Правительство Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего профессионального образования

«Национальный исследовательский университет «Высшая школа экономики»

Факультет Бизнес-информатика

Отделение Программной инженерии

Кафедра Управление разработкой программного обеспечения

УТВЕРЖДАЮ
Зав. Кафедрой УРПО
_____ С.М. Авдошин
«___»_____ 2014 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

по направлению 231000.62 Программная инженерия

подготовки бакалавра

На тему: Программа отбора информативных признаков по прецедентам с использованием алгоритма GLMNET

Студент группы №471ПИ _____ /Мордовин Д.И. /

«____» _____ 2014 г.

Руководитель ВКР проф. каф. УРПО, д.ф.-м..н._____ /Подбельский В.В./

«____» _____ 2014 г.

Москва, 2014

Реферат

В работе рассматривается проблема отбора информативных признаков в задачах машинного обучения. Эта задача является составной частью методов сокращения признакового пространства, которая занимается улучшением качества исходных данных и, соответственно, алгоритмов машинного обучения. В связи с тем, что в последние годы алгоритмы машинного обучения приобрели огромную популярность в областях промышленности, активно использующих информационные технологии, и прикладных научных областях, данная задача имеет высокую актуальность.

В соответствии с техническим заданием разработан метод отбора признаков с использованием алгоритма GLMNET. В процессе выполнения работы выполнен анализ предметной области, изучены существующие методы отбора информативных признаков и смежные понятия и определения. Выполнена программная реализация разработанного метода и произведено его тестирование на прикладной задаче.

Ключевые слова: glmnet, elastic-net, регуляризация, логистическая регрессия, машинное обучение, отбор информативных признаков.

Работа состоит из 4 глав и 61 страницы, включающих в себя 18 иллюстраций, 2 таблицы, 10 источников и 5 приложений.

Содержание

Введение

Современная индустрия информационных технологий, в том числе вовлеченная в различные сферы промышленности и прикладных областей науки, позволяет решать все новые и новые проблемы, которые встают перед компаниями и исследовательскими центрами.

В последние несколько лет бурное развитие переживает такая область computer science как data mining. Рост популярности данного направления может быть обусловлен тем, что в процессе своей деятельности компании накапливают все большее количество данных, которые на первый взгляд кажутся несистематизированными и беспорядочными, однако потенциально могут содержать в себе некоторую полезную информацию.

Под data mining можно подразумевать поиск некоторых полезных закономерностей в имеющихся данных, включающий в себя некоторый теоретический аппарат. Так как здесь мы имеем дело с автоматизированной обработкой данных, хранящихся на ЭВМ посредством некоторого хранилища данных, то помимо чисто статистического или математического подходов в качестве такого аппарата может выступает некоторый комбинированный подход, включающий в себя алгоритмическую составляющую. В связи с этим для решения многих задач, которые ставятся в data mining, используются методы, предоставляемые таким разделом computer science как machine learning (машинное обучение) [2].

Однако с применением методов машинного обучения связаны несколько основных проблем. Во-первых, большинство из используемых алгоритмов содержат один или несколько внешне определяемых параметров, которые в значительной степени влияют на результат выполнения того или иного алгоритма. В связи с этим возникает проблема оптимального выбора параметров. Во-вторых, качество результата выполнения алгоритма машинного обучения напрямую зависит от самих входных данных.

Критерии качества исходных данных можно определить по-разному в зависимости от типа алгоритма машинного обучения. Например, в случае задачи классификации о качестве исходной выборки можно говорить с точки зрения того, насколько достоверно информация, в ней содержащаяся, позволит алгоритму отнести тот или иной объект к определенному классу.

Соответственно, в связи с чувствительностью методов машинного обучения к исходным данным возникает необходимость в предобработке с целью повышения их качества. Одним из подходов к предобработке данных является отбор информативных признаков.

Задача отбора информативных признаков [1] является одной из задач из области dimensionality reduction, которая занимается сокращением исходного признакового пространства выборки. Качественно, отбор представляет собой определение нерелевантных признаков, то есть таких, которые не содержат себе полезной информации в контексте некоторой задачи машинного обучения. Для иллюстрации примера неинформативного признака рассмотрим следующую задачу классификации (пол кодируется 0 для обозначения женщины или 1 — для мужчины):

Таблица 1

Пример выборки для задачи классификации по полу

Пол (класс)	Заработная плата (руб.)	Возраст	Количество зубов	...
0	32000	34	32	...
0	45000	43	31	...
1	55000	37	32	...
...	…	…	...	...

В этом случае можно из соображений здравого смысла определить самый неинформативный признак — количество зубов. В данной конкретной задаче данное свойство не несет никакой информации относительно пола человека, следовательно, включение его в выборку не позволит улучшить качество алгоритма классификации. Информативность остальных двух свойств (заработная плата и возраст) не является столь очевидной, и их релевантность необходимо определять существующими методами, которые будут рассмотрены далее.

Отбор информативных признаков позволяет решить несколько проблем, возникающих при построении модели.

Во-первых, задача отбора информативных признаков становится актуальной в силу того, что при составлении обучающей выборки зачастую невозможно заранее определить важность того или иного признака. Вместе с этим кажется естественным желание учесть при формировании выборки как можно больше признаков, ведь потенциально они могут увеличить точность алгоритма. Однако на практике часто получается так, что многие из тех признаков, которые оказались включенными в модель, на самом деле не несут полезной информации, либо являются шумовыми. Соответственно, при использовании такой избыточной модели в алгоритмах машинного обучения их качество только падает, а не возрастает, как ожидалось. Отбор информативных признаков в данном случае позволяет

методу машинного обучения отбросить из модели шумовые признаки.

Во-вторых, при увеличении количества признаков в модели, несмотря на то, что ошибка на обучающей выборке может монотонно убывать, ошибка при тестировании на независимых данных может убывать только до прохождения через точку минимума, а далее возрастать. Это явление называется переобучением. Если в процессе предобработки данных произвести отбор информативных признаков, то переобучение можно свести к минимуму, выбрав оптимальный набор признаков.

В-третьих, сбор информации при составлении выборки может потребовать значительных затрат. Соответственно, уменьшение количества признаков, значения которых необходимо собрать, может позволить минимизировать затраты.

В-четвертых, большинство методов машинного обучения обладает значительной вычислительной трудоемкостью, зависящей, в первую очередь, от размера выборки. Отбор информативных признаков позволяет увеличить скорость работы алгоритма.

В-пятых, меньшее число признаков делает модель более простой и прозрачной для восприятия.

