и способности
   8. Обеспечить себе материальный комфорт
   9. Повышать уровень мастерства и компетентности
   10. Избегать неприятностей
   11. Стремиться к новому и неизведанному
   12. Обеспечить себе положение влияния
   13. Покупать хорошие вещи
   14. Заниматься делом, требующим полной отдачи
   15. Быть понятым другими.

Бланк ответов



4. Диагностика мотивации участия в социологических опросах.

4) Если участие в опросе не подразумевает денежного вознаграждения, какие еще могут быть причины для участия?
(открытый вопрос, обязательный).
5) Оцените, пожалуйста, в порядке возрастания значимости для вас следующие суждения. 0 до 3 (где «0» - совсем не важно, «1» - не очень важно, «2» - важно, «3» - очень важно).
   a) Мне нравится выражать свое мнение
   b) Я хочу участвовать в благотворительности
   c) Я хочу заработать деньги
   d) Я хочу, чтобы производители считались с моим мнением
   e) Мне интересно принимать участие в опросах
   f) Мне нравится узнавать о новинках на рынке и результатах исследований в науке
6) Оцените, пожалуйста, насколько важны для вас различные возможности, предоставляемые участникам онлайн опросов, расставив оценки от 0 до 4 (где «0» - совсем не важно, «1» - не очень важно, «2» - средне важно, «3» - важно, «4»- очень важно).
   a) Знать результаты исследований
   b) Накопление баллов
   c) Деньги
   d) Розыгрыш призов
   e) Высказать свое мнение по новым продуктам /социальным проблемам
   f) Участие в благотворительности
   g) Участие в разработке новых продуктов/ решении социальных проблем
   h) Выбор опроса по интересу
   i) Возможность обучения и развития
   j) Новые социальные коммуникации
   k) Возможность самореализации.

5. Удовлетворенность качеством опросов.

7) Оцените, пожалуйста, насколько вы удовлетворены качеством возможностей, предоставляемых участникам онлайн опросов, расставив оценки от 0 до 4 (где «0» - совсем не удовлетворен, «1» - не удовлетворен, «2» - частично удовлетворен, «3» - удовлетворен, «4»- полностью удовлетворен).
   a) Знать результаты исследований
   b) Накопление баллов

c) Деньги
d) Розыгрыш призов
e) Высказать свое мнение по новым продуктам /социальным проблемам
f) Участие в благотворительности
g) Участие в разработке новых продуктов/ решении социальных проблем
h) Выбор опроса по интересу
i) Возможность обучения и развития
j) Новые социальные коммуникации
k) Возможность самореализации.

8) Оцените сервис онлайн панелей по следующим характеристикам, расставив оценки от 0 до 4 (где «0» - совсем не удовлетворен, «1» - не удовлетворен, «2» - частично удовлетворен, «3» - удовлетворен, «4»- полностью удовлетворен).
   a) Этап отбора (скрининг)
   b) Размер оплаты
   c) Удобство получения оплаты
   d) Интересная тема
   e) Общение с другими участниками опросов
   f) Взаимодействие с администраторами
   g) Внешнее представление и удобство навигации на странице панели
   h) Внешнее представление и удобство заполнения анкет
   i) Баланс затрат и выгод
   j) Возможность высказать мнение
   k) Размер анкеты.

6. Заключительная часть.

9) Чтобы вы изменили в сервисе онлайн панелей, если бы были администратором? (открытый вопрос, по желанию)
10) Когда я не вижу определенных выгод для себя, я участвую в социологических опросах.
    a) Да
    b) Нет
11) Какое у вас сейчас настроение?
    a) ужасное
    b) очень плохое
    c) плохое
    d) нормальное
    e) хорошее
    f) очень хорошее

g) замечательное

Список источников

1. Мавлетова А. Борьба за качество и надежность данных в онлайн исследованиях: основные результаты панельной конференции CASRO 2009 . Онлайн исследования в России 2.0 . Под ред. А.В. Шашкина, И.Ф. Девятко, С.Г. Давыдова. М: РИЦ »Северо-Восток», 2010.

2. Шашкин А.В. Влияние заинтересованности панелистов на качество/Онлайн исследования в России 2.0. Под редакцией Шашкина А.В., Девятко И.Ф., Давыдова С.Г.- М.:РИЦ «Северо-восток», 2010

3. Мотивация: сегментация панелистов.Онлайн маркетинговые исследования Tiburon / http://www.4p.ru/main/cnews/147169/

4. Huitt, W. (2007). Maslow's hierarchy of needs. Educational Psychology Interactive. Valdosta, GA: Valdosta State University /http://www.edpsycinteractive.org/topics/regsys/maslow.html .

5. Мавлетова А. Борьба за качество и надежность данных в онлайн исследованиях: основные результаты панельной конференции CASRO 2009 . Онлайн исследования в России 2.0 . Под ред. А.В. Шашкина, И.Ф. Девятко, С.Г. Давыдова. М: РИЦ »Северо-Восток», 2010.

6. Кравец А.А. Основные направления совершенствования технологий стимулирования и мотивации кадрового потенциала лечебно-профилактических учреждений. / ecsocman.hse.ru/data/2011/03/18/1268219327/15.pdf.

7. Акиндиновой И. А. «Иерархия потребностей» / http://testoteka.narod.ru/ms/1/02.html

8 . Бакурадзе А. Б. Факторы-мотиваторы. Что может почерпнуть администратор из теории Ф. Герцберга/ http://ecsocman.hse.ru/rubezh/msg/16754348.html

9. Тимошина А. Love brands: штрихи к портрету. Онлайн исследования в России 2.0. Под редакцией Шашкина А.В., Девятко И.Ф., Давыдова С.Г.- М.:РИЦ «Северо-восток», 2010

Горная ГИС на основе OpenCASCADE

А.Г. Уймин[1], В.И. Суханов[2]

[1]au-mail@ya.ru, [2] cyx-fat@mail.ru

[1]ГБОУ СПО СО Уральский радиотехнический колледж им. А.С. Попова, Екатеринбург, Россия

[2]ФГАОУ ВПО «УрФУ имени первого Президента России Б.Н. Ельцина», Екатеринбург, Россия

Аннотация. В работе рассматривается аспекты разработки ГГИС с открытым кодом, на базе открытого геометрического ядра, предназначенная для горных предприятий с различными способами добычи твёрдых полезных ископаемых.

Ключевые слова: геоинформационные системы, геометрическое ядро, модель карьера, разработка твердых полезных ископаемых.

