Moscow Institute of Physics and Technology (National Research University)

As a manuscript

Bartunov Sergey Olegovich

Nonparametric probabilistic models and inference methods

PhD Dissertation Summary
for the purpose of obtaining academic degree
Doctor of Philosophy in Computer Science

Moscow — 2023

The **PhD dissertation was prepared at** Moscow Institute of Physics and Technology (National Research University).

Academic supervisor: Dmitry Petrovich Vetrov, Candidate of Sciences, Professor,
Constructor University.

Contents

Page

Abstract

This dissertations studies the problems of building nonparametric probablistic machine learning models. The standard inference techniques applied in parametric models with conjugate priors are inapplicable in this setting which motivates development of new, efficient methods of approximate inference that exploit models structure. A few important cases of such models are considered in this work: a mixture model for interdependent objects based on sequential distance-dependent Chinese restaurant process, a nonparametric word vector model with automatic extraction of senses of ambiguous words in natual languages and a non-conjugate variational autoencoder generative model for images with few-shot learning abilities. For all three cases new variational inference algorithms are proposed that make the models computationally efficient and therefore expand their practical applications which is demonstrated experimentally.

Introduction

Probabilistic modelling has become one of the most important tools of machine learning. [1]. On the one hand, this was caused by the deep connection to probability theory in the language of which many concepts and results in the theory of machine learning are formulated [2; 3]. On the other hand, which is equally important, probabilistic formalism appeared to be quite convenient for specifying machine learning models and allowed a universal way of constructing new models as well as reusing existing ones [4].

One of main tools in this formalism is *Bayesian approach* which consists in decomposing a joint distribution over model's variables (e.g. hidden parameters and observations) into *a prior distribution* and a *likelihood model*. This substantially simplifies model description and allows to easily add new variables which would only require defining the corresponding conditional distributions. Besides, such decomposition can often reflect real causal relationships between variables, as far as the generative process is concerned [5]. In addition, in the context of many problems and models a prior and a likelihood distributions can have a quite natural distinction in their purpose; for example, prior may serve as a *regularizer* and likelihood as a *loss function* or a *reward model* [6].

Thus, almost all areas of machine learning benefit from an account of uncertainty that the probabilistic approach offers. However, there is a class of problems where probabilistic models are qualitatively differ from their analogs, in particular, when constructing *nonparametric models*. Nonparametric models cannot be mathematically formulated as computations depending on parameters of some fixed dimensionality. An example may be the well-known nearest neighbours classification model, where the training examples play the role of model parameters which addition automatically increases the parameter space. In a number of cases such property not only allows formulation of simpler models, but is also *principally necessary* to adequately describe the observed data.

Among such situations we will mainly consider the following ones: firstly, when a parametric model definition is a priori impossible or cumbersome (e.g. defining a mixture model under the unknown number of mixture components) and secondly, when very few training examples are available and fast learning is required as new observations arrive, which is most often implemented exactly by nonparametric models.

In this dissertation both these cases are thoroughly studied. The first one – from the point of view of the *nonparametric Bayesian approach* in which a specially constructed prior distribution provides *structural regularization* on the *model complexity* and a compromise between the model complexity and the fit of data is achieved via probabilistic inference [7]. The second case is studied on the example of a classical problem of learning a neural generative model of images with the ability of fast *incremental learning*. Since traditional methods for training neural networks are all based on gradient optimization, fast incremental learning in generative models is arguably more convenient to implement using a specialized nonparametric architecture, also built upon probabilistic inference.

Besides the model definition itself, equal difficulty in applying probabilistic nonparametric approach in practice lies in performing *probabilistic inference*, i.e. accessing posterior distribution over unobserved variables given the observed ones. In the general case, exact probabilistic inference is an NP-hard problem [8; 9] which makes it impossible in practice even for moderately large models. This fundamental issue acts as a motivation for developing *approximate inference* methods which approximate the true posterior distribution with another one, typically, having a simpler form but still good enough to be used as a proxy and, thus, more affordable for computations.

One of the fastest developing research directions currently is *variational inference* in which approximate inference is formulated as an optimization problem of finding an approximation that minimizes a certain distribution approximation criterion [10]. Such view on the problem often leads to emergence of simple but still significantly more efficient algorithms than those derived in the Monte-Carlo family, which currently concedes in parallelization and scaling aspects [11]

Being such a popular and efficient approach, variational inference has been studied in the context of a number of standard probabilistic models, from the simplest Bayesian logistic regression to modern deep generative models. Nevertheless, the experience gathered so far may be inapplicable for non-standard models differing significantly from those already studied which require developing radically new techniques since naive application of standard methods leads to quite inefficient optimization procedures. Two classes of such problems studied in this dissertation is *non-conjugate* and *nonparametric* probabilistic models.

Non-conjugacy of a prior and a likelihood is manifested by the corresponding posterior distribution not being in the same parametric family as the prior. This common phenomenon substantially preventing efficient inference currently attracts a lot of researchers attention due to the popularity of neural network based likelihood models, which is specifically studied in this dissertation.

The aim of the work is study and development of nonparametric probabilistic models as well as development of variational inference algorithms for applying those. The considered models are purposed for solving important nonparametric data modelling problems: handling dependant observations, automatic extraction of vector representations for word senses and fast learning in deep generative models. Applying these models must improve performance criteria used in each of the considered applications, which puts certain requirements on the develop inference algorithms such as computational efficiency and adequacy of the posterior approximation.

Main results:

1. Derivation of a variational inference algorithm for sequential distance-dependent Chinese restaurant process.
2. Development of a nonparametric Bayesian model based on Dirichlet process for learning vector representations for word senses in a natural language. Derivation of an efficient parallel learning algorithm for the model based on stochastic variational inference method.
3. Using principles of meta-learning and nonparametric probabilistic modelling for designing a deep generative model with fast incremental learning capabilities and a corresponding neural variational inference model.

Personal contribution. Main results have been obtained by the author himself, some results from the chapter 2 were obtained in collaboration with Dmitry Kondrashkin and Anton Osokin.

Publications and approbation of work

Author’s publications towards the dissertation

1. **Sergey Bartunov** and Dmitry Vetrov. Variational inference for sequential distance dependent Chinese restaurant process. *31st International Conference on Machine Learning, ICML 2014*, 2014, pp. 3259–3267.
2. **Sergey Bartunov**, Dmitry Kondrashkin, Anton Osokin, and Dmitry Vetrov. Breaking sticks and ambiguities with adaptive skip-gram. *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016*, 2016, pp. 130–138.

3. **Sergey Bartunov** and Dmitry Vetrov. Few-shot generative modelling with generative matching networks. *21st International Conference on Artificial Intelligence and Statistics, AISTATS 2018*, 2018, стр. 670–678.

Results of the dissertation have been used in the following publications

1. **Sergey Bartunov**, Jack Rae, Simon Osindero, and Timothy Lillicrap. Meta-Learning Deep Energy-Based Memory Models. *International Conference on Learning Representations*, 2019. **(Bartunov2019)**
2. Jack Rae, **Sergey Bartunov**, and Timothy Lillicrap. Meta-learning neural bloom filters. *International Conference on Machine Learning*, 2019, стр. 5271-5280. **(Rae2019)**
3. **Sergey Bartunov**, Fabian Fuchs, and Timothy Lillicrap. Equilibrium aggregation: Encoding sets via optimization. *Uncertainty in Artificial Intelligence*, 2022, стр. 139-149. **(Bartunov2022)**

The developments of this work have been further applied in a broader spectrum of problems that are beyond probabilistic models and inference in them. The close connection between the memory mechanisms and few-shot learning allowed to construct a non-probabilistic model of associative memory for sensory data **(Bartunov2019)** and also a neural analogue of Bloom filter [12] which can adapt to different distributions of stored data **(Rae2019)**. Finally, variational inference techniques in generative models over sets have inspired the new approach to processing sets, sequences and graphs in deep learning which was then published in **(Bartunov2022)** and generalized similar methods based on the attention mechanism in neural networks.

Presentations at conferences and seminars

1. The 31st International Conference on Machine Learning (ICML 2014), Beijing, China, 22.06-24.06.2014. «Variational inference for sequential distance dependent Chinese restaurant process».
2. The 19th International Conference on Artificial Intelligence and Statistics (AISTATS 2016), Cadiz, Spain, 9.05-11.05.2016. «Breaking sticks and ambiguities with adaptive skip-gram».
3. The 21st International Conference on Artificial Intelligence and Statistics (AISTATS 2018), Playa Blanca, Lanzarote, Canary Islands, 9.04-11.04.2018. «Few-shot generative modelling with generative matching networks».