В данной работе ставится задача отбора информативных признаков с использованием алгоритма GLMNET (elastic-net regularized generalized linear models) [3], где в качестве признаков выступают численные свойства элементов некоторых химических соединений, а обучающая выборка выглядит следующим образом:

Таблица 2

Таблица исходных данных для поставленной задачи

Class	E8-1	I8-2	S6-3	S5-4	E2-5	E5-6	E6-7	…	C5-9
1	2.25	102	1.32	4.6	0.8	418.8	3051	…	89.2
1	2.1	52.8	1.43	4.9	0.8	403	2632	…	80.9
1	2.25	102	1.32	4.6	0.8	418.8	3051	…	89.2
1	2.25	102	1.32	4.6	0.8	418.8	3051	…	89.2
1	2.25	102	1.32	4.6	0.8	418.8	3051	…	89.2
1	3.9	46.1	1.53	2.29	1.8	589.3	1971	…	182.2
1	2.25	102	1.32	4.6	0.8	418.8	3051	…	89.2
1	2.25	102	1.32	4.6	0.8	418.8	3051	…	89.2
1	2.25	102	1.32	4.6	0.8	418.8	3051	…	89.2
1	2.25	102	1.32	4.6	0.8	418.8	3051	…	89.2

1	2.25	102	1.32	4.6	0.8	418.8	3051	…	89.2
1	2.25	102	1.32	4.6	0.8	418.8	3051	…	89.2
2	1.95	35.9	1.55	5.45	0.7	375.7	2229	…	76.1
2	1.95	35.9	1.55	5.45	0.7	375.7	2229	…	76.1
2	5.9	46.9	1.93	2.99	0.9	389.3	2979	…	180.4
2	2.7	141	1.07	3.73	0.9	495.8	4562	…	107.3
2	2.7	141	1.07	3.73	0.9	495.8	4562	…	107.3
2	2.7	141	1.07	3.73	0.9	495.8	4562	…	107.3
….	…	…	…				…	…	…
K	4.35	429	1.3	2.9	1.93	731	2074	…	284.6

В данной таблице первый столбец содержит в себе метки классов (типов химических соединений), а остальные столбцы — значения свойств элементов данного химического соединения.

В качестве целей выпускной квалификационной работы ставятся:

- Создание библиотеки, реализующей отбор информативных признаков, основанный на алгоритме GLMNET, на языке Java для задач вида, описанного таблицей 2;
- Создание консольного приложения с целью его предоставления в Институт Металлургии им. Байкова РАН для последующей интеграции с информационно-аналитической системой в составе модуля отбора информативных признаков.

В процессе выполнения выпускной квалификационной работы ставятся следующие задачи:

- Анализ существующих методов отбора информативных признаков;
- Изучение алгоритма GLMNET;
- Разработка метода отбора признаков, основанного на GLMNET;
- Разработка библиотеки на языке Java для разработанного метода отбора признаков;
- Согласование формата входных и выходных данных для консольного приложения, подлежащего дальнейшей интеграции с информационно-аналитической системой в составе модуля отбора информативных признаков;
- Разработка консольного приложения;

- Разработка приложения с графическим пользовательским интерфейсом для дальнейшего тестирования разработанного метода;
- Проведение тестирования метода на некоторых наборах исходных данных;
- Соотнесение результатов работы реализованного метода с существующим решением.

1. Обзор предметной области

1.1. Критерии качества моделей

Перед тем, как рассматривать методы отбора информативных признаков, необходимо понять, какие существуют пути для выбора модели машинного обучения.

После того, как алгоритмом машинного обучения была произведена настройка параметров модели, необходимо определить насколько хорошей предсказательной способностью она обладает.

Обычно рассматривают два критерия качества [1]: внешний и внутренний.

Внутренний критерий используется для настройки параметров модели и чаще всего представляет собой некий функционал ошибки, который характеризует качество метода только на обучающей выборке. Этот критерий не следует использовать для выбора модели, так как его лучшее значение (в случае, если вычисляется ошибка, то минимальное значение) в большинстве случаев достигается на такой модели, которая подверглась переобучению. Это объясняется тем фактом, что, как было указано выше, ошибка на обучающей выборке при увеличении сложности модели (использовании большего числа признаков) в большинстве случаев монотонно убывает.

Внешний критерий позволяет оценить построенную модель, используя те данные для тестирования, которые не были включены в обучающую выборку. Следовательно, критерий вычисляется на данных, для которых модели заранее не был известен ответ.

Рассмотрим некоторые виды внешних критериев:

1. *Ошибка на отложенных данных* (hold-out error).

   Основная идея этого критерия заключается в том, что мы разделяем исходную (полную) выборку данных на две части: на одной обучается модель, на другой в дальнейшем будет производиться тестирование. Однако здесь важно соблюдать следующие правила: во-первых, выборки не должны пересекаться, во вторых — должны быть независимы. В противном случае критерий может мало отличаться от внутреннего критерия, и его использование для выбора модели может привести к переобучению.

2. *Ошибка скользящего контроля* (cross-validation error).

   Данный критерий в некоторой степени является обобщением предыдущего. В отличие

от критерия ошибки на отложенных данных, функционал скользящего контроля в целях независимости результата от выбранного разбиения вычисляет ошибку на нескольких разбиениях. В зависимости от способа разбиения выборки выделяют несколько видов скользящего контроля:

- *Полный скользящий контроль* (complete cross-validation) заключается в вычислении средних ошибок на всех возможных разбиениях. Так как число всех разбиений экспоненциально зависит от количества элементов выборки (прецедентов), то даже при небольших выборках вычисление такого функционала становится довольно трудоемким. Для реальных же выборок, которые достигают несколько сотен и тысяч прецедентов, применить данный критерий с использованием современных машин и вовсе невозможно.
- *Скользящий контроль по отдельным элементам* (leave-one-out cross-validation). Здесь в качестве тестовой (контрольной) выборки выбирается один объект из полной выборки. При этом длины обучающих выборок всего на один элемент меньше полной выборки. Однако при данном подходе к разбиению повышается ресурсоемкость, так как приходится выполнять обучение K раз, где K — длина полной выборки. Поэтому данный метод применим на практике, если изменение параметров модели может происходить достаточно эффективно при изменении одно элемента из обучающей выборки.
- *Скользящий контроль по k блокам* (k-fold cross-validation). В данном случае полная выборка разбивается на k непересекающихся множеств приблизительно одинаковой длины. Затем каждое из k разбиений выступает в качестве контрольной, в то время как остальные k-1 вместе составляют обущающую выборку.

В случае использования разбиения по k блокам имеет смысл выбирать разбиения случайно. Однако здесь встает проблема репрезентативности подвыборки, которая заключается в том, что подвыборка должна сохранять характеристики полной выборки. В противном случае процесс обучения модели может оказаться несогласованным. В задачах классификации для того, чтобы обеспечить некоторое статистическое «подобие» подвыборки полной выборке, рекомендуется сохранять пропорции, по которым классы распределены в полной выборке, в каждой из подвыборок.

Существуют и другие виды внешних критериев. Некоторые из них основаны на *гипотезе непротиворечивости*, согласно которой модели, обученные на различных частях

выборки должны давать в некотором приближении одинаковые результаты. В частности, используется формула средней невязки ответов алгоритмов, обученных по двум случайным разбиениям выборки:

$$Q_{ext}(\mu, X^L) = \frac{1}{L}\sum_{i=1}^{L} |a_1(x_i) - a_2(x_i)|$$

В критериях, построенных на понятии *регуляризации* используется внутренний критерий качества с некоторыми поправками. Так как применение внутреннего критерия способствует переобучению, то можно использовать свойства модели, сигнализирующие о переобучении, при построении внешнего критерия.

Например, в задаче построения линейной регрессии [6], при построении критерия регуляризации часто используется штрафное слагаемое, чаще всего равняющееся норме вектора весов построенной модели:

$$Q_{ext}(\mu, X^L) = Q_{\text{int}}(\mu, X^L) + \lambda \|\alpha\|$$

Причина использования нормы вектора в качестве регуляризационного параметра заключается в том, что одним из последствий переобучения может служить появление в модели больших положительных и отрицательных весов, что, в свою очередь, приводит к увеличению нормы.

1.2. Постановка задачи отбора информативных признаков и обзор существующих методов отбора

Предположим, что исходная выборка D состоит из n-мерных векторов признаков *x* и соответствующих им ответов *y:*

$$D = \{(x^i; y^i)\}_{i=1}^{k}$$

Необходимо среди множества признаков вектора $F = \{x_j^i\}_{j=1}^{n}$ выбрать некоторое подмножество информативных признаков.