Введение

Повышение количества пространственно-временных приложений контроля окружающей среды, геологии и мобильной связи – являются новым вызовом в разработке географических баз данных. Современные геоинформационные системы являются замкнутыми системами главным образом поддерживающими географические задачи в пространстве 2D. В прикладных областях, таких как геология, при планировании разработки открытых месторождений, требуется моделирование и обработка 3D/4D объектов.

Основными компаниями производителями программного обеспечения для горной промышленности [1] являются: Gemcom Australia Pty Ltd; GeoExpress 3D integration Software by IGM Ltd;MineMap Pty. Ltd;

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16-18 марта, 2012.

Snowden Group; Vulcan 3D Software by Maptek; CSIRO – Mine Production Control Research; ER Mapper; Fractal Graphics; GeoExpress 3D integration Software by IGM Ltd; Geosoft Inc; LTC Pty Ltd; Micromine Pty Ltd; Mintec Inc; Neural Mining Solutions; Petrosys Pty Ltd; The Soft Earth; Whittle Strategic Mine Planning. Широко известные горно-геологические информационные системы (ГГИС) DATAMINE, VULCAN, MINESCAPE, GEMCOM, TECHBASE успешно применяемые за границей характеризуются низким масштабом распространения в России, главным образом по причинам их довольно высокой стоимости, удаленности разработчиков и трудностью модификации систем под внутренние отраслевые и федеральные стандарты [2]. На территории России получили относительно широкое распространение зарубежные продукты SURPAC, Micromine, к основным недостаткам которых можно отнести не только высокую стоимость, но и необходимость существенной доработки под специфику предприятия (что затруднительно в условиях недостатка справочных и методических материалов). Отечественные системы GeoToolKit, Майнфрэйм Технология+БВР отвечают требованиям для решения задач горно-геологического моделирования, но реализованы как закрытые проприетарные системы с обязательными лицензионными отчислениями поставщикам программного обеспечения, и требуют дополнительных затрат на приобретение платформы развертывания [3]. На отечественном рынке открытых ГГИС в полном объеме закрывающих потребности геологов, маркшейдеров, технологов, взрывников и транспортников не представлено. Открытые ГИС в основном специализируются на просмотре геопространственных данных, поиске требуемых пользователю объектов и анализе геоданных (GRASS).

Возрастающая потребность горных предприятий с различными способами добычи твёрдых полезных ископаемых в разработке, как решений отдельных технологических задач, так и полномасштабных ГГИС для решения прикладных задач на этапах проектирования, эксплуатации, консервации и погашения карьеров позволяет говорить о необходимости разработки современной отечественной ГГИС на базе открытого программного обеспечения. В связи с тем, что основным критерием отбора инструментальных средств является информационная совместимость используемых для разработки ГГИС программных фрагментов и возможность их интеграции в единый комплекс то, на данном этапе развития отрасли требуется разработка отечественной ГГИС, которая будет являться технологической платформой. Технологическая платформа на базе открытых исходных кодов не только объединит наработки в различных прикладных сферах, но и позволит создать гибкую систему, легко адаптируемую под нужды предприятия.

Задачи горно-геометрического анализа

Основной целью горно-геометрического анализа являются расчеты метрических свойств по сортам и видам горной массы в контуре отработки месторождения по горизонтам и периодам времени. Результатом является таблица горно-геологического анализа, в которой указывается состав и объемы горной массы, в том числе по сортам ископаемого для каждого горизонта контура отработки на планируемый период или этап. Эта информация необходима для принятия решения о направлении развития горных работ в карьере.

Решение задач планирования горных работ и оптимизации технологических процессов требует горно-геологической и технологической информации о месторождении, техногенных образованиях, рельефе местности и ситуации, транспортных, энергетических, водоотводных и других коммуникациях. Традиционными средствами для этих целей служат графические документы в виде погоризонтных качественных планов или разрезов месторождения с нанесенными на них контурами карьера. Эти задачи позволяет решать совокупность аппаратных, программных средств и хранимых моделей месторождения, карьера, отвалов, топографии и ситуации называемая геоинформационным обеспечением горного производства или ГГИС.

Обзор инструментальных средств

Системы компьютерной графики – сложные программные комплексы. Они включают СУБД, среду для разработки приложений для выбранного языка программирования, подсистему машинной графики, интерфейсы для разработчиков и пользователей, тестирования, документирования, установки, конфигурирования и многое другое. Наиболее сложным компонентом является подсистема машинной графики. Большинство промышленных САПР и ГИС основываются на использовании готовых геометрических ядер. Например, AutoCAD использует ядра ACIS, ShapeManager, SolidWorks — Parasolid, CADKEY – ACIS. Геометрическое ядро решает основные геометрические задачи, например, построение поверхности сопряжения, булевы операция между телами и другие. С настоящее время можно выделить:

- лицензируемые ядра геометрического моделирования, которые разработаны и поддерживаются одной компанией, которая лицензирует их другим компаниям для их CAD-систем. Например, ядро Parasolid, разработано UGS. Оно используется в Unigraphics и Solid Edge и лицензировано другим компаниям, включая CADMAX Corp. (True Solid/Master) и SolidWorks Corp. (SolidWorks);

- частные ядра геометрического моделирования, которые разрабатываются и поддерживаются разработчиками CAD-систем для использования исключительно в своих приложениях. Например, ядро thinkdesign являющееся основой CAD-системы think3(think3 Inc.);
- ядра, доступные в исходном коде, которые также разрабатываются и поддерживаются одной компанией и затем лицензируются другим компаниям для использования в CAD-приложениях, но разработчики поставляют исходный код ядра и позволяют дорабатывать его (с учктом специфики конкретной лицензии). Например, Open CASCADE набор библиотек и средств разработки программного обеспечения, ориентированный на 3D-моделирование, которые позволяют пользователям компилировать код Open CASCADE на их платформах.

Разработка ГГИС с открытым кодом может быть выполнена только на полнофункциональном ядре геометрического моделирования и визуализации, распространяемого по свободной лицензии с возможностью самостоятельной доработки основных модулей. В качестве геометрического ядра нами был выбран Open CASCADE.

