Indian Buffet Process Deep Generative Models

Abstract

Deep generative models (DGMs) have brought about a major breakthrough, as well as renewed interest, in generative latent variable models. However, an issue current DGM formulations do not address concerns the data-driven inference of the number of latent features needed to represent the observed data. Traditional linear formulations allow for addressing this issue by resorting to tools from the field of nonparametric statistics: Indeed, nonparametric linear latent variable models, obtained by appropriate imposition of Indian Buffet Process (IBP) priors, have been extensively studied by the machine learning community; inference for such models can been performed either via exact sampling or via approximate variational techniques. Based on this inspiration, in this paper we examine whether similar ideas from the field of Bayesian nonparametrics can be utilized in the context of modern DGMs in order to address the latent variable dimensionality inference problem. To this end, we propose a novel DGM formulation, based on the imposition of an IBP prior. We devise an efficient Black-Box Variational inference algorithm for our model, and exhibit its efficacy in a number of semi-supervised classification experiments. In all cases, we use popular benchmark datasets, and compare to state-of-the-art DGMs.

Keywords: Black-box variational inference; deep generative model; Indian Buffet Process; semi-supervised learning.

1. Introduction

Linear latent variable (LLV) models, including, among others, factor analysis (FA) and probabilistic principal component analysis (PPCA), have a long tradition in the field of generative modeling of high-dimensional observations with underlying latent structure. Their properties have been extensively studied, variants capable of capturing artifacts such as heavy tails and skewness have been developed in several works, while inference algorithms for such models have been derived using maximum-likelihood, variational inference (VI), as well as Markov chain Monte Carlo (MCMC) sampling (Lin et al., 2016; Montanari and Viroli, 2010). One of the difficulties related with the utilization of LLV models concerns the determination of the most appropriate number of latent variables (latent vector dimensionality) for representing a given dataset. Traditionally, this problem has been addressed by means of cross-validation; thus, multiple model configurations are trained, and each one is evaluated on the basis of some criterion that measures model fit to a separate validation dataset. However, such techniques are considerably wasteful both in terms of data exploitation (since a fraction of the data must be retained as validation set), as well as computational time (McLachlan and Peel, 2000).

To alleviate these issues, several researchers have considered utilization of concepts from the field of Bayesian nonparametrics. Nonparametric Bayesian models postulate a (theoretically) infinite-dimensional latent variable space. Appropriate priors are imposed over the postulated (infinite-dimensional) latent variables, that allow for deriving effective, data-driven posterior distributions over the latent dimension generation process. Specifically, nonparametric formulations of LLV models are most often obtained by imposition of an Indian Buffet Process (IBP) prior over the model latent variables (Chatzis, 2013; Gershman et al., 2015; Meeds et al., 2006; Doshi-Velez and Ghahramani, 2009). The IBP prior (Griffiths and Ghahramani, 2005) is a nonparametric prior for latent feature models in which observations are influenced by a combination of hidden features; it offers a principled prior in diverse contexts where the number of latent features is unknown. Eventually, the so-obtained hierarchical graphical model uses only a finite set of "effective" latent variables to represent the observed data points; this set is determined in a heuristics-free, data-driven way, as an integral part of the resulting model inference algorithm (Chatzis, 2012).

Despite these advances, the linear assumptions of LLV models cannot be considered realistic in most real-world data modeling scenarios. Traditionally, a solution towards the amelioration of these issues has been obtained by postulating mixtures of local LLV models, e.g. mixture of FA (MFA)-type models (Tipping and Bishop, 1999; Ghahramani and Hinton, 1997; Chatzis et al., 2008). This way, instead of attempting data modeling by means of a global linear model, a global nonlinear model is obtained by combining local linearities. Recently though, a much more potent solution has been obtained in the context of deep learning techniques (LeCun et al., 2015). Specifically, in the last couple of years, immense research interest has concentrated on the development of nonlinear latent variable models, where the inferred latent variable posteriors are parameterized via deep neural networks. This novel class of latent variable models is commonly referred to as deep generative models (DGMs) (Rezende et al., 2014; Kingma and Welling, 2014; Rezende and Mohamed, 2015; Gershman and Goodman, 2014; Burda et al., 2016). Inference for DGMs is performed by means of stochastic gradient variational Bayes (SGVB). This mainly consists in a smart reparameterization of the variational lower bound (Jaakkola and Jordan, 2000), which yields simple differentiable unbiased estimators, amenable to standard stochastic gradient ascent techniques (e.g., Adagrad (Duchi et al., 2010)).