Большинство методов отбора признаков опираются на функционалы качества, которые были описаны в предыдущем разделе, для определение степени информативности того или иного набора признаков. Если задан некий критерий качества Q(L), где L — некоторое подмножества признаков, то, минимизируя данный функционал для заданных подмножеств, мы можем выбрать оптимальный из них.

Основная проблема здесь заключается в порождении подмножеств исходного множества признаков. В зависимости от способа генерации наборов L для критерия качества методы отбора признаков могут варьироваться.

Рассмотрим некоторые из подходов:

- *Полный перебор признаков.*

  В данном случае происходит перебор всех возможных подмножеств признаков L и вычисление для каждого заданной оценки Q(L). Этот подход позволяет определить *самую* оптимальную с точки зрение выбранного функционала комбинацию признаков. Также он довольно прост в реализации.

  Однако, как и в случае полного скользящего контроля, спектр задач, которые можно решить на практике с помощью данного метода, довольно узок. Это связано с тем, что полный перебор подмножеств имеет функцию трудоемкости порядка $O(2^n)$, что в силу вычислительных способностей современных машин не поддается широкому применению.

  Тем не менее данный метод полезно рассмотреть, так как он содержит себе иллюстрацию основной схемы отбора информативных признаков.

  В алгоритме можно выделить следующие шаги:

  1. перебираем i = 1 .. n, где i — мощность подмножества признаков
  2. для текущего i ищем наилучший набор сложности i:

     $$L_i^{best} = \underset{L_i \subseteq F : |L_i| = i}{argmin} Q(L_i)$$

  3. запоминаем мощность набора i^*, при которой был минимальный $Q(L_i^{best})$:

     $$i^* = \underset{j\,:\,j \le i}{argmin} Q(L_j^{best})$$

  4. проверяем условие завершения: если выполняется $i - i^* \ge d$, то завершаем алгоритм и возвращаем $L_{i^*}^{best}$

  Параметр d в данном случае является единственным. Если на очередной итерации алгоритма оказывается так, что $Q(L_i^{best})$ не улучшается, то это может служить сигналом к тому, что точка минимума ошибки была пройдена, и началось переобучение.

- *Итерационное добавление признаков*.

Подход, реализующий итерационное добавление признаков, по сути опирается на парадигму жадной стратегии. Жадная стратегия подразумевает выбор на каждом этапе некоторого шага, оптимального с точки зрения текущего состояния, при этом все остальные шаги из текущего состояния выполнены не будут, так как нет повторного перехода в то же состояние.

При итерационном добавлении признаков на каждом шаге происходит добавление такого признака, который приведет к наибольшему уменьшению выбранного функционала качества по сравнению с текущим состоянием.

- *Итерационное удаление признаков*.

  Данный метод также относится к использующим элементы жадной стратегии. Здесь реализуется симметричный алгоритм: вместо того, чтобы начинать итерации с пустого подмножества признаков, в качестве начального набора фиксируется весь исходный набор признаков $L_0 = F$. Далее на каждом шаге удаляется тот признак, при котором оставшийся набор дает минимальное значение функционала качества Q.

  Данный метод может оказаться предпочтительнее предыдущего в случае, когда имеется априорная информация о том, информативных признаков значительно больше, чем неинформативных.

- *Последовательное удаление и добавление признаков (шаговая регрессия)*.

  Такой способ призван улучшить результат применения предыдущих двух подходов к отбору информативных признаков. Общие шаги в данном алгоритме следующие: сначала выполняется алгоритм итерационного добавления признаков с исходным пустым подмножеством признаков. В какой-то момент алгоритм останавливается и запускается метод последовательного удаления признаков, который исключает из набора избыточные признаки, добавленные алгоритмом добавления. Затем алгоритм удаления завершается и снова запускается добавление и т.д.

  В данном случае условием останова может служить прекращение или замедление уменьшения функционала для оптимального набора на текущей итерации i ($Q(L_{i^*})$) по сравнению с предыдущими итерациями.

  Также можно отслеживать состав оптимального набора признаков и останавливать алгоритм, если набор стабилизируется.

  Данный подход имеет более медленную сходимость по сравнению с методами итерационного добавления и удаления признаков в отдельности, однако на

практических задачах он зачастую дает лучшие результаты.

- *Метод ветвей и границ при обходе в глубину*.

  Если рассматривать полный перебор подмножеств признаков, то его структуру разумно представить в виде дерева (Рисунок 1) [1].

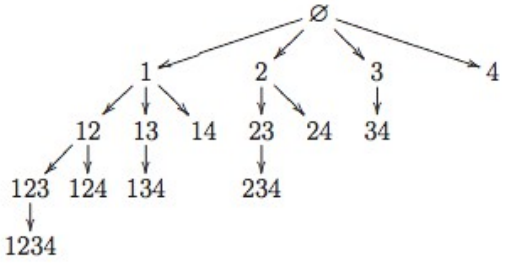

Рисунок 1. Дерево полного перебора всех возможных подмножеств признаков из множества, состоящего из 4 элементов

В данном случае порождение новых подмножеств из каждого узла дерева происходит без добавления (наращивания) признаков, имеющих меньший номер, чем текущий узел. Это делается для того, чтобы в процессе перебора не порождались наборы признаков, являющиеся перестановками друг друга.

При этом задача полного перебора все еще имеет функцию трудоемкости порядка $O(2^n)$ и неприменима для большинства практических задач. Однако при работе с древовидной структурой можно попробовать применить некоторые эвристические подходы классического метода ветвей и границ.

Во-первых, при совершении обхода в глубину можно отслеживать релевантность текущей ветки и, если по истечении нескольких итераций срабатывает условие останова, то отказываться от дальнейшего ее обхода. В качестве условие останова можно использовать следующее правило: если для нескольких последних пройденных вершин оказалось, что значение функционала качества перестало уменьшаться, либо стало возрастать, то обход текущей ветви прекращается.

Во-вторых, можно попытаться держать в приоритете обхода потенциально более информативные ветви, то есть выбирать на каждом шаге более значимый признак. В этом случае первая эвристика наиболее эффективно будет ускорять общую

сходимость алгоритма.

Для того, чтобы иметь возможность применить вторую эвристику, необходимо каким-либо образом ранжировать признаки по их единичной информативности. Сделать это можно, например, упорядочив признаки по их коэффициенту корреляции с ответами y_i .

В итоге данный подход позволяет сократить полный перебор признаков и сделать его применимым для решения практических задач, в которых изначальное количество признаков может достигать нескольких десятков.

Имеется немалое количество других методов, применяемых для отбора информативных признаков. Например, для решения данной задачи модно применить генетический алгоритм [7], который оперирует наборами признаков как популяциями в рамках дарвиновской теории эволюции.

Также в качестве подхода к отбору признаков может применяться кластеризация. В данных методах происходит разделение исходного множества признаков на определенные группы (кластеры), которые формируются согласно наложенной на пространство признаков метрике. После формирования кластеров из всех выбирают по одному признаку-представителю, каждый из которых добавляется в результирующее множество. Несмотря на то, что после выполнения кластеризации в одной из групп могут оказаться только неинформативные признаки, алгоритм по крайней мере позволяет сократить (иногда на порядок) исходное множество признаков.

Далее будет рассмотрен иной подход к отбору информативных признаков, основанный на обучении модели логистической регрессии с регуляризатором особого вида с использованием алгоритма GLMNET.

2. Методика решения задачи отбора информативных признаков

Перед тем, как перейти к описанию предложенного метода для решения поставленной задачи, необходимо привести некоторые определения.