Средствами Python (высокоуровневый язык программирования общего назначения) рационально решать задачи объединения различных компонентов свободных и открытых промышленных библиотек, написанных на C++: геометрическое ядро Open CASCADE Community Edition 3D, SALOME GEOM for parametric modeling и SALOME SMESH for advanced meshing features, которые были разработаны компанией Open CASCADE SAS (до 2000 года Matra Datavision). Библиотека PythonOCC является связующим звеном между средой разработки на языке Python и открытым ядром 3D геометрического моделирования Open CASCADE на языке C++. PythonOCC (программный интерфейс для Open CASCADE на языке Python) в настоящее время содержит более 10000 классов и использует около 90% функциональности Open CASCADE. PythonOCC является кроссплатформенной библиотекой для разработки CAD/CAE/PLM, обеспечивающей следующие возможности: трехмерное гибридное и параметрическое моделирование, работа с топологией, обмен данными (поддержка форматов файла STEP/IGES), совместная разработка и web-сервисы, поддержка управления графическим интерфейсом пользователя (wxPython, PyQt, Python-xlib). PythonOCC расширяет функциональность ядра OpenCASCADE, давая возможность использовать wxPython, что позволяет не только создавать графический интерфейс (например, имеет построенную на основе ODBC библиотеку работы с базами данных), систему обмена данных между процессами, сетевую библиотеку и множество классов для работы с различными приложениями.

Современные стандарты представления информации ГИС ориентируются на использование специализированных СУБД, примером которой может быть PostgreSQL (свободная объектно-реляционная система управления базами данных) с расширением PostGIS (расширение PostgreSQL средствами работы с географическими объектами). Возможны различные варианты организации данных ГГИС в базе данных:

- хранение геометрии в классических структурах точек для каждого типа бровок и контуров рудных тел с последующим созданием во временной таблице нужной для визуализации и анализа геометрии средствами PostGIS;
- хранение геометрии средствами PostGIS с выборкой для визуализации и анализа нужных фрагментов.

Достоинства хранения геометрии в классических структурах:

- непосредственный доступ к координатам точек;
- возможность управлять генерацией геометрии PostGIS по ситуации.

Недостатки:

- необходимость написания промежуточного кода на языке для формирования геометрии;
- необходимость идентификации фрагментов на разных уровнях представления;
- низкое быстродействие.

Достоинства хранения геометрии средствами PostGIS:

- максимальное приближение к идее геореляционных БД;
- однообразный формат представления для всех манипуляций, отображения, анализа;
- единая система идентификации объектов;
- доступность всего арсенала функций PostGIS;
- возможность выборки геометрии PostGIS по запросам.

Недостатки:

- низкое быстродействие работы с фрагментами;
- необходимость написания промежуточного запроса для доступа к фрагментам геометрии (запросы точек, разрезов, истории модификаций).

Хранение всей геометрии в PostGIS с выборкой во временную таблицу для визуализации и анализа нужных фрагментов. Приведем пример некоторых объектов:

Бровки

1) Идентификатор (ПК)

2) Горизонт

3) Тип (верхняя, нижняя, съезд, предельная верхняя, предельная нижняя, развал верхняя,

развал нижняя, зона верхняя, зона нижняя, прирезка верх/низ)

4) Полилиния

Запрос к базе данных PostgreSQL с использованием внутреннего формата PostGIS:

«INSERT INTO edge (hor,edge_type,geom) VALUES («+str(id_hor)+», 2, GeomFromEWKT('SRID=-1;LINESTRING(0.5 0.5 0.5);»

Рудные тела

1) Идентификатор (ПК)

2) Горизонт

3) Высота слоя

4) Сорт

5) Полилиния

Запрос к базе данных PostgreSQL с использованием внутреннего формата PostGIS:

«INSERT INTO body (id_hor,h_body,id_sort,geom) VALUES («+str(id_hor)+»,»+str(Hust)+»,4, GeomFromEWKT('SRID=-1;LINESTRING(0.5 0.5 0.5));»

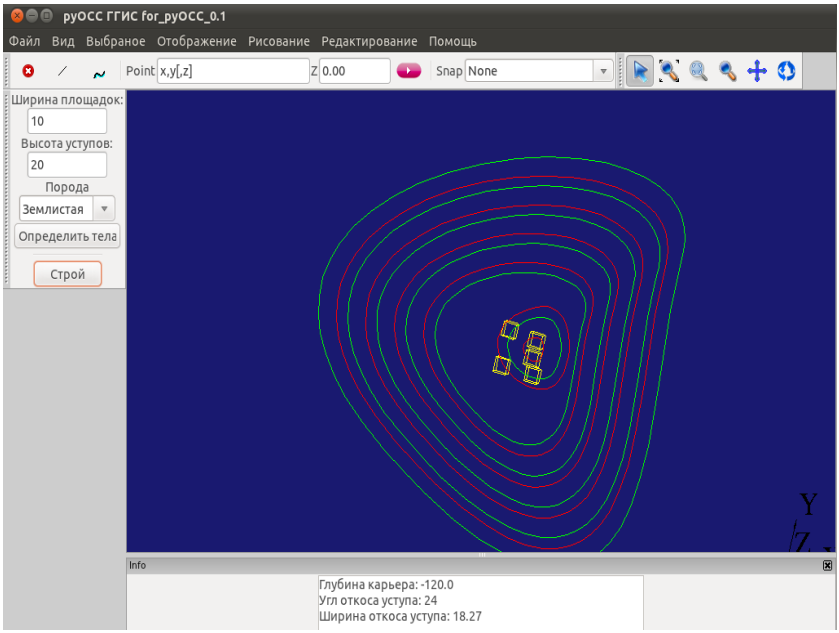
Для разработки демонстрационного прототипа были использованы: операционная система Ubuntu 10.04, язык программирования Python 2.6, PythonOCC 0.5, СУБД PostgreSQL 9.0 с расширениями PostGIS 1.3.4, адаптер для доступа к БД — Psycopg2-2.0.14 , графическая среда — wxPython 2.8.

Выводы

Моделью геологического строения месторождения является совокупность рудных тел. Тело характеризуется координатами вершин замкнутого полигона по подошве уступа и высотой. Техногенные образования – бровки уступов, съезды транспорта, скважины, контуры прирезок и другие, задаются 3-D полилиниями..

Разработан демонстрационный прототип выполняющий построение модели месторождения в выбранных нами погоризонтных планах в упрощенных телах - призмах с заданным полилинией основанием. Карьер задается бровками и отметками на площадках. Координаты точек полилинии на бровке задаются ручным вводом. Расчёт глубины карьера происходит на основании данных о глубине залегания тел. В соответствии с типом грунта рассчитывается угол откоса уступов..

Построение трехмерного карьера, для добычи пяти рудных тел продемонстрированно на рисунке.