Inspired from these advances, in this paper we address the problem of automatic data-driven inference of the latent variable dimensionality in DGMs. Specifically, we examine whether a nonparametric Bayesian formulation of DGMs, based on the utilization of the IBP prior, would offer an attractive solution to this problem. To this end, we devise a novel nonparametric hierarchical graphical formulation of DGMs, whereby the observed data are described via a factorized latent variable construction, driven by some latent indicators of data point allocation which are imposed an IBP prior. We derive an efficient inference algorithm for our model by resorting to Black-Box VI (BBVI) (Ranganath et al., 2014; Ruiz et al., 2016). We exhibit the efficacy of our approach in terms of semi-supervised classification performance, as well as in terms of the obtained effective model sizes, considering several popular benchmark datasets.

The remainder of this paper is organized as follows: In the next Section, we provide an overview of the theoretical foundation of our work: We first provide a brief overview of DGMs; subsequently, we concisely introduce the IBP prior and its utilization in FA-type models; finally, we review the inferential framework that will be used in the context of the proposed approach, namely BBVI. In the following Section, we introduce our proposed model, and derive its inference algorithm. Next, we perform the experimental evaluation of our approach. Finally, we conclude this paper, summarizing our contribution and discussing our results.

2. Theoretical Background

2.1. DGMs

In their basic formulation, DGMs assume that the observed random variables $\boldsymbol{x}$ are generated by some random process, involving an unobserved continuous random vector $\boldsymbol{z}$, with some prior distribution $p(\boldsymbol{z})$. The observed variables $\boldsymbol{x}$ are considered i.i.d. given the corresponding latent variables $\boldsymbol{z}$, with conditional likelihood function $p(\boldsymbol{x}|\boldsymbol{z};\boldsymbol{\theta})$. This way, the model's log-marginal likelihood can be lower-bounded as (evidence lower bound, ELBO):

$$\log p(X) \geq \mathcal{L}(\boldsymbol{\phi}) = \mathbb{E}_{q(\boldsymbol{z};\boldsymbol{\phi})}[\log p(\boldsymbol{x},\boldsymbol{z}) - \log q(\boldsymbol{z};\boldsymbol{\phi})] \tag{1}$$

where $\mathbb{E}_{q(\boldsymbol{z};\boldsymbol{\phi})}[\cdot]$ is the expectation of a function w.r.t. the random variable $\boldsymbol{z}$, drawn from $q(\boldsymbol{z};\boldsymbol{\phi})$, and $q(\boldsymbol{z};\boldsymbol{\phi})$ is the approximate (variational) posterior over the latent variable $\boldsymbol{z}$, that is inferred from the data.

DGMs assume that the adopted likelihood and prior distributions come from a parametric family, and that their probability density functions (pdf's) are differentiable almost everywhere w.r.t. their parameters and the (latent) variables $\boldsymbol{z}$. Specifically, DGMs assume that the likelihood function of the model, $\log p(\boldsymbol{x}|\boldsymbol{z};\boldsymbol{\theta})$, as well as the resulting latent variable posterior, $q(\boldsymbol{z};\boldsymbol{\phi})$, are parameterized via deep neural networks. For computational efficiency, $q(\boldsymbol{z};\boldsymbol{\phi})$ is typically taken as a diagonal Gaussian:

$$q(\boldsymbol{z};\boldsymbol{\phi}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}(\boldsymbol{x};\boldsymbol{\phi}), \operatorname{diag} \boldsymbol{\sigma}^2(\boldsymbol{x};\boldsymbol{\phi})) \tag{2}$$

where the $\boldsymbol{\mu}(\boldsymbol{x};\boldsymbol{\phi})$ and $\boldsymbol{\sigma}^2(\boldsymbol{x};\boldsymbol{\phi})$ are parameterized via deep neural networks, and $\operatorname{diag}\boldsymbol{\chi}$ is a diagonal matrix with $\boldsymbol{\chi}$ on its main diagonal.