Chapter 1. Variational inference for sequential distance-dependent Chinese restaurant process

In the first chapter a variational inference algorithm is described for nonparametric Bayesian models based on sequential distance-dependent Chinese restaurant process (seqdd-CRP) [13]. Seqdd-CRP has been proposed as a generalization of Chinese restaurant process [14] for nonparametric modelling of dependant data where between each pair of observations (i, j) a distance $d_{ij} \geqslant 0$ is defined, such that $d_{ii} = 0$ and $i > j \implies d_{ij} = \infty$.

Seqdd-CRP is traditionally introduced via the metaphor of a restaurant which customers $i = 1, 2, \ldots$ (representing observations) are entering. Each customer i chooses an already entered customer $c_i = j, j \leqslant i$ to share a table with them. Shall the customer choose himself, i.e. $c_i = i$, they sit a new table. Formally the customer assignment process is defined with the following probability:

$$p(c_i = j | f, D, \alpha) \propto \begin{cases} f(d_{ij}), & j > i, \\ \alpha, & i = j, \\ 0, & j < i, \end{cases} \tag{1.1}$$

where $f(d) \geqslant 0$ - is a non-negative distance transformation function and $\alpha > 0$ - is a hyperparameter governing the frequency of new tables emergence.

Customer assignments $\mathbf{c}$ formed according to this process induce a *table assignment* $z(\mathbf{c})$ consisting of $K(\mathbf{c})$ tables, each of which is associated with a modelled group of observations such as a cluster, mixture component etc. From the generative process point of view, for each table $k = 1, \ldots, K(\mathbf{c})$ a corresponding parameter $\theta_k \sim G$ is sampled from some *base measure* G which in its turn controls generation of all observations having $z_i(\mathbf{c}) = k$. For notational convenience, one can say that each customer i already has his own table θ_i defined but they might not choose to sit at it.

In most cases, a base measure plays a role of simply some prior distribution over parameters $p(\theta)$ and generation of observations is defined via the chosen likelihood model $p(x|\theta)$. In other words, a nonparametric mixture model based on seqdd-CRP can be described with the following joint distribution:

$$p(\mathbf{x}, \mathbf{c}, \theta | f, D, \alpha) = \left[\prod_{i=1}^{N} p(c_i | f, D, \alpha) \right] \left[\prod_{j=1}^{N} p(\theta_j) \prod_{z_i(\mathbf{c}) = j} p(x_i | \theta_j) \right]. \tag{1.2}$$

One can note that such formulation with the full set of N tables is fully equivalent to the original one in terms of the marginal likelihood $p(\mathbf{x}|f, D, \alpha)$.

Seqdd-CRP generalizes the classical Chinese restaurant process (and the closely related Dirichlet process) to the case of dependent observations providing a flexible family of priors on observation groups using only information about pairwise distances. Thus, seqdd-CRP combines the existing strengths of CRP such as automatic adjustment of model complexity with addition of new data and features of sequence models such as Hidden Markov Model [15].

This functionality comes at a price of combinatorial explosion when trying to perform exact inference on the customer assignments $\mathbf{c}$. In the original paper, [13] use Gibbs sampling as an approximate inference method with all inherent downsides in terms of convergence speed and computational efficiency. To overcome those, we propose a variational inference algorithm which allows to obtain a fully-factorized variational approximation $q(\mathbf{c}, \theta) = \prod_{k=1}^{N} q(\theta_k) \prod_{i=1}^{N} q(c_i)$, optimal from the Kullback-Leibler divergence perspective:

$$\mathrm{KL}(q(\mathbf{c}, \theta) || p(\mathbf{c}, \theta | \mathbf{x}, f, D, \alpha)) \to \min_q . \tag{1.3}$$

The corresponding variational lower bound on the marginal likelihood is then written as:

$$\begin{aligned} \log p(\mathbf{x}|f, D, \alpha) &\geqslant \mathcal{L}(q) \\ &= \mathbb{E}_{q(\mathbf{c},\theta)} \Bigg[\sum_{i=1}^{N} \big(\log p(c_i|f, D, \alpha) - \log q(c_i) \big) + \\ &\qquad \sum_{j=1}^{N} \Big(\log p(\theta_j) - \log q(\theta_j) + \sum_{i=1}^{N} z_{ij}(\mathbf{c}) \log p(x_i|\theta_j) \Big) \Bigg], \end{aligned} \tag{1.4}$$

where $z_i(\mathbf{c}) = j$ is equivalent to $z_{ij}(\mathbf{c}) = 1$ и $z_{ik}(\mathbf{c}) = 0, k \neq j$.

The main difficulty in computing the lower bound and obtaining the variational approximation is in efficient computation of the expected assignment of a customer i to a table j or $\mathbb{E}_{q(\mathbf{c})} z_{ij}(\mathbf{c})$. In this work we managed to express this quantity via *Laplacian* of the random graph constructed in seqdd-CRP.

Denote the adjacency matrix in this graph as A, where $A_{ij} = \mathbb{1}[c_i = j]$ и $A_{ii} = 0$, and the table assignment matrix as $z(\mathbf{c}) = (z_{ij}(\mathbf{c}))_{1 \leqslant i \leqslant N, 1 \leqslant j \leqslant N}$, which contains table assignments for each customer simultaneously, the mathematical expectation of the

Algorithm 1: Variational inference for seqddCRP

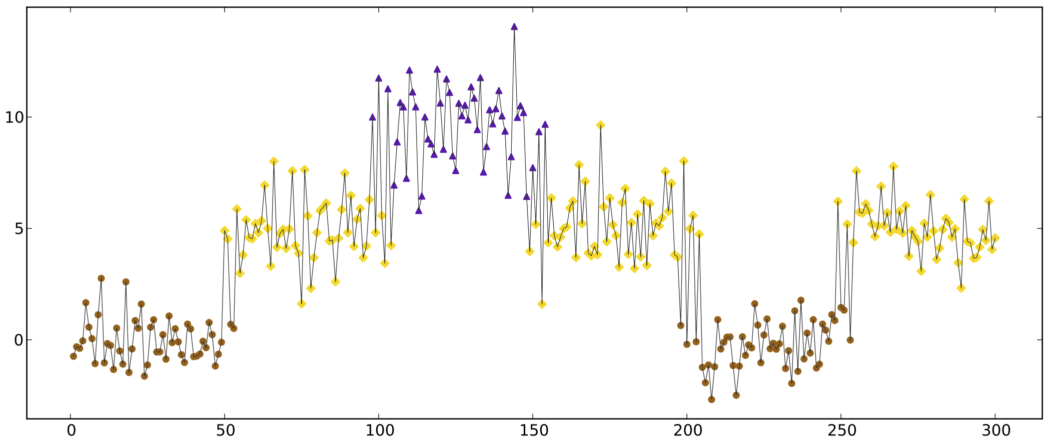
```
Data: Observations $\mathbf{x}$, initial $q(\mathbf{c})$ and $q(\theta)$, hyperparameters $\eta = (f, D, \alpha)$;
Compute $R = (I - \mathbb{E}_{q(\mathbf{c})}A)^{-1}$;
do
    for $i = 1$ to $N$ do
        Initialize zero vector $a_i \in \mathbb{R}^N$;
        for $k = 1$ to $i$ do
            $a_{ik} \leftarrow \sum_{i \geqslant i} R_{si} \mathbb{E}_q \log p(x_s|\theta_k)$;
        end
        Initialize $\gamma_{ij} = \log p(c_i = j|\eta)$ for all $j$;
        for $j = 1$ to $i$ do
            $\gamma_{ij} \leftarrow \gamma_{ij} + \sum_{k \leqslant i} a_{ik} R_{jk} q(c_k = k)$;
        end
        Update $q(c_i = j) \propto \exp(\gamma_i)$;
        Perform rank-1 update to $R$ by Woodbury formula;
    end
    for $k = 1$ to $N$ do
        Update $q(\theta_k) \propto p(\theta_k) \prod_{i \geqslant k} p(x_i|\theta_k)^{R_{ik} q(c_k = k)}$;
    end
while not converged;
```