В дальнейшем планируется наращивание функционала приложения до функционала аналогичных продуктов, таких как SURPAC, Micromine, Майнфрэйм Технология+БВР с сохранением открытости кода. Для удобства развертывания системы будет собран пакет установки для основных веток Linux.

Список источников

1. Geosoft. [Электронный ресурс]. — Режим доступа http://www.geosoft.com/, Дата обращения 01.09.11.

2. Аналитика и обзоры [Электронный ресурс]. — http://www.gisa.ru/analitiks.html Дата обращения 03.10.11.

3. Дискусии [Электронный ресурс]. — http://www.gisa.ru/discation.html Дата обращения 03.10.11

4. Википедия [Электронный ресурс]. — Режим доступа: http://ru.wikipedia.org/wiki/ Дата обращения 10.12.11.

5. Трубецкой К.Н. Справочник. Открытые горные работы / К.Н. Трубецкой, М.Г. Потапов, К.Е. Виницкий, Н.Н. Мельников и др. / — М. Горное бюро, 1994. 590 с.: ил.

Бинокулярное зрение в режиме реального времени

М. Хрущев

michael.khr@gmail.com

Аннотация. Цель данной работы — разработать систему плотного бинокулярного зрения, способную строить карту глубины в режиме реального времени.

Ключевые слова: бинокулярное зрение.

Цель задачи плотного бинокулярного зрения: по двум изображениям с камер нужно построить карту расстояний, которая будет описывать расположение каждой отдельной точки с изображения в пространстве относительно одной из камер. Для возможности трекинга объектов на изображении задачу необходимо решать в режиме реального времени — не менее 2 кадров в секунду. Алгоритм должен обладать хорошей степенью распараллеливаемости для возможности обработки на мультипроцессорных системах.

Если известно, какой пиксель левого изображения соответствует заданному пикселю правого изображения, то определить его реальное положение не сложно. Поэтому задачу бинокулярного зрения можно сформулировать, как задачу минимизации следующей суммы

$$\sum_{p \in P} C(p, f(p)) + \sum_{p \in D_1} D(p) + \sum_{q \in D_2} D(q) + \sum_{(p,q) \in N_1} L(f(p), f(q)) + \\ + \sum_{(p,q) \in N_2} L(f^{-1}(p), f^{-1}(q)),$$

Игнатов Д. И., Яворский Р. Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16–18 марта, 2012

где f — отображение из множества пикселей левого изображения в соответствующие пиксели правого;

f^{-1} — обратное отображение из множества пикселей правого изображения в соответствующие пиксели левого;

P — множество пикселей левого изображения, для которых существует отображение;

D_1, D_2 — множества скрытых пикселей на левом и правом изображении соответственно;

N_1, N_2 — множество пар стоящих рядом по горизонтали и вертикали пикселей с левого и правого изображений соответственно и имеющие образ;

C — штраф за различие цветов соответствующих пикселей;

D — штраф за скрытие пикселя;

L — расстояние между пикселями.

В одномерном пространстве существует похожая задача — вычисление расстояния Левенштейна. Это расстояние определяет минимальное количество вставок, добавлений и удалений символов, необходимое, чтобы перевести одну строку в другую. Расстояние Левенштейна можно также сформулировать, как минимизацию суммы

$$\sum_{(p,q)\in P} 1 + \sum_{(p)\in S_1} 1 + \sum_{(q)\in S_2} 1,$$

где P — замененные символы, S_1 — удаленные, S_2 — добавленные.

Для решения этой задачи существует алгоритм, работающий за время $O(m \cdot n)$, где m — длина первоначальной строки, n — измененной.

В качестве штрафа за удаление, добавление и изменение можно брать не константные выражения. Тогда минимизирующая функция будет выглядеть так:

$$\sum_{(p,q)\in P} C(p,q) + \sum_{(p)\in S_1} D(p) + \sum_{(q)\in S_2} A(q)$$

Алгоритм поиска расстояния Левенштейна основан на динамическом программировании. Для поиска расстояния между строками s_1 и s_2 вводится понятие состояния $S(i,j)$ ($i \leqslant |s_1|$, $j \leqslant |s_2|$), которое показывает, какое минимальное количество операций вставок и удаления требуется, чтобы из префикса s_1 длины i получить префикс s_2 длины j.

$S(0,0) = 0$,
$S(i,0) = i$,
$S(0,j) = j$;
если $s_1[i] = s_2[i]$, то $S(i,j) = S(i-1,j-1)$;

иначе $S(i,j) = \min(S(i-1,j)+1,\ S(i,j-1)+1,\ S(i-1,j-1)+1)$.

В качестве состояния для универсальности модели имеет смысл использовать более сложные объекты. Также можно использовать более сложные штрафные функции, вроде тех, что рассматривались ранее.

Для того, чтобы свести задачу бинокулярного зрения к задаче попарного анализа строк, необходимо сделать некоторые допущения.

1) Каждая пиксельная строка соответствует некоторой плоскости и содержит только точки из сечения этой плоскостью с реальными объектами.

2) Для каждой пары изображений для соответствующих пиксельных строк эти плоскости совпадают. Иначе говоря, на изображениях можно выделить такие пары строк, которые содержат наборы проекций точек на камеру, принадлежащих одному сечению.

Сложность поиска модифицированного расстояния Левенштейна между двумя строками пикселей будет достаточно велика — для изображений ширины N пикселей, это потребует порядка O(N^2) операций и выделения O(N^2) памяти. Поэтому в качестве символов имеет смысл использовать более сложные объекты, например отрезки пикселей, предположительно принадлежащие одному объекту. Для выделения таких отрезков можно использовать граничные точки. После выделения и совмещения отрезков, каждый из таких отрезков можно анализировать попарно.

При построчной обработке каждую пару строк можно анализировать в отдельном потоке, при этом у потоков не возникает разделяемых ресурсов.

На данный момент реализована ранняя версия решения на C#.Net + CUDA, принцип работы:

1) принимаются изображения с 2 камер;

2) на изображении ослабляется шум при помощи размытия или при помощи использования предыдущих снимков с камер;

3) на каждом изображении для каждой строки при помощи GPU находятся слитные отрезки примерно одного цвета; для определения границ используется модификация палитры $Lu*v*$;

4) при помощи GPU отрезки совмещаются модифицированным алгоритмом Левенштейна;

5) результат переводится в карту расстояний.