2.1. Линейный классификатор

Для начала поставим задачу классификации и в качестве алгоритма рассмотрим линейный классификатор.

Определим исходные данные. Пусть имеется множество классов Y. Положим, что классов всего два, соответственно Y = {-1, +1}. Также задано пространство признаковых описаний объектов (n — мерных векторов) исходной выборки $X = R^n$.

Линейным классификатором для случая двух классов будет называться алгоритм, отображающий X в Y следующего вида [8]:

$$a(w, x) = sign(\sum_{i=1}^{m} w_j x_j - w_0)$$

Где w_j — вес j-го признака, w_0 - порог принятия решения.

В случае, когда мощность множества Y больше двух, линейный классификатор выражается следующим образом:

$$a(x, w) = \underset{y \in Y}{argmax} \sum_{j=0}^{n} w_{yj} f_j(x)$$

После определения множеств X и Y и выбора линейного классификатора необходимо подобрать веса в векторе w таким образом, чтобы полученная модель обладала хорошей способностью разделение объектов из X на классы.

Базовым подходом к обучению классификатора в данном случае является

минимизация эмпирического риска. Основная идея данного подхода состоит в том, чтобы подбирать веса признаков таким образом, чтобы минимизировать количество неправильно классифицированных объектов на обучающей выборке:

$$Q(w)=\sum_{i=1}^{m}[a(x_i,w)\neq y_i]\rightarrow \min_{w} \qquad (1)$$

Применяемые на практике методы обучения различаются способом решения этой задачи оптимизации.

Основная проблема, возникающая при решения задачи (1) в неизменном виде заключается в том, что минимизация дискретнозначного функционала в этом случае является эквивалентной поиску максимальной совместной подсистемы в системе неравенств. Данная задача является NP-полной, следовательно, для нее не существует оптимального решения.

Общепринятый подход к решению этой проблемы заключается в аппроксимации дискретнозначного функционала некоторой непрерывной функцией:

$[a(x_i;w)\neq y_i]\leq L(M)$, где M — отступ ($y\langle x,w\rangle$)

Теперь вместо функционала (1) минимизируется его оценка сверху:

$$Q(w)\leq Q^{*}(w)=\sum_{i=1}^{m}L(M(x_i))$$

2.2. Логистическая регрессия

Рассмотрим функцию следующего вида:

$$\sigma=\frac{1}{1+e^{-z}} \text{ , где } z\in R$$

Данная функция называется *сигмоидной* и ее поведение выглядит следующим образом [9]:

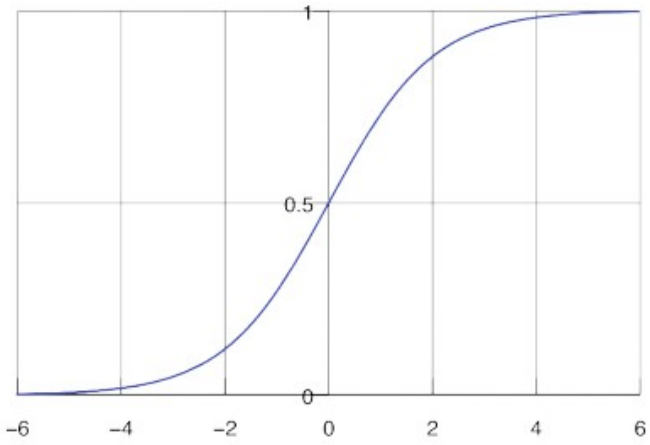

Рисунок 2. График сигмоидной функции

Как видно из графика, сигмоидная функция обладает следующими свойствами:

1. Она является отображением из множества R в интервал (0;1)
2. Стремится к 1 при $x \to +\infty$
3. Стремится к 0 при $x \to -\infty$

Теперь рассмотрим следующую проблему: пусть для целей бинарной классификации (обозначим классы как 0 и 1) требуется определить некоторую функцию h(x), которая будет принимать значения, близкие к 0 в случае, если x не принадлежит классу 1, и значения, близкие к 1 в случае принадлежности к классу 1.

Зададим данную функцию как:

$$h_\theta(x) = \sigma(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$$

Тогда можно попробовать интерпретировать вероятность принадлежности x к классу 1 как:

$$p(y=1; x, \theta) = h_\theta(x) \qquad (2)$$

Так как сигмоидная функция принимает значения от 0 до 1, то можно определить вероятность попадания x в класс 0 как:

$$p(y=0; x, \theta) = 1 - p(y=1; x, \theta)$$

Таким образом, можно определить следующий критерий классификации: если $h_\theta(x) \geq 0.5$, что эквивалентно $\theta^T x \geq 0$, то объект x причисляется к классу 1, если же $h_\theta(x) < 0.5$ (или $\theta^T x < 0$) - то к классу 0.

Рассмотрим пример бинарной классификации двумерных векторов [9]:

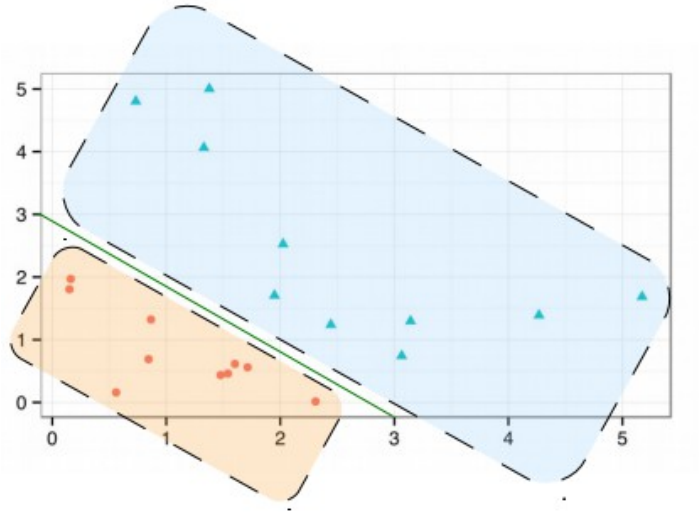

Рисунок 3. Пример бинарного разделения объектов на классы

Предположим что после обучения мы получили оценку столбца весов $\theta = [-3, 1, 1]^T$ (на рисунке отмечена соответствующая граница классов). Тогда класс 1 (верхний на рисунке), будет определяться при выполнении условия $\theta^T x > 0$, что эквивалентно $-3 + x_1 + x_2 > 0$ или $x_1 + x_2 > 3$.

Функция (2) нашла применения в задаче *логистической регрессии*.