latter can be computed as

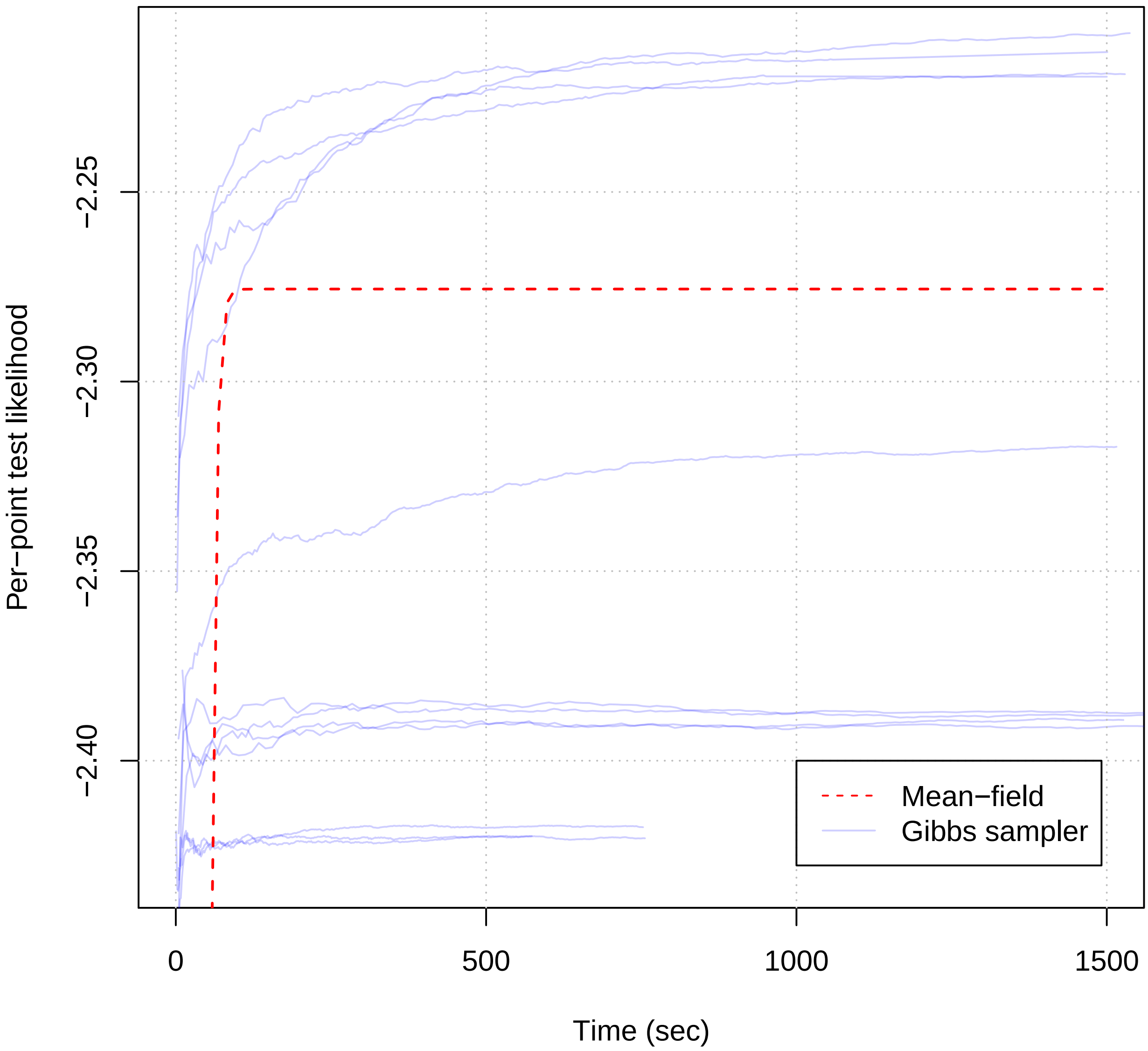
$$\mathbb{E}_{q(\mathbf{c})} z(\mathbf{c}) = (I - \mathbb{E}_q A)^{-1} \begin{pmatrix} q(c_1 = 1) & 0 & \dots & 0 \\ 0 & q(c_2 = 2) & \dots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & \dots & 0 & q(c_N = N) \end{pmatrix}.$$

Using this formula, we derive the variational inference algorithm 1. A singe iteration of this algorithm has complexity of $O(N^3)$, which is the same as for computing the variational lower bound due to inversion of a full-rank matrix.

The proposed inference algorithm has been tested on a number of tasks. Figure 1.1 contains results from applying the new variational inference algorithm and the originally proposed Gibbs sampler on synthetic data for a single-dimensional mixture model with time dependencies and stochastic switching between mixture components. As one can see on the Figure 1.1б, variational inference is capable of delivering predictive



a) Example data. Colors denote different mixture components.



б) Test likelihood as a function of number of iterations for the proposed variational inference method (Mean field) and Gibbs sampler.

Figure 1.1 — Mixture model with time dependencies.

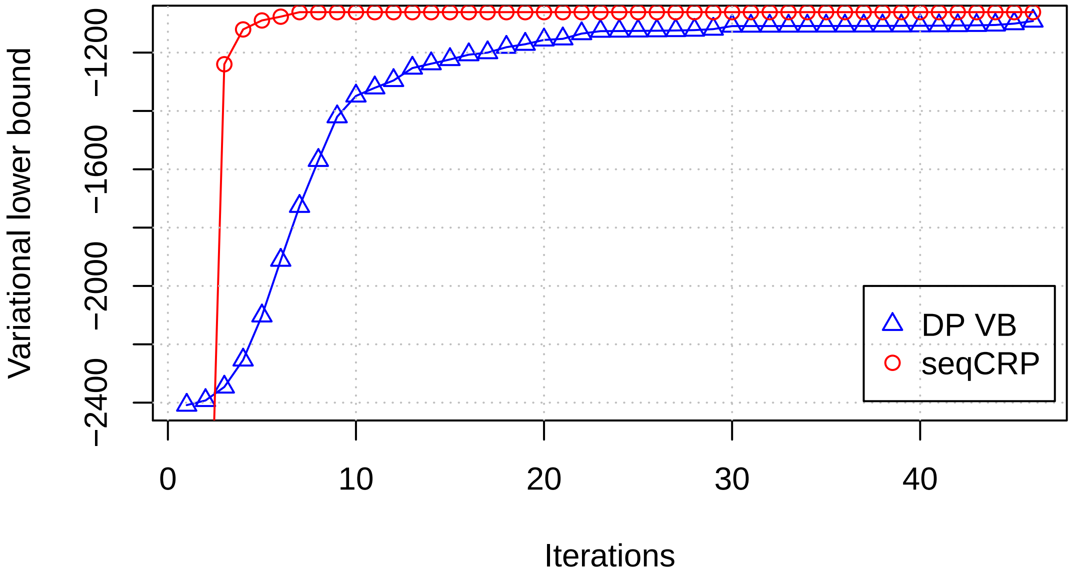


Figure 1.2 — Variational lower bound of the Gaussian mixture model as computed by the classical truncated stick-process based variational algorithm (DP VB) and the proposed algorithm using seq-CRP representation (seqCRP).

efficiency comparable to the best performing Markov chains produced by the Gibbs sampler while showing better empirical convergence speed.

We have also performed an empirical study of the proposed algorithm for the problem of inference in a simple Dirichlet process Gaussian mixture model. Since with a constant decay function $f(d) = 1$ seqdd-CRP is equivalent to the standard Dirichlet process for which variational inference algorithms have already been developed [16], a comparison between various inference algorithms is of some interest from both theoretical and practical points of view.

As shown on the Figure 1.2, the variational approximation (1.3) is better at describing the true posterior distribution and algorithm 1 empirically has a much better convergence speed.

Chapter 2. Bayesian nonparametric model for learning vector word representations

In the second chapter we consider the problem of learning vector representations for different meaning of words in a natural language. This problem currently possesses a high actuality due to the great efficiency of vector word representations in various natural language processing tasks, on one hand, and to the fundamental language *ambiguity* immanent to most natural languages. Thus, the same word «apple» may mean a fruit or refer to the famous technology company and hence require different processing depending on the context.