В модифицированном алгоритме Левенштейна при подсчете очередного состояния в зависимости от минимизации штрафа принимается одно из решений:

— рассматриваемые отрезки объявляются похожими; добавляется штраф за их различие;
— левый отрезок объявляется скрытым; добавляется штраф за удаление в зависимости от длины;
— правый отрезок объявляется скрытым; добавляется штраф за удаление в зависимости от длины;
— левый отрезок объявляется похожем на предыдущий левый добавленный и они склеиваются; штраф за различие отрезков пересчитывается за счет склейки, добавляется штраф за расстояние между отрезками;
— правый отрезок объявляется похожим на предыдущий правый добавленный и они склеиваются; штраф пересчитывается аналогично предыдущему случаю.

Последние 2 варианта улучшают результат в тех случаях, когда один отрезок разбивается на несколько из-за шума или тонких объектов, или 2 отрезка сливаются в один из-за нечеткой границы.

Данная реализация при изображениях размера 640×480 на видеокарте Nvidia 260 GTX в среднем успевает обработать более 5 кадров в секунду и дает неплохой результат для многих обработанных строк.

Рис. 1. На изображении видно отутствие вертикальной синхронизации

Проблемы, которые есть в текущей реализации:
— не выполняется вертикальная синхронизация строк для устранения ошибок совмещения;
— отрезки пока что не совмещаются попиксельно — это должно убрать большую часть ошибки;
— при подсчете штрафа почти не используются реальные координаты камер;
— близко находящиеся объекты могут сильно изменить порядок следования отрезков, что не учитывается текущей моделью; есть предположение, что это может быть решено введением в состо-

яние слоев, каждый из которых будет содержать в себе отрезки, найденные на определенном предположительном расстоянии от камер;

— текущая математическая модель содержит много констант и функций, которые очень сложно правильно оценить; для улучшения результата планируется использовать генетические алгоритмы или другой метод оптимизации;
— для выделения цветовых отрезков используются только границы; планируется использовать более сложные элементы.

Список источников

1. Yang Q., Wang L., Yang R., Wang S., Liao M., and Nister D. Real-time global stereo matching using hierarchical belief propagation. BMVC 2006.
2. Salmen J., Schlipsing M., Edelbrunner J., Hegemann S., and Lueke S. Real-time stereo vision: making more out of dynamic programming. CAIP 2009.

Анализ ассоциативных тезаурусов и возможность их применения в задачах машинного перевода[1]

Е.А.Выломова

evylomova@gmail.com

МГТУ им. Н. Э. Баумана, каф. Системы Обработки Информации и Управления

Аннотация. В работе представлен анализ ассоциативных тезаурусов и, в частности, Русского Ассоциативного Тезауруса (РАС). Показано, что сеть, основанная на данных тезауруса, принадлежит к классам «small-world» и «scale-free». Помимо этого, приведены результаты сравнения тезаурусов на различных языках. В работе также рассматривается вопрос о возможности применения данных тезаурусов для улучшения параметров машинного перевода.

Ключевые слова: ассоциативные тезаурусы; ассоциативные сети; языковые ресурсы; извлечение знаний.

Введение

В соответствии с предположением Фирта [1], что смысл понятия раскрывается в его взаимосвязи с соседними концептами, а также согласно представлению об ассоциации как базовом механизме сознания человека, описанному в работах Deese [2] and Cramer [3], вербальные ассоциации отражают структурные шаблоны взаимоотношений между понятиями. Со времен Ф. Гальтона ассоциативные эксперименты нача-

[1] Работа выполнена в рамках гранта РГНФ №12-04-12039в

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16-18 марта, 2012.

ли активно использоваться как эмпирический метод наблюдения процессов мышления, запоминания и организации знаний.

Результаты ассоциативных экспериментов наиболее часто хранятся в виде ассоциативных словарей (тезаурусов), представляющих собой набор триплетов <*стимул, реакция, частота стимульно-реактивной пары*>. В качестве стимула и реакции выступают слова и словосочетания. Подобные ассоциативные тезаурусы на текущий момент существуют на английском [2,3,4,5], японском [6,7], шведском [8], русском [9, 10], чешском [11], корейском [12], голландском [13] языках и иврите [14].

Описанная выше структура данных позволяет на базе ассоциативного тезауруса создать ассоциативную сеть. Узлами в ассоциативной сети являются концепты (например, *«еда»*, *«Великая Отечественная Война»*). Концепты можно разделить на три вида: концепты-стимулы (S), концепты-реакции (R) и концепты-стимулы-реакции (SR). Два узла u и v сети соединены между собой дугой, если u является стимулом (S или SR), а v – реакцией (SR или R) и существует ассоциативная связь между ними. В основе большинства семантических сетей, к коим можно полноправно отнести и ассоциативную сеть, лежит граф $G = (V, E)$, где V – множество узлов (вершин), а E – множество ребер или дуг в случае ориентированного графа. В дальнейших рассуждениях понятия «граф» и «сеть» рассматриваются как синонимичные.

Основные понятия теории графов

В данном разделе хотелось бы кратко перечислить основные параметры, характеризующие граф, которые затем будут использованы для сравнения их между собой.

Две вершины, связанные ребром или дугой, являются *соседями*. В случае орграфа вершина характеризуется *степенью* k^{in} и k^{out}, то есть количеством входящих и исходящих дуг соответственно. Если же граф неориентированный, то параметру k соответствует количество ребер вершины.

В неориентированном графе под *путем* понимается последовательность ребер, соединяющая пару вершин. В ориентированном графе *путь* – это набор дуг, по которым можно перейти от одной вершины к другой. В случае конкретного *графа длина пути* определяется как количество ребер, или дуг, между вершинами. Расстояние между вершинами u и v – длина кратчайшего пути между ними.

Диаметр сети D – это максимальное расстояние между вершинами для всех пар вершин. Поэтому диаметр сети D и средняя длина кратчайшего пути L взаимосвязаны.

Функция распределения степени $P(k)$ представляет собой вероятность, что случайно выбранная вершина будет иметь степень k. Для ориентированного графа количество входящих ребер считается более важным показателем и степень в случае такого графа вычисляется как количество входящих ребер.

Таблица 1. Основные характеристики графов

Обозначение	**Определение**
N	общее количество вершин
L	средняя длина кратчайшего пути между парами вершин
D	диаметр сети
k, k^{in}, k^{out}	степень вершины
$\langle k \rangle$	средняя степень вершины
γ	показатель в функции распределения степени

Сети «small-world» и «scale-free»

Впервые феномен «small-world» наблюдал Milgram [15] в экспериментах с социальными сетями. Он предположил, что любые два человека в США разделены сравнительно малым количеством знакомых или друзей («6 степеней разделения»). Феномен прослеживается на случайных графах, где каждая пара вершин соединена ребром с вероятностью p. Когда p достаточно высока, вся сеть становится связной и средняя длина пути L между двумя случайными вершинами возрастает логарифмически по отношению к общему числу вершин: $L \propto \log(N)$[16].