В логистической регрессии используется следующая оценка функционала эмпирического риска:

$$Cost(h_\theta, y) = \begin{cases} -\log(h_\theta(x)), \text{если } y = 1 \\ -\log(1 - h_\theta(x)), \text{если } y = 0 \end{cases} \quad (3)$$

Для обоснования выбора данной оценки рассмотрим два случая [9]:

- y = 1

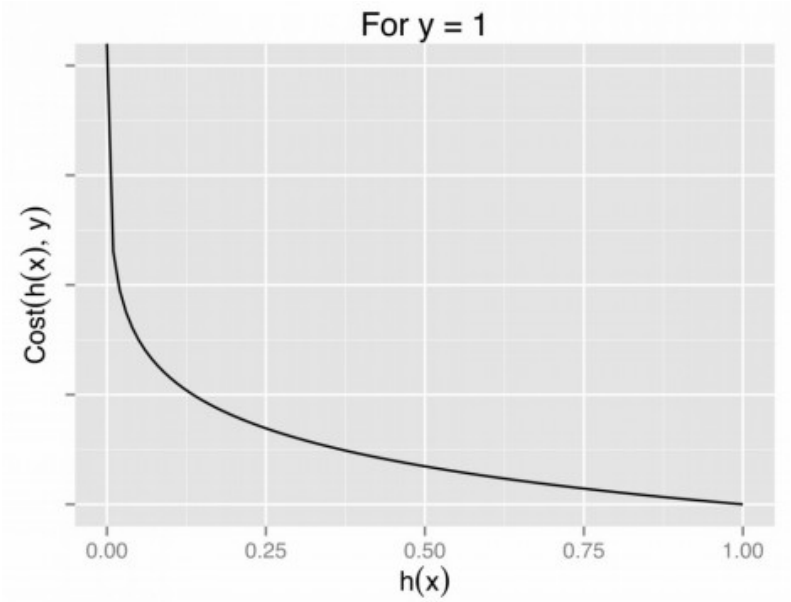

Рисунок 4. Функция стоимости при y = 1

Как видно из рисунка, если h(x) стремится к 1 (то есть алгоритм классифицирует объект как 1 и, соответственно, не ошибается), то функция Cost стремится к 0. Если же алгоритм ошибается, приписывая объект к классу 0 (то есть h(x) стремится к 0), то функция стремится к бесконечности.

- y = 0

  Здесь происходит обратная ситуация (см. Рисунок 5)

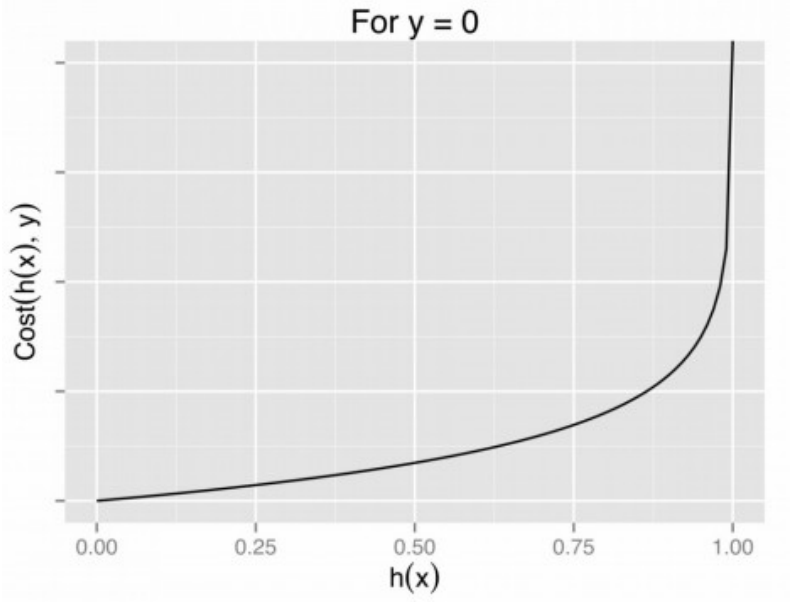

Рисунок 5. Функция стоимости при y = 0

В данном случае при стремлении h(x) к 0 алгоритм верно определяет класс и, соответственно, функция cost стремится к 0. Если алгоритм ошибочно отнес объект к классу 1, то функция cost стремится к бесконечности.

Исходя из приведенных выше соображений, можно построить следующую оценку эмпирического риска как функцию (3). Для случая двух классов (0 и 1) функцию можно записать в следующем виде:

$$Cost(h_\theta(x), y) = -y\log(h_\theta(x)) - (1-y)\log(1-h_\theta(x)) \quad .$$

Соответственно, функция Cost для обучающей выборки:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} Cost(h_\theta(x^{(i)}), y^{(i)})$$

В этом случае *оптимизационная задача логистической регрессии* выглядит таким образом:

$$\theta^* = \underset{\theta}{argmin}\, J(\theta) = \underset{\theta}{argmax}(-J(\theta)) \qquad (4)$$

После того, как вектор весов θ^* был подобран, модель можно использовать непосредственно для классификации новых объектов.

В случае бинарной классификации вычисляется вероятность принадлежности объекта к классу 1 и используется правило $h_{\theta^*}(x) > 0.5$ для определения объекта в класс 1, либо в класс 0 в противном случае.

В случае же большего количества классов используется другой прием [6].

Предположим, что мы имеем k классов. Обучим для каждого из классов свою модель, решив оптимизационную задачу (4). При этом текущий класс, для которого решается задача, играет роль класса 1 из задаче бинарной классификации, а все остальные классы — роль класса 0.

Решением будет являться набор оцененных векторов весов $\theta^*_1, \theta^*_2, \theta^*_3 \ldots \theta^*_k$. Теперь мы можем определить апостериорную вероятность принадлежности объекта x классу i, вычислив соответствующую $h_{\theta^*_i}(x)$. Таким образом, классификация происходит так:

$$predict(x) = \underset{i}{argmax}\, h_{\theta^*_i}(x) \quad \text{для i = 1..k.} \qquad (5)$$

В следующей главе рассматривается модификация логистической регрессии и метод, решающий задачу оптимизации (4), использованные в представленном решении.

2.3. Алгоритм GLMNET

Метод GLMNET (elastic-net regularized generalized linear models) был впервые представлен исследователями из Стэнфордского университета Фридманом, Хасти и Тибширани [3].

В зависимости от используемой модели GLMNET решает для них различные оптимизационные задачи. Среди основных используемых моделей выступают:

- Линейная регрессия
- Биномиальная логистическая регрессия
- Мультиномиальная регрессия

В нашем случае наибольший интерес представляет мультиномиальная (многоклассовая) регрессия, потому что, как описывалось в предыдущем разделе, она позволяет строить классификатор при числе классов, большем двух, при помощи логистической регрессии.

Алгоритм GLMNET также применяет регуляризацию в используемых моделях. Исходная задача оптимизации для GLMNET в случае мультиномиальной регрессии выглядит следующим образом:

$$\max_{\{\beta_{0\ell},\beta_\ell\}_1^K \in \mathbb{R}^{K(p+1)}} \left[\frac{1}{N} \sum_{i=1}^{N} \log p_{g_i}(x_i) - \lambda \sum_{\ell=1}^{K} P_\alpha(\beta_\ell) \right] \quad (6)$$

где K — число классов, N — длина обучающей выборки, g_i - метка класса для i-го объекта выборки, $p_l(x)$ - вероятность принадлежности x классу l.

Данная формула отличается от рассмотренной в разделе 2.2 ранее тем, что в ней добавлен *регуляризатор.*

Регуляризация — один из способов не дать вектору весов содержать больших по модулю значений.

Обычно регуляризаторы являются штрафными слагаемыми при функционалах качества и равны норме вектора:

$$Q = Q_{old} + \lambda \, \|\theta\|_p$$

Обоснование подобного подхода состоит в том, что при определении функционала

качества мы можем считать вектор весов случайной величиной и сделать некоторые предположения относительно его поведения, исходя из его распределения [5]. Соответственно, при использовании различных видов регуляризаторов можно задавать различное поведение коэффициентов.

В случае использования в качестве регуляризатора L1-нормы (метод Lasso) предполагается многомерное распределение Лапласа у весов признаков. Этот тип регуляризатора обладает несколькими важными свойствами [3, 5]:

- Большинство значений весов признаков после обучения оказывается близким к нулю;
- Происходит *отбор признаков*, то есть некоторые веса обнуляются. При этом признак, при котором оказался нулевой вес, не учитывается в модели. Количество нулевых признаков возрастает с увеличением параметра λ ;
- В случае наличия нескольких сильно коррелированных признаков при обучении с данным регуляризатором имеется тенденция к выбору одного из них и игнорированию остальных.