The proposed in this work Adaptive Skip-Gram model or AdaGram is based on the principles of Bayesian nonparametrics and allows to automatically discover a number of different meanings for each word with a required *semantic resolution*. Thus, for each word in the dictionary $v \in 1, \dots, V$ is learned an infinite number of D-dimensional real vectors or *prototypes* $\text{in}_{v,k} \in \mathbb{R}^D, k = 1, 2, \dots$. Then the conditional model for predicting context words $\mathbf{y}$ based on the current word x and its meaning z is defined as:

$$p(\mathbf{y}|x, z = k, \theta) = \prod_j p(y_j|x, z = k, \theta), \tag{2.1}$$

$$p(v|w, z = k, \theta) = \prod_{n \in \text{path}(v)} \sigma(\text{ch}(n)\text{in}_{w,k}^T \text{out}_n),$$

where $\text{out}_n \in \mathbb{R}^D$ – us a vector representation for a node n in the binary tree where leaves are words $1, 2, \dots, V$. Function $\text{path}(v)$ deterministically maps each word v into a path in this tree, and function $\text{ch}(n)$ returns $+1$, if node n is a left child and -1 otherwise. Function $\sigma(x) = 1/(1 + \exp(-x))$ is the sigmoid function which maps real numbers into $[0, 1]$ interval. One can show that a conditional model defined this way is a correctly defined discrete distribution over dictionary elements which allows efficient computation of the conditional probability even for a large dictionary [17].

Prototypes for each word w are modelled independently using Dirichlet process [14] in the stick-breaking form [18]:

$$p(\beta_{wk}|\alpha) = \text{Beta}(1, \alpha), \quad k = 1, 2, \dots$$

$$p(z = k|w, \beta) = \beta_{wk} \prod_{r=1}^{k} (1 - \beta_{wr})$$

$$p(\theta_{wk}) = \text{Uniform}(\theta_{wk}).$$

Therefore, the number of prototypes for each word is not fixed a priori but can be inferred from data. Parameter α adjusts the expected number of prototypes each of which is assumed to be distributed according to the improper prior uniform over $\mathbb{R}^D$.

Learning in the model is expressed as maximization of the marginal likelihood of the observed sequence of words $o_1, o_2, \dots o_N$ split into independent word predictions of form (2.1). For each word indexed by i the corresponding training example is the current word $x_i = o_i$ and its context $\mathbf{y}_i = o_{t|t-i|\leqslant C/2}$ of length C (ignoring words on each end of the sequence for notational convenience). Thus, the likelihood of all observed contexts is

$$\log p(Y, \theta | X, \alpha) = \log p(\theta) + \sum_{i=1}^{N} \sum_{j=1}^{C} \log p(y_{ij} | x_i, \theta, \alpha). \tag{2.2}$$

Since direct optimization of the likelihood is computationally intractable due to the necessity of performing full Bayesian inference over hidden variables, a variational EM-algorithm with the fixed upper bound on the number of prototypes T has been employed for learning in AdaGram. Thus, the variational inference over the hidden variables has the following form:

$$q(B, Z) = \prod_{i=1}^{N} q(z_i) \prod_{w=1}^{V} \prod_{k=1}^{T} q(\beta_{wk}) \prod_{k>T} \bar{q}(\beta_{wk}), \quad \bar{q}(\beta_{wk}) = \text{Beta}(1, \alpha).$$

The corresponding variational lower bound can be written as:

$$\begin{aligned}\mathcal{L}(q, \theta) =& \mathbb{E}_q \left[\sum_{w=1}^{V} \sum_{k=1}^{T} \log p(\beta_{wk} | \alpha) - \log q(\beta_{wk}) \right. \\ &\left. + \sum_{i=1}^{N} \left(\log p(z_i | x_i, \beta) - \log q(z_i) + \sum_{j=1}^{C} \log p(y_{ij} | x_i, z_i, \theta) \right) \right] \\ &+ \text{const.}\end{aligned}$$

For optimizing the variational lower bound over the variational approximation $q(B, Z)$ and all vector representations θ a streaming algorithm 2 is proposed which is based on the stochastic variational inference method [19]. Such algorithm has the asymptotic complexity analogous to the original Skip-gram learning algorithm (with respect to N) and in practice works no more than T times slower. A sparse implementation of the algorithm can be practically independent of the maximal number of prototypes T.

Algorithm 2: Training AdaGram model

```
Data: training set $\{(x_i, \mathbf{y}_i)\}_{i=1}^N$, hyperparameter $\alpha$
Initialize $n_{w1} = n_w$ and $n_{wk} = 0$ for $k > 1$;
Initialize representations $\theta$;
for $i = 1$ to $N$ do
    Select word $w = x_i$ and its context $\vec{y}_i$;
    for $k = 1$ to $T$ do
        $\gamma_{ik} = \mathbb{E}_{q(\beta_w)}[\log p(z_i = k|\vec{\beta}, x_i)]$;
        for $j = 1$ to $C$ do
            $\gamma_{ik} \leftarrow \gamma_{ik} + \log p(y_{ij}|x_i, k, \theta)$;
        end
    end
end
$q(z_i = k) \leftarrow \exp(\gamma_{ik}) / \sum_\ell \exp(\gamma_{i\ell})$;
$\rho_t \leftarrow 0.025(1 - i/N)$, $\lambda_t \leftarrow 0.025(1 - i/N)$;
for $k = 1$ to $T$ do
    Update $n_{wk} \leftarrow (1 - \lambda_t)n_{wk} + \lambda_t n_w \gamma_{ik}$;
end
Update $\theta \leftarrow \theta + \rho_t \nabla_\theta \sum_k \sum_j \gamma_{ik} \log p(y_{ij}|x_i, k, \theta)$;
Return $\{q(\beta_{wk}) = \mathrm{Beta}(1 + n_{wk}, \alpha + \sum_{r=k+1}^T n_{wr})\}_{1 \leqslant w \leqslant V, 1 \leqslant k \leqslant T}, \theta$;
```

After training, learned vector representations can be used either on their own, as feature vectors, similarly to Skip-gram vectors, or for the word sense disambiguation task, which is choosing the word sense for a word depending on the context:

$$p(z = k|x, \mathbf{y}, \theta) \propto p(\mathbf{y}|x, z = k, \theta) \int p(z = k|x, \boldsymbol{\beta})q(\boldsymbol{\beta})d\boldsymbol{\beta}).$$

AdaGram has been trained on the standard for such studies corpus of «Wikipedia» articles up to April 2010 [20] and studied in terms of semantics of learned vectors and their use in downstream tasks. Figure 2.2 displays statistics on the number of learned senses for different values of α which controls *semantic definition* of the model. It can be seen that larger values of α generally lead to extracting more senses and that for more frequent words more senses is discovered.

To assess agreement between learned senses and conventional semantic inventories a number of standard «SemEval» test collections [21] were used, and, in addition, new collection «WWSI» has been automatically extracted based on the

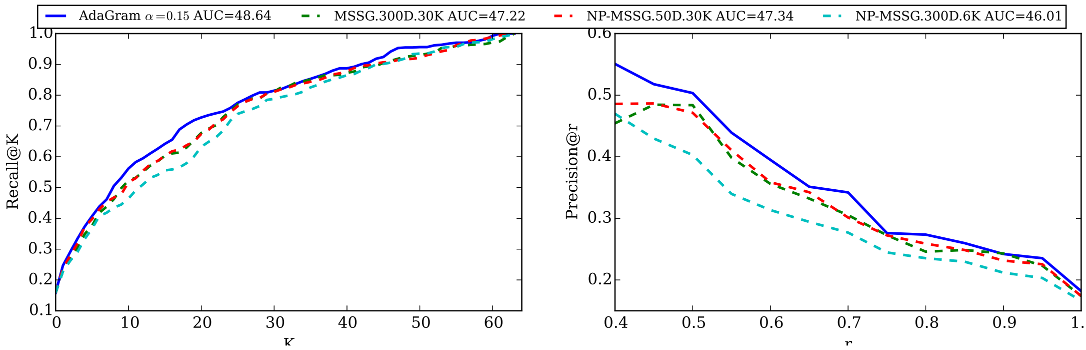


Figure 2.1 — Recall (left) and Precision (left) metrics for diversified information retrieval using different models on the SemEval 2013 task. K denotes position in search results, r – the target recall level using which precision is measured.