Watts, Strogatz [17] в своих работах возобновили интерес к такому типу сетей и показали, что система энергоснабжения запада США, нервная система червя Caenorhabditis elegans, международная сеть киноактеров также к ним относятся.Watts и Strogatz обозначили понятие «small-world» структур, особенностью которых является малое значение среднего кратчайшего пути между парами вершин и сравнительно высокое значение среднего коэффициента кластеризации.

В дальнейших исследованиях было выявлено, что World Wide Web, сети научного сотрудничества, метаболические сети в биологии также обладают структурой «small-world».

Amaral, Scala, Barthélémy, Stanley [18] изучали различные классы сетей со структурой «small-world», сравнивая функцию распределения степени *P(k)*. В результате были выделены два типа распределений: экспоненциальное и степенное. Во втором случае функция имеет вид $P(k) \approx k^{-\gamma}$, где $\gamma \in (2..4)$. Данный тип сетей получил название «scale-

free», их основной особенностью является наличие вершин-хабов, через которые проходит множество путей, соединяющих остальные вершины.

Результаты

В рамках данного исследования преимущественно изучался Русский Ассоциативный Словарь [10] (РАС). Объем экспериментальной выборки РАС составил *102516* различных стимульно-реактивных пар вида $\langle c_i, r_j, freq_{ij}, \rangle$, где $c_i = \overline{1{,}6577}$ – стимулы, $r_j = \overline{1{,}21312}$ – реакции, $freq_{ij}$- частота стимульно-реактивных пар. Впоследствии частота была преобразована в относительную $weight_{ij} = \frac{freq_{ij}}{\sum_{j=1}^{n} freq_{ij}}$, где n – общее количество различных реакций на данный стимул. На основе ассоциативного тезауруса была построена сеть. Ниже приведено сравнение параметров полученной сети с американским аналогом, а также с сетью WordNet.

Таблица 2. Сравнение параметров американской, русской ассоциативных сетей и WordNet

	Ориент. (РАС)	**Неориент. (РАС)**	**Ориент. (амер.)**	**Неориент. (амер.)**	**WordNet**
N	23196	23196	5018	5018	122005
L	3.989	3.836	4.27	3.04	10.56
D	8	7	10	5	27
γ	2.12	2.103	1.79	3.01	3.11
<*k*>	4.423	8.236	12.7	22	1.6

Как видно, РАС можно также отнести к сетям типа «small-world» и «scale-free». Свойства этих сетей, а именно малая величина среднего кратчайшего пути и наличие вершин-хабов, позволили разработать методику эффективного хранения ассоциативного графа. Методика рассчитана на частые запросы кратчайших путей между вершинами и в большинстве случаев позволяет, во-первых, не загружать полный граф в операционную память, а, во-вторых, обходить лишь необходимую часть графа.

Кроме того, было получено, что для сетей на различных языках множества концептов, соответствующих вершинам-хабам, во многом пересекаются и содержат такие базовые понятия как *«отец»*, *«дитя»*, *«язык»*, *«вода»*, *«хороший»* и другие. Данный факт послужил толчком

для начала исследования возможности построения отображения сети на русском языке в аналоги на других. Эта технология основана на поиске близких концептуальных полей для двух и более языков с использованием словаря-переводчика. Предполагается, что подобный подход позволит улучшить машинный перевод по следующим причинам: 1) ассоциативные сети достаточно хорошо отображают взаимосвязи между понятиями; 2) ассоциативные сети имеются на многих языках и содержат схожий набор базовых стимулов.

Заключение

Целью данной статьи являлось не только представление результатов исследования, но также и привлечение внимания к ассоциативным тезаурусам как к дополнительным языковым ресурсам, на основе которых возможно проводить валидацию онтологий и семантических отношений, полученных автоматическими методами.

Основные направления дальнейшего исследования следующие: 1) построение комбинированной модели русского и английского ассоциативных тезаурусов, которая позволит улучшить параметры машинного перевода; 2) сравнение статистических характеристик данных ассоциативного тезауруса и результатов квантитативного анализа текстовых корпусов.

Список источников

1. Firth, J.R. Selected Papers of J. R. Firth 1952-1959. Palmer, F.R. (ed.), Longman. London. 1968

2. Deese, J. The Structure of Associations in Language and Thought. The John Hopkins Press. Baltimore. 1965

3. Cramer, P. Word association. NY: Academic Press, 1968

4. Nelson, D. L., McEvoy, C. L., & Schreiber, T. A. Word Association, Rhyme and Fragment Norms.The University of South Florida, 1999 http://luna.cas.usf.edu/~nelson/

5. Kiss, G.R., Armstrong, C., Milroy, R., and Piper, J. An associative thesaurus of English and its computer analysis. In Aitken, A.J., Bailey, R.W. and Hamilton-Smith, N. (Eds.), The Computer and Literary Studies. Edinburgh: University Press, 1973

6. Okamoto and S. Ishizaki.. Construction of associative concept dictionary with distance information, and comparison with electronic concept dictionary («　概念間距離の定式化と既存電子化辞書との比較　自然言語処理»), vol. 8, pages 37-54, 2001 (Japanese)

7. Joyce, Terry. Building a word association database for basic Japanese vocabulary. Poster session presented at The 3rd Annual Meeting of the Japanese Society for Cognitive Psychology, 28-29 May, Kanazawa University, Kanazawa, Japan, 2005

8. Lönngren, L. A Swedish Associative Thesaurus. Euralex '98 Proceedings, Vol. 2, pages 467-474, 1998

9. Леонтьев А.А. Словарь ассоциативных норм русского языка. Москва, 1977

10. Караулов Ю.Н., Тарасов Е.Ф., Сорокин Ю.А., Уфимцева Н.В., Черкасова Г.А. Ассоциативный тезаурус современного русского языка. РАН, 1999

11. Novák, Z. Volné slovní párové asociace v češtině. Praha. 1988 (Czech)

12. Jung J., Na L., Akama H. Network analysis of korean associations,Proceedings of the NAACL HLT 2010 First Workshop on Computational Neurolinguistics, - Los Angeles, CA, 2010, pages 27-35, 2010