L2-норма в качестве регуляризатора предполагает n-мерное Гауссовское распределение весов. В частности, этот тип регуляризатора полезен при наличии большого числа зависимых признаков и способствует усреднению весов среди них [3].

Регуляризатор, примененный в формуле (6), называется elastic-net регуляризатором:

$$P_\alpha(\beta) = (1-\alpha)\frac{1}{2}||\beta||_{\ell_2}^2 + \alpha||\beta||_{\ell_1}$$

Он представляет собой компромисс между L1 и L2 регуляризацией, позволяя сочетать их свойства. Например выбор значения α немного меньшего, чем единица, позволяет регуляризатору вести себя во многом как Lasso (L1), однако при этом происходит предотвращение нежелательного поведения метода в случае существования коррелированных признаков.

Поскольку заранее неизвестно с каким параметром λ учитывать слагаемое регуляризации, то алгоритм предусматривает его вычисление для ряда значений.

Общая схема алгоритма GLMNET для решения оптимизационной задачи (6) выглядит следующим образом:

- Идет перебор значений λ по логарифмической шкале;
- Идет перебор классов l = 1..k;

- Для каждого класса происходит обучение модели методом CD (coordinate descent). Подробно метод описан в [3].

2.4. Метод отбора признаков

Для решения задачи отбора информативных признаков по прецедентам использовался алгоритм GLMNET, о котором шла речь в предыдущем разделе.

Как уже было упомянуто, при обучении модели с использованием elastic-net регуляризатора происходит отбор признаков (при параметре регуляризатора $\alpha > 0$). Это означает, что по мере увеличения параметра λ модель будет становится все более и более разряженной (содержащей меньшее число признаков с ненулевыми весами).

Предложенный метод отбора признаков состоит из нескольких шагов:

1. *Чтение входных данных*. Входные данные представляют собой выборку прецедентов и некоторые параметры алгоритма, большинство из которых в подавляющем большинстве случаев выставляются по умолчанию. Принципиален для задания лишь один параметр - α (по умолчанию устанавливается значение 0.5).
2. *Стандартизация значений признаков.* Стандартизацию (нормирование) требуется производить для того, чтобы результирующие веса признаков можно было ранжировать по значимости согласно модулю их значений.
3. *Первоначальный запуск алгоритма GLMNET.* На данном этапе производится обучение модели для каждого из классов на основе *всей* выборки из входных данных. Выходными данными алгоритма являются:
   - Значения λ, для которых производилось обучение (по умолчанию таких значений <= 100). Они необходимы для дальнейшего выбора оптимального λ;
   - Веса признаков для каждого значения λ для каждого из классов.
4. *Поиск наилучшего решения.* После того, как было произведено обучение по полной выборке и получен набор значений λ, необходимо определить какое значение λ приводит к построению лучшей модели. Для этого был применен один из критериев качества, описанных в разделе 1.1 — скользящий контроль по k блокам (по умолчанию k = 10). При разбиении полной выборки на блоки соблюдалось условие представительности каждой из подвыборок (сохранялись пропорции распределения

по классам). При выполнении скользящего контроля для каждой из k обучающих выборок снова запускался алгоритм GLMNET.

После каждого выполнения GLMNET производилось тестирование полученной модели для каждого значения λ на оставшейся части выборки (контрольной) на данной итерации. Вычисление класса для каждого объекта происходило путем максимизации апостериорной вероятности (5). При этом происходило обновление вектора ошибок классификации на контрольной выборке (прибавление текущей ошибки к уже накопленной для каждого λ). Под ошибкой классификации подразумевается процент неверных ответов алгоритма.

После k итераций производилось окончательное вычисление ошибок классификации для каждой из λ путем деление накопленной ошибки на k.

Наилучшим признавалось решение с наименьшей ошибкой.

Для подбора параметра α скользящий контроль не используется, так как он устанавливается из априорных соображений.

5. *Вывод результатов.* Для значения λ , при котором в ходе кросс-валидации были построены модели, давшие минимальную среднюю ошибку на контрольных данных, выбираются веса для каждого из классов, которые были построены в ходе обучения на *полной* выборке (на шаге 3). В качестве выходных данных выступают наиболее информативные признаки для каждого из классов (имеющие наибольшие абсолютные значения). Их количество задается отдельным параметром (по умолчанию их 10).

3. Программная реализация метода отбора признаков

Основной целью программной реализации метода было создание консольного приложения, взаимодействие с которым было бы возможно в рамках информационно-аналитической системы Института Металлургии им. Байкова РАН. Однако создание библиотеки отбора информативных признаков также является важным результатом данной работы. Помимо консольной версии необходимо было реализовать графический пользовательский интерфейс для базовых возможностей тестирования метода.

Исходные коды приложения и библиотек содержаться в Приложении Д.

3.1. Используемые технологии

В качестве основного языка разработки был выбран язык Java. Одной из причин такого решения является отсутствие подобных библиотек, реализующих отбор признаков с использованием алгоритма GLMNET.

В качестве стороннего ресурса была использована библиотека, реализующая алгоритм Coordinate descent для мультиномиальной регрессии с elastic-net регуляризатором, написанная на языке Fortran 90. Ее вызов из среды Java происходит посредством технологии JNI и языка C. Данная библиотека написана создателями GLMNET исследователями Хасти и Фридманом [3].

Для вызова из них нативных функций необходимо снабдить программу скомпилированными под требуемую систему библиотеками. Для этих целей использовались компиляторы MinGW под 64-разрядную ОС Windows и стандартный gcc на Mac OS X 9.2 Mavericks.

3.2. Форматы входных и выходных данных

В консольной версии приложения использовалась передача данных через параметры командной строки. Среди них были путь к файлу с данными, а также параметры самого алгоритма. В качестве формата файла с исходной выборкой использовался формат csv.

Выходные данные представляли из себя порядковые номера отобранных признаков согласно из расположению в исходном файле.

В приложении с пользовательским интерфейсом форматы несколько различаются.

Подробнее о формате входных и выходных данных в Приложении Г.

4. Апробация программы

Для оценки результатов работы метода воспользуемся версией приложения с графическим пользовательским интерфейсом.

В качестве выборки для анализа используем данные из реальной задачи (см. Приложение А).

В начале загрузим исходные данные:

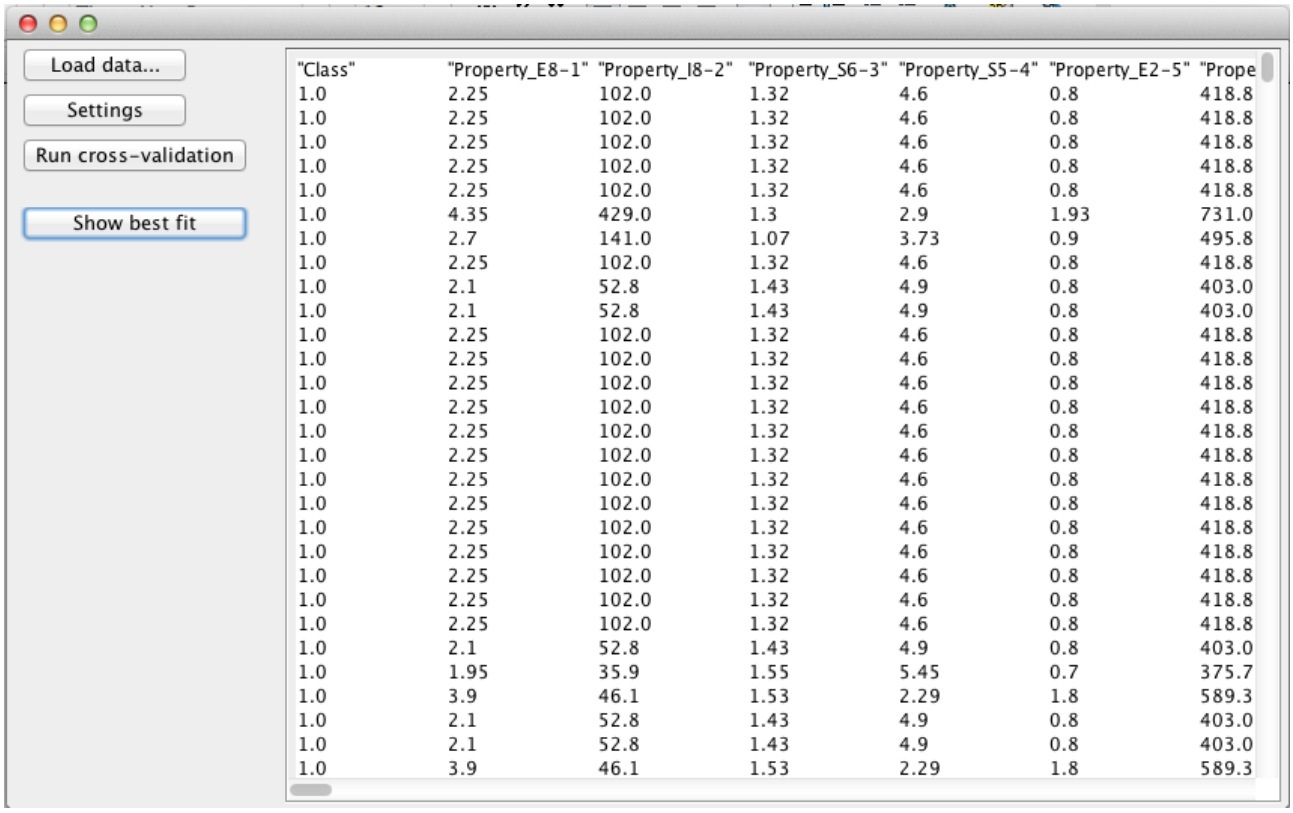

Рисунок 6. Загруженные данные в главном окне программы

Заметим, что в данной выборке имеются алгебраические функции от изначальных элементарных признаков, а именно A5(1)*M1(4), C1(1)-C2(1) и A5(1)*A5(3). Этот прием называется *синтезом признаков* и наряду с отбором является одним из способов сокращения признакового пространства. В скобках указаны номера элементов, свойства которых они характеризуют. Они введены в выборку искусственно в силу неких априорных соображений, связанных с предметной областью, и являются основными кандидатами на роль информативных признаков.

Затем запустим кросс-валидацию с параметрами по умолчанию (см. раздел 2.4) за

исключением α , которое равняется 0.8.

После двух с половиной минут вычислений появляется результат, который можно просмотреть, нажав на кнопку «show best fit»:

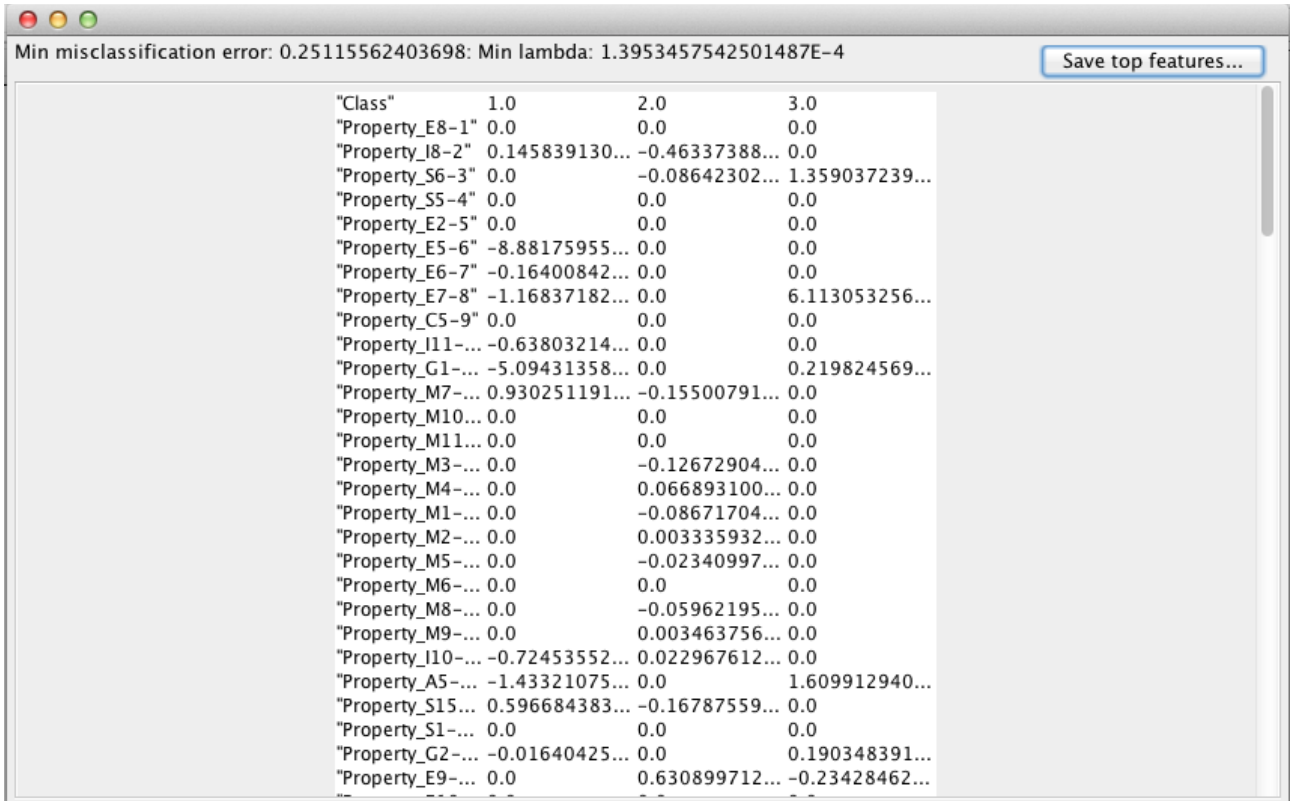

Рисунок 7. Окно результата

В этом окне отображены веса обученной модели для каждого из классов, имеющей наименьшую ошибку (в данном случае равную 0.25).