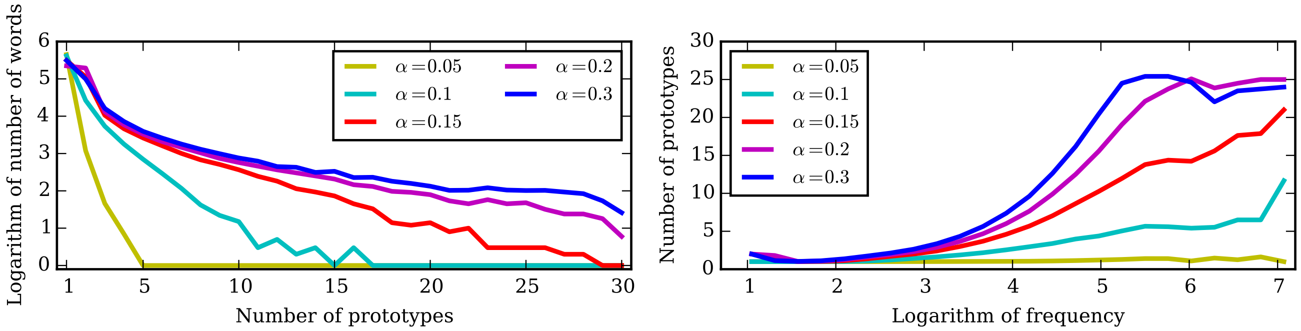


Figure 2.2 — Left: word distribution according to the number of senses extracted by AdaGram using different values of α (for number of senses k the displayed value is calculated as, where n_k is the number of words with k senses). Right: number of extracted senses depending on the word frequency.

«Wikipedia» articles and metadata. Each collection contains examples of use contexts of different words assigned to one of the senses. A word-sense induction system such as Skip-gram independently maps each of the contexts into senses it extracted after which the produced assignment is compared to ground truth used Adjust Rand index (ARI) [22].

AdaGram was compared to a number of analogous models for learning vector representations for word senses: Multi-sense Skip-gram (MSSG) and Nonparametric Multi-sense Skip-Gram [23] (NP-MSSG). These two models employ learning principles similar to the ones used in AdaGram as well as EM-like learning algorithms but either assume the number of senses fixed (in MSSG) or determine it heuristically (in NP-MSSG). In addition, we considered Multi-prototype Skip-Gram model

Table 1 — Word-sense induction task evaluation using adjusted Rand index (ARI) on different collections.

Model	**Dimensionality**	**SE-2007**	**SE-2010**	**SE-2013**	**WWSI**
MSSG	300	0.048	0.085	0.033	0.194
NP-MSSG	50	0.031	0.058	0.023	0.163
NP-MSSG	300	0.033	0.044	0.033	0.110
MPSG	300	0.044	0.077	0.014	0.160
AdaGram $\alpha = 0.15$	300	**0.069**	**0.097**	**0.061**	**0.286**

(MPSG) [24] as a simple baseline which only assumes uniform distribution over word senses. Results of the comparison are shown in Table 1.

Besides this, AdaGram along with the competing approaches has been evaluated in the search results diversification task for ambiguous queries on the SemEval 2013 [25] collection. In this task it is assumed that under high uncertainty about the implied sense of the query, search results are better to be shown not just ranked according to some lexical relevance, but also trying to cover all possible interpretations of the query. Therefore, a diversification system (in this case, a multi-sense vector representation model) has to cluster search results according to the senses it managed to discover. Quality of the system is then determined by the *recall* of the represented senses (as determined by the expert clustering) in the first K search results and the *precision* level necessary to achieve the corresponding recall value [25]. Figure 2.1 contains comparison of the studied models for single-word queries (because for them word vector representations can be used straightforwardly). As one can see, both recall and precision curves for AdaGram almost uniformly cover from the above curves for other models which demonstrates its superiority in this task.

Chapter 3. Deep generative model with fast learning capabilities

In the third chapter of the dissertation, a novel neural network-based generative model is proposed which is capable to incrementally learn from a small set of new observations. We assume that the true distribution of the observed objects $\mathbf{x} \in \mathcal{X}$ can be expressed as $p_d(\mathbf{x}) = \int p_d(\mathbf{x}|\gamma)p_d(\gamma)d\gamma$, i.e. as a mixture of conditional distributions depending on some parameter γ and that that dependency is smooth. Further, without overly restricting generality of the approach we will assume that the observation space is included in the set of D-dimensional binary vectors: $\mathcal{X} \subseteq \{0,1\}^{D}$[1].

At the training stage, the model has access to a sample from this distribution $p_d(\mathbf{x})$:

$$\mathbf{X}_{\text{train}} = \{X_m\}_{m=1}^{M}, \quad X_m = \{\mathbf{x}_{mr}\}_{r_m=1}^{R_m}, \quad \mathbf{x}_{mr} \sim p_d(\mathbf{x}|\gamma_m), \quad \gamma_m \sim p_d(\gamma).$$

At inference time, the model receives a similarly constructed incremental or conditioning training set $\mathbf{X}$:

$$\mathbf{X}' = \{X'_m\}_{m=1}^{M'}, \quad X'_m = \{\mathbf{x}'_{mr}\}_{r_m=1}^{R'_m}, \quad \mathbf{x}'_{mr} \sim p_d(\mathbf{x}|\gamma'_m), \quad \gamma'_m \sim p_d(\gamma),$$

which is substantially smaller than the training set $(M' \ll M, R'_m \leqslant R_u)$.

The goal of learning in such a model is to construct a distribution $p(\mathbf{x}|\mathbf{X}')$ as close as possible to the true $p_d(\mathbf{x}|\mathbf{X}') = \sum_{m=1}^{M'} p_d(\mathbf{x}|\gamma'_m)$. By fast learning or fast adaptation we mean that the model can incrementally take into account the new training set $\mathbf{X}'$ with the computational complexity only linearly dependant on the total number of the new objects $T = \sum_{m=1}^{M'} R'_m$.

The key quality of the «Generative Matching Networks» (GMN) proposed in this chapter is that it combines the flexibility of deep neural networks and high speed of incremental learning (in the sense defined above) which is in principle not achievable with gradient based learning methods for neural networks [26; 27]. Besides that, conceptually and empirically GMN works with *heterogeneous* incremental training sets (with $M' > 1$) better than previously proposed analogs [28—30].

GMN is built upon generative principles arising from variational autoencoders [31—33] which model data using a vector of latent variables $\mathbf{z} \in \mathbb{R}^L$. GMN expresses the

[1] Firstly, since computer-stored data is somehow discretized it can be represented as binary vectors. Secondly, it only requires to change the likelihood model in all further constructions in order to work with continuous observations

conditional distribution of interest as

$$p(\mathbf{x}|\mathbf{X};\theta) = \int p(\mathbf{z}|\mathbf{X};\theta)p(\mathbf{x}|\mathbf{z},\mathbf{X};\theta)d\mathbf{z},$$

where θ are model parameters.

Both the prior distribution $p(\mathbf{z}|\mathbf{X};\theta) = \mathcal{N}(z, \mu_{\text{prior}}(\mathbf{X};\theta), \Sigma_{\text{prior}}(\mathbf{X};\theta))$, and the decoder $p(\mathbf{x}|\mathbf{z},\mathbf{X};\theta) = \prod_{j=1}^{D} \text{Bernoulli}(x_j|\nu_j(\mathbf{z},\mathbf{X};\theta))$ are paremtrized using neural networks implementing functions μ_{prior}, Σ_{prior} and $\{\nu_j\}_{j=1}^{D}$.

The traditional method for training in similar models consists of direct maximization of the marginal likelihood of the whole available training data $\mathbf{X}_{\text{train}}$ However, in the considered setting this would be difficult since there is no guarantee that generalization to the new observations will occur. Instead, for training GMNs it is proposed to use the *meta-learning* paradigm [34—37]. In our work the paradigm has been adapted to the needs of generative modelling. Thus, from the available training set $\mathbf{X}_{\text{train}}$ we randomly generate subsets $\mathbf{X}' = \{X'_m\}_{m=1}^{M'}$ where objects $X'_m = \{\mathbf{x}'_{mr}\}_{r_m=1}^{R'_m}$ are chosen without return from a randomly chosen subset X_{u_m}, where $\mathbf{u}$ – is a set of random indices from 1 to M of size M'. The learning itself can be then expressed as maximization of the expected marginal likelihood of $\mathbf{X}'$:

$$\mathbb{E}_{\mathbf{X}'} \log p(\mathbf{X}';\theta) = \mathbb{E}_{\mathbf{X}'}\left[\sum_{t=1}^{T} \log p(\mathbf{x}'_t|\mathbf{X}'_{<t};\theta)\right] \to \max_{\theta}. \tag{3.1}$$

As one can see, the recurrent application of the chain rule for distributions allows us to connect the task of adaptive modelling of conditional distributions $p(\mathbf{x}|\mathbf{X};\theta)$ and the task of learning the whole incremental training set.