13. De Groot, A. M. B. Word association norms with response times(«Woordassociatienormen met reactietijden»). Nederlands Tijdschrift voor de Psychologie (Dutch Journal of Psychology), 43, pages 280-296, 1988

14. Rubinsten, O., Anaki, D., Henik, A., Drori, S., Faran, Y. Free association norms in the Hebrew language. In A. Henik, O. Rubinsten, & D. Anaki, (Eds.). Word Norms in Hebrew, Ben-Gurion University of the Negev, pages 17-34, 2005 (Hebrew)

15. Milgram, S. The small-world problem. Psychology Today, 2, pages 60–67, 1967

16. Erdös, P., & Réyni, A. On the evolution of random graphs. Publications of the Mathematical Institute of the hungarian Academy of Sciences, 5, pages 17–61, 1960

17. Watts, D. J., Strogatz, S. H. Collective dynamics of ‘small-world’ networks. Nature, 393, pages 440–442, 1998

18. Amaral, L., Scala A., Barthelemy, M., Stanley, H. Classes of small-world networks.Proc. Natl. Acad. Sci. U. S. A. 97, pages 11149-11152, 2000

Распознавание дорожных знаков на основе машины опорных векторов и показателя сопряжённости

Р.К. Захаров, В. А. Фурсов

E-mail: roman.zakharovp@yandex.ru

E-mail: fursov@ssau.ru

Самарский государственный аэрокосмический университет имени академика С.П. Королёва (национальный исследовательский университет)

Институт систем обработки изображений РАН

Аннотация. В работе рассматривается метод и строится алгоритм распознавания дорожных знаков, основанный на использовании в качестве меры близости так называемого показателя сопряженности. Приводятся результаты сравнительных экспериментальных исследований алгоритмов, использующих различные меры близости.

Ключевые слова: распознавание; метрики; машина опорных векторов; показатель сопряжённости.

Введение

В связи с работами последних лет по созданию систем активной безопасности автомобилей появляется все больше публикаций, связанных с задачей распознавания дорожных знаков. Обычно эта задача решается в два этапа: детектирование и локализация области знака на изображении и собственно распознавание. Общая постановка задачи распознавания следующая.

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений, Сетей и Текстов, Екатеринбург, 16-18 марта, 2012.

Предполагается, что имеется M изображений каждого из K объектов. Каждое изображение представляется вектором $x = [x_1, x_2, \dots, x_N]^T$ размерности N, где $x_1, x_2, \dots, x_N$ – признаки. Векторы, соответствующие изображениям одного объекта, составляют класс. Совокупность векторов признаков всех классов образует обучающую выборку. Решение задачи распознавания состоит в конструировании решающей функции $f: R^N \to \{0,1, \dots, K\}$, которая каждому вектору x ставит в соответствие некоторый класс. Для уменьшения числа неправильных классификаций вводится также класс с номером 0, соответствующий отказу в распознавании.

Качество распознавания, зависит от выбора системы признаков. Наряду с выбором системы признаков большую роль играют также используемая при распознавании мера близости и построенное на ее основе решающее правило. Известны и широко используются следующие меры близости для анализа изображений: евклидово расстояние, расстояние Махаланобиса, расстояние Хаусдорфа. В работе [1] предложен классификатор, основанный на использовании в качестве меры близости так называемого показателя сопряженности. Сравнительные исследования этого классификатора с другими, построенными на других мерах близости, показали его эффективность в задаче распознавания лиц [2].

Для решения задачи распознавания дорожных знаков наиболее широко используется машина опорных векторов, а в качестве признаков наиболее популярными являются гистограммы ориентированных градиентов(HOG). В настоящей работе ставится задача провести сравнительные исследования машины опорных векторов с методом классификации на основе показателя сопряженности.

Построение решающего правила на основе показателя сопряжённости

Для построения решающего правила введем в рассмотрение так называемый показатель сопряженности с подпространством, натянутым на векторы признаков образов объектов из заданного класса:R

$$R_k = \frac{x^T X_k [X_k^T X_k]^{-1} X_k^T x}{x^T x}$$

Здесь x − вектор признаков неизвестного образа, предъявленный для установления близости к k-му классу, а Xk – NxM – матрица, составленная из векторов образов, принадлежащих k-му классу.

Предположим, что для каждого (k-го) класса сформирована следующая NxN-матрица Q:

$$Q_{k,R} = X_k [X_k^T X_k]^{-1} X_k^T$$

Соответствующая решающая функция *f(x)* строится следующим образом. Вектор *x* принадлежит *m*-му классу, то есть $f(x) = m, m = 1 \dots K$,

$$\text{если } R_m = \max_k R_k \text{, где } R_k = \frac{x^T Q_{k,R} x}{(x^T x)}$$

Описание эксперимента

В сравнительном эксперименте осуществлялось распознавание дорожных знаков с использованием решающих правил на основе различных метрик. В частности, рассматривались семь различных метрик: расстояния Евклида, Манхэттена, Махаланобиса, Чебышева, Камберра и Хаусдорффа. Задача заключалась в сравнении числа ложных срабатываний указанных методов с показателем сопряженности на тестовом наборе данных дорожных знаков German Traffic Sign Recognition Benchmark (GTSRB).

На рисунке 1 показаны примеры двух групп (классов) дорожных знаков: «Ограничение скорости» и «Проезд запрещен», на которых происходило обучение. Использовалась общая схема распознавания, включающая этапы локализации знака (1) и собственно распознавания (2).

Рис. 1. Примеры обучающих классов из базы данных German Traffic Sign Recognition Benchmark

На этапе локализации осуществлялась фильтрация с целью выделения знака по цветовым компонентам в пространстве RGB по следующим правилам:

$$\frac{r}{g} > \alpha, \frac{r}{b} > \beta$$

При этом может быть выявлено несколько областей (кандидатов в знаки). Затем выделяются контуры областей, при этом происходит отсечение контуров небольшого размера. По известному контуру

выделяется сам знак, и осуществляется этап распознавания. В эксперименте по исследованию меры Хаусдорфа в качестве набора признаков использовались координаты точек контура форм дорожных знаков. Ниже приводится краткое описание исследовавшихся метрик.