Чтобы сохранить N (по умолчанию 10) самых информативных признаков, следует нажать кнопку «Save top features...»:

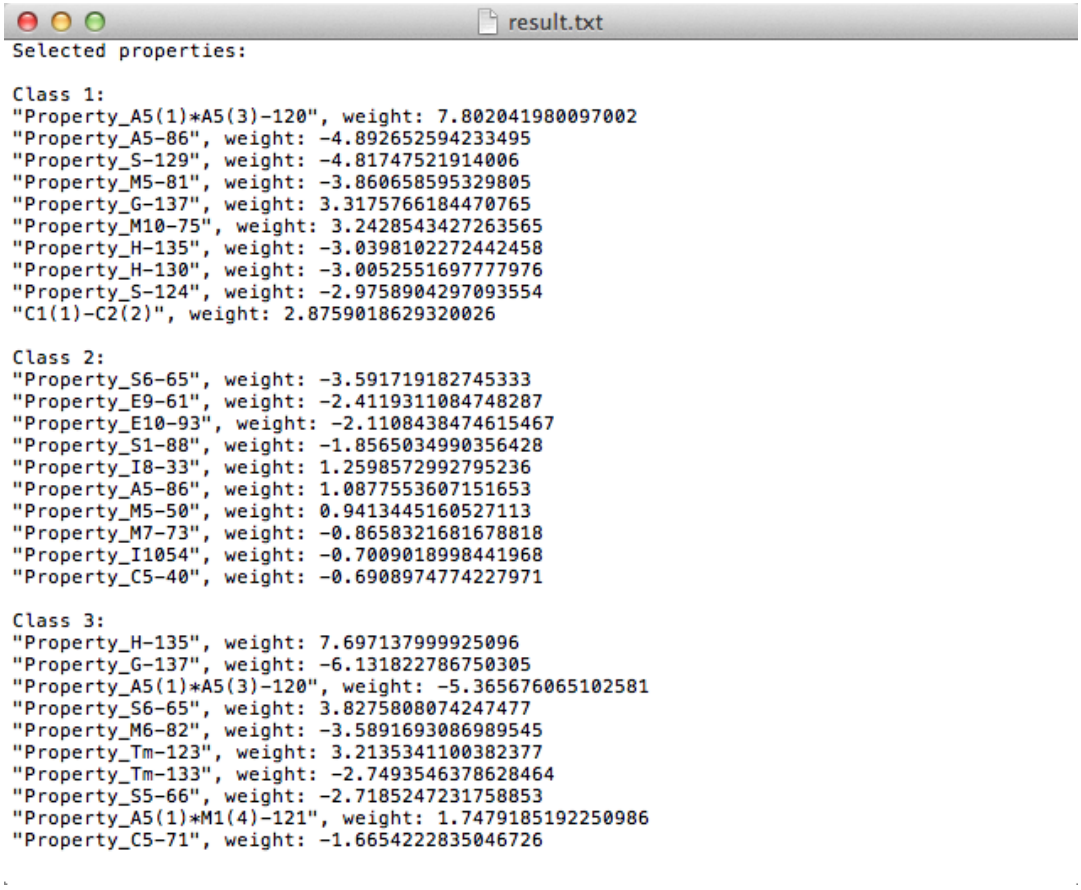

Рисунок 8. Найденные нформативные признаки для каждого класса

Адекватность полученных результатов можно попытаться охарактеризовать несколькими способами:

1. Оцененная ошибка модели. В данном случае она составляет 0.25, что является вполне достойным практическим показателем. Так что коэффициенты данной модели являются с этой точки зрения лучшим показателем информативности признаков.
2. Как уже было сказано выше, в выборку включены обобщенные признаки, которые являются алгебраическими функциями от других элементарных признаков и потенциально должны повысить качество выборки. Из Рисунка 8 видно, что все из этих признаков были включены в ответ, несмотря на то что один из них оказался гораздо информативнее других (A5(1)*A5(3)).
3. Сравнение с результатами других алгоритмов. В данном случае речь идет об алгоритме CCP [10]. После предоставления автору результатов работы данного метода оказалось, что самыми информативными как раз оказались синтезированные свойства:

Свойство (A5(1)*M1(4)) - вес: 0,605492
Свойство (C1(1)-C2(1)) - вес: 0,296012
Свойство (A5(1)*A5(3)) - вес: 0,098496

Рисунок 9. Информативные признаки, отобранные алгоритмом CCP

Еще одним заметным результатом анализа является тот факт, что свойство A5(1)*A5(3) входит в тройку информативных признаков в классе 1 и 3, причем с противоположными знаками весов. Это означает, что данные классы являются своего рода антагонистами с точки зрения этого свойства.

Заключение

Представленная выпускная квалификационная работа была посвящена задаче отбора информативных признаков. Данная задача является важной частью процесса предобработки данных, целью которого является повышения качества алгоритмов машинного обучения. Сам отбор представляет собой процесс отсеивания нерелевантных признаков, то есть таких, которые не несут в себе информации в контексте некоторой задачи машинного обучения, либо являются шумовыми. На практике к данному подходу прибегают часто ввиду того обстоятельства, что при составлении выборки не всегда имеется возможность определить важность того или иного признака.

В процессе выполнения ВКР были получены следующие результаты:

1. Анализ предметной области и методов отбора информативных признаков;
2. Реализация библиотеки отбора признаков с использованием алгоритма GLMNET на языке Java;
3. Реализация консольной версии приложения с целью его интеграции с информационно-аналитической системой Института Металлургии им. Байкова РАН в составе модуля отбора признаков;
4. Реализация версии приложения с графическим пользовательским интерфейсом для базового тестирования;
5. Анализ работы алгоритма на примере реальной задачи;

Дальнейшие направления исследования использования алгоритма GLMNET для отбора информативных признаков могут быть направлены на изучение существующих модификаций данного алгоритма и их применения для улучшения результатов отбора. Также может представлять интерес исследование совместного использования нескольких подходов к отбору информативных признаков.

Источники

[1] Лекции по методам оценивания и выбора моделей [Электронный ресурс] : методические материалы / д. ф.-м. н. Воронцов К.В. - электрон. текст. дан. - 26 с. - Режим доступа: http://www.ccas.ru/voron/download/Modeling.pdf (дата обращения: 03.05.2014)

[2] Data mining. / Witten I., Frank E., Hall M. - Burlington: Morgan Kaufmann Publishers. - 2011. - 629 с.

[3] Regularization Paths for Generalized Linear Models via Coordinate Descent / Friedman J., Hastie T., Tibshirani R. // Statistical Software. - 2009. - с. 1–22.

[4] Методы выбора регрессионных моделей / Стрижов В.В., Крымова Е.А. - Вычислительный центр РАН. - 2010. - с. 60

[5] Лекции по машинному обучению. Линейные методы классификации. Логистическая регрессия [Электронный ресурс] : видеолекция / д. ф.-м. н. Воронцов К.В. - компания «Яндекс». - электрон. граф. дан. - Режим доступа: http://shad.yandex.ru/lectures/machine_learning_7.xml(дата обращения: 17.04.2014)

[6] Введение в методы статистического обучения [Электронный ресурс] : конспект лекций / Мерков А.Б. - электрон. текс. дан. - Режим доступа: http://www.recognition.mccme.ru/pub/RecognitionLab.html/slbook.pdf (дата обращения: 07.05.2014)

[7] Генетические алгоритмы [Электронный ресурс] : учебно-методическое пособие / Панченко Т. В. - электрон. текст. дан. - 88 с. - Режим доступа: http://mathmod.aspu.ru/images/File/ebooks/GAfinal.pdf (дата обращения: 03.05.2014)

[8] Линейный классификатор [Электронный ресурс] : тематический информационный портал . - электрон. текст. дан. - Режим доступа: http://www.machinelearning.ru/wiki/index.php?title=Линейный_классификатор (дата обращения: 04.05.2014)

[9] Classification using Logistic Regression [Электронный ресурс] : презентация / Andrew Ng, Ingmar Schuster, Patrick Jähnichen . - электрон. текст. дан. - 29 с. - Режим доступа: http://asv.informatik.uni-leipzig.de/uploads/document/file_link/530/TMI05.2_logistic_regression.pdf(дата обращения: 03.05.2014)

[10] An Optimal Ensemble of Predictors in Convex Correcting Procedures / Senko A.V. // Mathematical theory of pattern recognition. - 2009. - с. 465 — 468