Since the direct optimization of the marginal likelihood $\log p(\mathbf{x}|\mathbf{X};)$ is infeasible, we apply variational inference for training GMNs:

$$\log p(\mathbf{x}|\mathbf{X};\theta) \geqslant \mathcal{L}(\theta,\varphi) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x},\mathbf{X};\varphi)}\big[\log p(\mathbf{z}|\mathbf{X};\theta) \\ + \log p(\mathbf{x}|\mathbf{z},\mathbf{X}';\theta) - \log q(\mathbf{z}|\mathbf{x},\mathbf{X};\varphi)\big]. \tag{3.2}$$

The key idea for implementing all model components that depend on the conditioning data $\mathbf{X}$ is in using the *attention* mechanism for choosing similar objects from $\mathbf{X}$ for generation (decoding) or recognition (encoding) [38; 39]. Since GMN does not have access to the true generative process $p(\mathbf{x}|\gamma)$, the model learns its own internal representation for objects such that the linear interpolation between representations of similar objects (generated by the same value of γ) adequately approximates the true

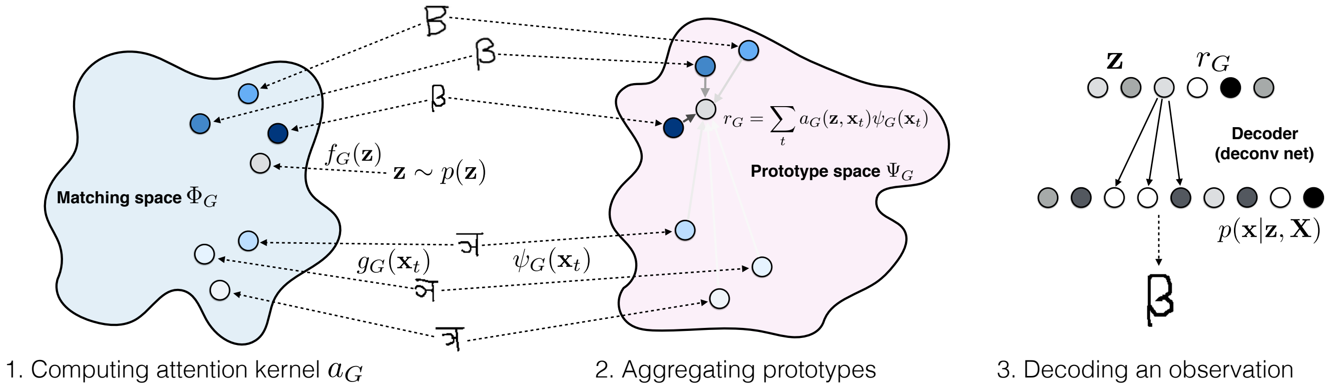

Matching space Φ_G
$f_G(\mathbf{z})$
$\mathbf{z} \sim p(\mathbf{z})$
$g_G(\mathbf{x}_t)$
$\psi_G(\mathbf{x}_t)$
$r_G = \sum_t a_G(\mathbf{z}, \mathbf{x}_t)\psi_G(\mathbf{x}_t)$
Prototype space Ψ_G
$\mathbf{z}$
r_G
Decoder (deconv net)
$p(\mathbf{x}|\mathbf{z}, \mathbf{X})$
1. Computing attention kernel a_G
2. Aggregating prototypes
3. Decoding an observation

Figure 3.1 — Illustration of the generative process modelled by GMNs.

distribution $p(\mathbf{x}|\gamma)$. Thus, our model is governed by the following equations:

$$\begin{aligned} a_G(\mathbf{z}, \mathbf{x}_t, \mathbf{h}_k) &= \frac{\exp(< f_G(\mathbf{z}, \mathbf{h}_k), g_G(\mathbf{x}_t, \mathbf{h}_k) >)}{\sum_{t'=0}^{T} \exp(< f_G(\mathbf{z}, \mathbf{h}_k), g_G(\mathbf{x}_{t'}, \mathbf{h}_k) >)}, \\ r_G^{k+1} &= \sum_{t=0}^{T} a_G(\mathbf{z}, \mathbf{x}_t, \mathbf{h}_k)\psi_G(\mathbf{x}_t, \mathbf{h}_k), \\ \mathbf{h}_{k+1} &= \mathrm{NN}_{\mathbf{h},G}(\mathbf{h}_k, r_G^k), \\ \nu_j &= \mathrm{NN}_{\nu_j,G}(\mathbf{z}, r_G^K), \end{aligned} \tag{3.3}$$

where functions f_G, g_G, ψ_G, $\mathrm{NN}_{\mathbf{h},G}$ and $\mathrm{NN}_{\nu_j,G}$ - are nonlinear neural networks parametrized by θ, and $< \cdot, \cdot >$ denote vector dot product Equations (3.3) define sequential application of the described attention mechanism over K steps where on each step objects $\mathbf{x}_t \in \mathbf{X}$ and latent variables $\mathbf{z}$ are projected in the shared *matching* space Φ_G after which the transformed matching results $a_G(\mathbf{x}, \mathbf{z}, \mathbf{h}_k)$ are used to interpolate in the *prototype* space Ψ_G. Schematically this process for $K = 1$ is shown on Figure 3.1.

Similarly the recognition model is defined $q(\mathbf{z}|\mathbf{x}, \mathbf{X}; \varphi) = \mathcal{N}(\mathbf{z}|\mu_R(\mathbf{x}, \mathbf{X}; \varphi), \Sigma_R(\mathbf{x}, \mathbf{X}$

$$\begin{aligned} a_R(\mathbf{x}, \mathbf{x}_t, \mathbf{h}_k) &= \frac{\exp(< f_R(\mathbf{x}, \mathbf{h}_k), g_R(\mathbf{x}_t, \mathbf{h}_k) >)}{\sum_{t'=0}^{T} \exp(< f_R(\mathbf{x}, \mathbf{h}_k), g_R(\mathbf{x}_{t'}, \mathbf{h}_k) >)}, \\ r_R^{k+1} &= \sum_{t=0}^{T} a_R(\mathbf{x}, \mathbf{x}_t, \mathbf{h}_k)\psi_R(\mathbf{x}_t, \mathbf{h}_k), \\ \mathbf{h}_{k+1} &= \mathrm{NN}_{\mathbf{h},R}(\mathbf{h}_k, r_R^k), \\ \mu_R(\mathbf{x}, \mathbf{X}) &= \mathrm{NN}_{\mu_R}(r_R^K), \\ \Sigma_R(\mathbf{x}, \mathbf{X}) &= \mathrm{NN}_{\Sigma_R}(r_R^K), \end{aligned} \tag{3.4}$$

where functions f_R and g_R, correspondingly, map the object $\mathbf{x}$ being recognized and objects $\mathbf{x}_t \in \mathbf{X}$ into the shared space Φ_R, with interpolation happening in the space Ψ_R.

Finally, the prior $p(\mathbf{z}|\mathbf{X};\theta)$ is modelled as:

$$\begin{aligned}
a_P(\mathbf{x}_t, \mathbf{h}_k) &= \frac{\exp(< f_P(\mathbf{h}_k), g_P(\mathbf{x}_t, \mathbf{h}_k) >)}{\sum_{t'=0}^{T} \exp(< f_P(\mathbf{h}_k), g_P(\mathbf{x}_{t'}, \mathbf{h}_k) >)}, \\
r_P^{k+1} &= \sum_{t=0}^{T} a_P(\mathbf{x}_t, \mathbf{h}_k)\psi_P(\mathbf{x}_t, \mathbf{h}_k), \\
\mathbf{h}_{k+1} &= \mathrm{NN}_{\mathbf{h},P}(\mathbf{h}_k, r_P^k), \\
\mu_P(\mathbf{X}) &= \mathrm{NN}_{\mu_P}(r_P^{K_P}), \\
\Sigma_P(\mathbf{X}) &= \mathrm{NN}_{\Sigma_P}(r_P^{K_P}).
\end{aligned} \tag{3.5}$$

One can note that equations (3.3), (3.4) and (3.5) involve matching with $\mathbf{x}_0$ (or a pseudo-input as we call it) which is necessary for the model to support empty conditioning sets. The pseudo-input is not modelled explicitly as an observation, only as supposed output values of neural networks that formally take it as in input with these values being trainable parameters.

The optimization problem (3.1) is solved via the stochastic training algorithm 3.