Мера Евклида $d(x,y) = \sqrt[2]{\sum_{i=1}^{N}(x_i - y_i)^2}$, x и y – это вектора размерности N

Мера Манхэттена $d(x,y) = \sum_{i=1}^{N}|x_i - y_i|$, x и y – это вектора размерности N

Расстояние Махаланобиса $d(x,y) = (x_m - x)\left((X_k - x_m)(X_k - x_m)^T\right)^{-1}(x_m - x)$, Xk – это k-ый класс, xm – средний вектор k-го класса, x – тестовое изображение

Расстояние Чебышева $d(x,y) = \max_i|x_i - y_i|$, x и y – это вектора размерности N

Расстояние Камберра $d(x,y) = \sum_{i=1}^{N}\frac{|x_i - y_i|}{|x_i + y_i|}$, x и y – это вектора размерности N

Расстояние Хаусдорффа
$H(X,Y) = \max\{\sup_{x\in X}\inf_{y\in Y} d(X,Y), \sup_{y\in Y}\inf_{x\in X} d(X,Y)\}$, где $d(X,Y)$ это расстояние между векторами, к примеру, Евклидово расстояние, а X и Y это два множества.

Тестирование осуществлялось на 3-х тестовых наборах данных: 1000, 2000 и 4000 изображений. Для обучения использовались следующие обучающие выборки из 5, 10, 15, 20, 25 и 30 тестовых векторов. Сравнение осуществлялось с машиной опорных векторов и методом, основанным на показателе сопряженности.

Результаты экспериментов

На рисунке 2 приведены гистограммы процента ложных срабатываний на тестовой выборке в 1000 изображений. Здесь по оси ординат (p) – процент ложных срабатываний, а по оси абсцисс (n) - количество обучающих векторов.

Машина опорных векторов этот метод был реализован с использованием алгоритма Sequential Minimal Optimization (SMO). Это достаточно простой алгоритм, который за доступное время решает задачу квадратичной оптимизации для метода машины опорных векторов [3].

На рисунке 2, е) представлена гистограмма результатов сравнения машины опорных векторов со всеми другими методами, в частности, приведены проценты ложных срабатываний для различных метрик и машины опорных векторов на тестовой выборке в 1000 изображений.

Здесь введены обозначения: 1 – показатель сопряжённости: 2 – расстояние Евклида; 3 – расстояние Манхэттена; 4 – расстояние Чебышева; 5 – расстояние Камберра; 6 – машина опорных векторов с линейной ядровой функцией.

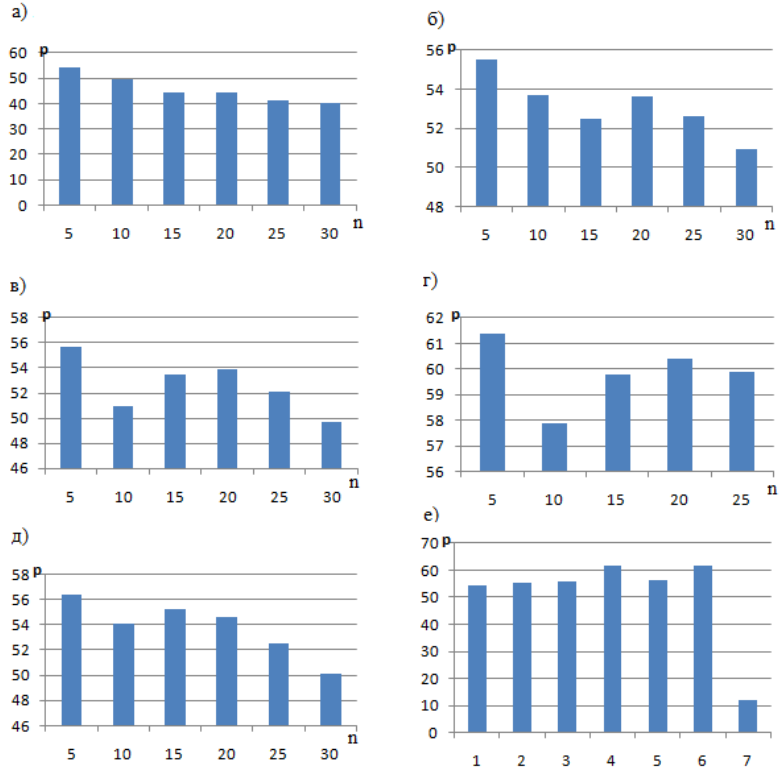


Рис. 2 – Результаты экспериментов: а) – показатель сопряжённости, б) –расстояние Евклида, в) – расстояния Манхэттена, г) – расстояния Чебышева, д) –расстояния Камберра, е) сравнение всех методов с машиной опорных векторов.

Выводы

Эксперименты показывают, что метод, основанный на показателе сопряжённости, имеет более высокую надёжность в задачах распознавания знаков изображений по сравнению с методами, основанными на указанных выше метриках, однако метод, реализующий машины опорных векторов превосходит его.

Работа выполнена при поддержке Министерства образования и науки (ГК № 07.514.11.4105) и РФФИ (проект № 11-07-12051-офи-м)

Список источников

1. Козин, Н.Е. Построение классификаторов для распознавания лиц на основе показателей сопряженности [Текст] / Н.Е. Козин, В.А. Фурсов // Компьютерная оптика. – 2006. – № 28. – С. 160-163.

2. Козин, Н.Е. Распознавание лиц по показателям сопряженности в пространстве суммирующих инвариантов [Текст] / Н.Е. Козин, В.А. Фурсов // Компьютерная оптика. – 2008. – Том 4, № 32.

3 Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines John C. Platt Microsoft Research jplatt@microsoft.com Technical Report MSR-TR-98-14 April 21, 1998 © 1998 John Platt.

Научное издание

Доклады по компьютерным наукам
и информационным технологиям
№ 1, 2012 г.

Доклады всероссийской научной конференции АИСТ'12
Екатеринбург, 16 – 18 марта 2012 года
«Модели, алгоритмы и инструменты анализа данных;
результаты и возможности для анализа изображений,
сетей и текстов»

Редакторы Дмитрий Игнатов, Ростислав Яворский
Компьютерная верстка Б. Агафонцев
Дизайн обложки Ю. Васильев

Подписано в печать 06.03.2012. Формат 60x90/16.
Гарнитура «Таймс». Бумага офсетная. Печать офсетная.
Усл. печ. л.26,25. Тираж 500 экз. Заказ № 975
Национальный Открытый Университет «ИНТУИТ»
Москва, Электрический пер., 8, стр.3.
Телефон: +7 (499) 253-9312, 253-9313, факс: +7 (499) 253-9310
E-mail: info@intuit.ru, http://www.intuit.ru