Algorithm 3: An iteration of the GMN training algorithm.

```
Data: Training set $\mathbf{X}$
Sample $\mathbf{X}'$ of size $T$ from $\mathbf{X}$;
for $t = 1$ to $T$ do
    Sample $\mathbf{z} \sim q(\mathbf{z}_t|\mathbf{x}'_t, \mathbf{X}'_{<t};\varphi)$;
    Estimate
      $\hat{\mathcal{L}}_t = \log p(\mathbf{z}|\mathbf{X}'_{<t};\theta) + \log p(\mathbf{x}'_t|\mathbf{z}, \mathbf{X}'_{<t}\theta) - \log q(\mathbf{z}|\mathbf{x}'_t, \mathbf{X}'_{<t};\varphi)$;
end
Update $\theta$ and $\varphi$ using gradients $\nabla_{\theta,\varphi} \sum_{t=1}^{T} \hat{\mathcal{L}}_t$;
```

GMN has been evaluated on the «Omniglot» dataset [40] consisting of 1623 classes of various hand-written characters, each of which only containing just 20 examples, which fully corresponds to the training regime assumed in this chapter. The model only had access to binarized 28 × 28 images of the characters. During training we used conditioning sets $\mathbf{X}'$ of 20 objects and $C_{\text{train}} = 2$ classes.

Firstly, GMN has been evaluated on the conditional density fitting task and compared with competing Neural Statistician [30] and One-shot VAE [29] models which were implemented using the exact same neural network architectures and differing only in the adaptation mechanisms. Results of this empirical comparison on the Omniglot test set are contained in Table 2.

Table 2 — Negative conditional log-likelihood on the Omniglot test set. C_{train} и C_{test} denote the maximal number of classes present in the conditioning set during training and inference correspondingly.

		Conditioning set size							
Model	C_{test}	0	1	2	3	4	5	10	19
GMN	1	**89.7**	**83.3**	**78.9**	**75.7**	**72.9**	**70.1**	**59.9**	**45.8**
GMN	2	**89.4**	**86.4**	**84.9**	**82.4**	**81.0**	**78.8**	**71.4**	**61.2**
GMN	3	**89.6**	**88.1**	**86.0**	**85.0**	**84.1**	**82.0**	**76.3**	**69.4**
GMN	4	**89.3**	**88.3**	**87.3**	**86.7**	**85.4**	**84.0**	**80.2**	**73.7**
VAE		89.1							
One-shot VAE	1		83.9						
Neural statistician, $C_{\text{train}} = 1$	1		102	83.4	77.8	75.2	74.6	71.7	71.5
Neural statistician, $C_{\text{train}} = 2$	2			86.4	**82.2**	82.3	80.6	79.7	79.0

Table 3 — Classification accuracy (%) for different number of classes and conditioning examples.

	5 classes		**20 classes**	
Model	**1-shot**	**5-shot**	**1-shot**	**5-shot**
GMN	82.7	**97.4**	64.3	**90.8**
One-shot VAE [29]	**90.2**	–	**76.3**	–
Neural statistician [30]	82.0	94.8	63.1	87.6
Matching networks [39]	98.1	98.9	93.8	98.5

As can be seen, GMN has a much better predictive performance for unseen characters when using only a handful available conditioning examples and the advantage is persistent even for larger conditioning sets. This is even more prominent for heterogeneous sets consisting of two and more character classes when attention-less models perform significantly worse.

An example of an incrementally generated sample from GMN is shown on Figure 3.2. One can see that as more conditioning objects are provided, GMN indeed is adapting its predictive distribution and generates more objects similar to the conditioning ones, importantly, not by simply copying them.

Finally, we applied GMN in the Omniglot classification task. Since for a well-trained model the predictive distribution $p(\mathbf{x}|\mathbf{X};\theta)$ would assign higher probability to objects similar to $\mathbf{X}$ than to dissimilar ones, this distribution can

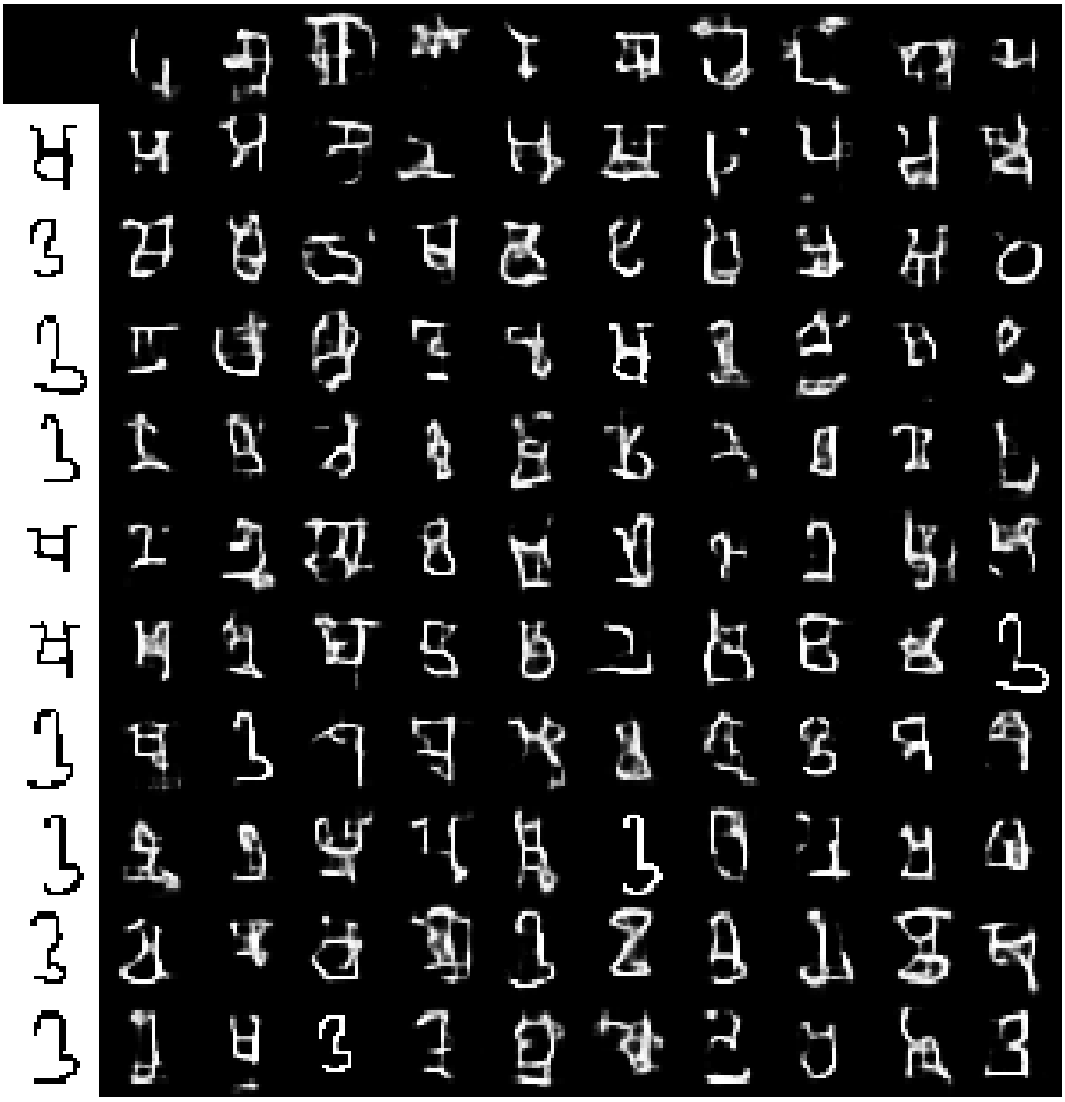


Figure 3.2 — GMN-generated samples. Left-most column contains the conditioning examples. Each row i (counting from the top to the bottom) contains independently generated samples from $p(\mathbf{x}|\mathbf{X}_{<i};\boldsymbol{\theta})$. The top-most row corresponds to an empty conditioning set $\mathbf{X} = \{\}$.

be used a voting function when building a classifier. Thus, when provided with training sets $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_C$ containing examples for each of the C classes, a GMN-based classifier chooses a class c about object $\mathbf{x}$ according to the following rule: $c = \arg\max \log p(\mathbf{x}|\mathbf{X}_c;\boldsymbol{\theta})$.

We report accuracy on the test set of Omniglot for this classifier, evaluated on 1000 random classification tasks in Table 3. GMN demonstrates competitive performance for the one-shot learning case and out-marches the competitive generative approaches, even getting close to the specialized Matching Networks classification model. By this, we demonstrate the usefulness of GMN for tasks that are not directly connected to density modelling but which can still be formulated through probabilistic inference.

Conclusion

In this work, the following results have been obtained:

1. A variational inference algorithm is developed for sequential distance-dependent Chinese Restaurant Process (seq-ddCRP) which connects the key statistics in the process and Laplacian of the modelled random graph. This algorithm allows to deterministically perform approximate probabilistic inference in seq-ddCRP based models and empirically shows faster convergence than Monte-Carlo methods while preserving high quality of the predictions. Besides, the obtained results are applicable to widely used models based on the Dirichlet process where they also lead to a novel algorithm capable of faster convergence and better variational lower bound than the standard one based on the truncated stick-breaking representation.
2. The novel nonparametric probabilistic AdaGram model is proposed for the problem of learning vector representations for multi-sense words. AdaGram relies on Dirichlet process for modelling the unknown number of senses for each of the words and is able to automatically determine it with a desired semantic resolution. Thanks to the developed stochastic variational EM-algorithm, AdaGram can be trained on large corpora taking advantage of parallel compute. Various quantitative studies of the model show the good quality of the learned word sense vectors comparing to competing approaches.
3. The problem of learning neural generative models with the incremental learning ability was studied. A nonparametric generative model called Generative Matching Network (GMN) has been proposed that is based on the variational auto encoders framework. GMN is capable of dynamically adapting its prior distribution, likelihood and amortized variational posterior as more data becomes available. Thanks to the attention mechanism, GMN handles heterogeneous incremental training sets better than simpler models which only maintain a single representation for the whole conditioning data. Experiments on Omniglot dataset has shown the rapid adaptation of GMN in terms of the predictive likelihood improvement on hold-out data after addition of each new observation, as well as applicability of GMN for problems different from pure generative modelling such as few-shot classification.

Bibliography

1. *Murphy*, *K. P.* Machine learning: a probabilistic perspective / K. P. Murphy. — MIT press, 2012.
2. An introduction to statistical learning. Vol. 112 / G. James [et al.]. — Springer, 2013.
3. *Vapnik*, *V.* The nature of statistical learning theory / V. Vapnik. — Springer science & business media, 2013.
4. *Murphy*, *K.* An introduction to graphical models / K. Murphy // Rap. tech. — 2001. — Vol. 96. — P. 1—19.
5. *Pearl*, *J.* [Bayesian analysis in expert systems]: comment: graphical models, causality and intervention / J. Pearl // Statistical Science. — 1993. — Vol. 8, no. 3. — P. 266—269.
6. *Idier*, *J.* Bayesian approach to inverse problems / J. Idier. — John Wiley & Sons, 2013.
7. *Orbanz*, *P.* Bayesian Nonparametric Models. / P. Orbanz, Y. W. Teh // Encyclopedia of machine learning. — 2010. — Vol. 1.
8. *Dagum*, *P.* Approximating probabilistic inference in Bayesian belief networks is NP-hard / P. Dagum, M. Luby // Artificial intelligence. — 1993. — Vol. 60, no. 1. — P. 141—153.
9. *Shimony*, *S. E.* Finding MAPs for belief networks is NP-hard / S. E. Shimony // Artificial Intelligence. — 1994. — Vol. 68, no. 2. — P. 399—410.
10. *Wainwright*, *M. J.* Graphical models, exponential families, and variational inference / M. J. Wainwright, M. I. Jordan. — Now Publishers Inc, 2008.
11. Markov chain Monte Carlo in practice: a roundtable discussion / R. E. Kass [et al.] // The American Statistician. — 1998. — Vol. 52, no. 2. — P. 93—100.

12. *Bloom*, *B. H.* Space/Time Trade-offs in Hash Coding with Allowable Errors / B. H. Bloom // Commun. ACM. — 1970. — T. 13, № 7. — C. 422—426. — URL: https://doi.org/10.1145/362686.362692.

13. *Blei*, *D. M.* Distance dependent Chinese restaurant processes. / D. M. Blei, P. I. Frazier // Journal of Machine Learning Research. — 2011. — Vol. 12, no. 8.

14. *Ferguson*, *T. S.* A Bayesian analysis of some nonparametric problems / T. S. Ferguson // The annals of statistics. — 1973. — P. 209—230.

15. *Rabiner*, *L.* An introduction to hidden Markov models / L. Rabiner, B. Juang // ieee assp magazine. — 1986. — Vol. 3, no. 1. — P. 4—16.

16. Variational inference for Dirichlet process mixtures / D. M. Blei, M. I. Jordan, [et al.] // Bayesian analysis. — 2006. — Vol. 1, no. 1. — P. 121—143.

17. *Mnih*, *A.* A scalable hierarchical distributed language model / A. Mnih, G. E. Hinton // Advances in neural information processing systems. — Citeseer. 2009. — P. 1081—1088.

18. *Sethuraman*, *J.* A constructive definition of Dirichlet priors / J. Sethuraman // Statistica sinica. — 1994. — P. 639—650.

19. Stochastic variational inference. / M. D. Hoffman [et al.] // Journal of Machine Learning Research. — 2013. — Vol. 14, no. 5.

20. *Shaoul*, *C.* The Westbury lab Wikipedia corpus / C. Shaoul, C. Westbury //. — 2010.

21. SemEval-2010 task 14: Word sense induction & disambiguation / S. Manandhar [и др.] // SemEval. — 2010. — С. 63—68.

22. *Hubert*, *L.* Comparing partitions / L. Hubert, P. Arabie // Journal of classification. — 1985. — Vol. 2, no. 1. — P. 193—218.

23. Efficient Non-parametric Estimation of Multiple Embeddings per Word in Vector Space / A. Neelakantan [и др.] // EMNLP. — 2014.

24. A Probabilistic Model for Learning Multi-Prototype Word Embeddings / F. Tian [и др.] // COLING. — 2014. — С. 151—160.

25. *Di Marco*, *A.* Clustering and diversifying web search results with graph-based word sense induction / A. Di Marco, R. Navigli // Computational Linguistics. — 2013. — Vol. 39, no. 3. — P. 709—754.

26. *McCloskey*, *M.* Catastrophic interference in connectionist networks: The sequential learning problem / M. McCloskey, N. J. Cohen // Psychology of learning and motivation. — 1989. — Vol. 24. — P. 109—165.

27. *Ratcliff*, *R.* Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. / R. Ratcliff // Psychological review. — 1990. — Vol. 97, no. 2. — P. 285.

28. *Salakhutdinov*, *R.* Learning with hierarchical-deep models / R. Salakhutdinov, J. B. Tenenbaum, A. Torralba // IEEE transactions on pattern analysis and machine intelligence. — 2013. — Vol. 35, no. 8. — P. 1958—1971.

29. One-Shot Generalization in Deep Generative Models / D. J. Rezende [et al.] // arXiv preprint arXiv:1603.05106. — 2016.

30. *Edwards*, *H.* Towards a Neural Statistician / H. Edwards, A. Storkey // arXiv preprint arXiv:1606.02185. — 2016.

31. *Kingma*, *D. P.* Auto-encoding variational bayes / D. P. Kingma, M. Welling // arXiv preprint arXiv:1312.6114. — 2013.

32. *Rezende*, *D. J.* Stochastic Backpropagation and Approximate Inference in Deep Generative Models / D. J. Rezende, S. Mohamed, D. Wierstra // Proceedings of the 31st International Conference on Machine Learning (ICML-14). — 2014. — P. 1278—1286.

33. *Mnih*, *A.* Neural variational inference and learning in belief networks / A. Mnih, K. Gregor // arXiv preprint arXiv:1402.0030. — 2014.

34. *Thrun*, *S.* Lifelong learning algorithms / S. Thrun // Learning to learn. — Springer, 1998. — P. 181—209.

35. *Vilalta*, *R.* A perspective view and survey of meta-learning / R. Vilalta, Y. Drissi // Artificial Intelligence Review. — 2002. — Vol. 18, no. 2. — P. 77—95.

36. *Hochreiter*, *S.* Learning to learn using gradient descent / S. Hochreiter, A. S. Younger, P. R. Conwell // International Conference on Artificial Neural Networks. — Springer. 2001. — P. 87—94.

37. One-shot Learning with Memory-Augmented Neural Networks / A. Santoro [et al.] // arXiv preprint arXiv:1605.06065. — 2016.

38. *Vinyals*, *O.* Pointer networks / O. Vinyals, M. Fortunato, N. Jaitly // arXiv preprint arXiv:1506.03134. — 2015.

39. Matching networks for one shot learning / O. Vinyals [et al.] // arXiv preprint arXiv:1606.04080. — 2016.

40. *Lake*, *B. M.* Human-level concept learning through probabilistic program induction / B. M. Lake, R. Salakhutdinov, J. B. Tenenbaum // Science. — 2015. — Vol. 350, no. 6266. — P. 1332—1338.