Under these assumptions, DGMs yield a non-conjugate formulation, which does not allow to analytically derive the expression of $\mathbb{E}_{q(\boldsymbol{z};\boldsymbol{\phi})}[\log p(\boldsymbol{x}|\boldsymbol{z};\boldsymbol{\theta})]$, and its gradient. To resolve these issues in a computationally efficient way, DGMs resort to SGVB: This consists in drawing Monte Carlo samples from $q(\boldsymbol{z};\boldsymbol{\phi})$, which are further reparameterized as deterministic functions of the posterior mean $\boldsymbol{\mu}(\boldsymbol{x};\boldsymbol{\phi})$, variance $\boldsymbol{\sigma}^2(\boldsymbol{x};\boldsymbol{\phi})$, and some white random noise variable $\boldsymbol{\epsilon}$. This novel reparameterization introduced by SGVB ensures derivation of low variance estimators, under some mild conditions (Kingma and Welling, 2014). We have:

$$\mathcal{L}(\boldsymbol{\phi}) \approx \frac{1}{L}\sum_{l=1}^{L}[\log p(\boldsymbol{x},\boldsymbol{z}^{(l)}) - \log q(\boldsymbol{z}^{(l)};\boldsymbol{\phi})] \tag{3}$$

where

$$\boldsymbol{z}^{(l)} = \boldsymbol{\mu}(\boldsymbol{x};\boldsymbol{\phi}) + \boldsymbol{\sigma}(\boldsymbol{x};\boldsymbol{\phi}) \cdot \boldsymbol{\epsilon}^{(l)} \tag{4}$$

and

$$\boldsymbol{\epsilon}^{(l)} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) \tag{5}$$

2.2. Nonparametric Modeling Using the IBP Prior

In many unsupervised learning problems, it is necessary to derive a set of latent variables given a set of observations. A characteristic example is FA-type models: High-dimensional observations are usually associated with a lower-dimensional space of possible latent features that generate them. Identifying these sets of possible latent "properties," and determining which observed data point has each one of these "properties," may be extremely beneficial for the data modeling algorithm. Unfortunately, most traditional machine learning approaches require the number of latent features as an input. In such cases, usually one has to resort to application of a model selection technique to come up with a trade-off between model complexity and fit.

A solution to this problem is offered in the context of Bayesian nonparametrics. Nonparametric Bayesian approaches treat the number of latent features as a random quantity to be determined as part of the posterior inference procedure. The IBP is the most common nonparametric prior for latent feature models (Griffiths and Ghahramani, 2005). It is a prior on infinite binary matrices that allows us to simultaneously infer which features influence a set of observations and how many features there are. The form of the prior ensures that only a finite number of features will be present in any finite set of observations, but more features may appear as more observations are received.

Let us consider a set of N objects that may be assigned to a total of $K \to \infty$ features. Let $\boldsymbol{Z} = [z_{ik}]_{i,k=1}^{N,K}$ be a $N \times K$ matrix of assignment variables, with $z_{ik} = 1$ if the ith object is assigned to the kth feature (multiple z_{ik}'s may be equal to 1 for a given object i), $z_{ik} = 0$ otherwise. The IBP imposes a prior over $[\boldsymbol{Z}]$, a canonical form of $\boldsymbol{Z}$ that is invariant to the ordering of the features (Griffiths and Ghahramani, 2005). The imposed prior takes the form

$$\begin{aligned} p([\boldsymbol{Z}]) =& \frac{\alpha^K}{\prod_{h \in \{0,1\}^N \backslash \{0\}^N} K_h!} \exp\{-\alpha H_N\} \\ & \times \prod_{k=1}^{K} \frac{(N - m_k)!(m_k - 1)!}{N!} \end{aligned} \tag{6}$$

Here, m_k is the number of objects assigned to the kth feature (s.t. $z_{ik} = 1$), α is the innovation hyperparameter of the IBP prior which regulates the number of effective model features K, H_N is the Nth harmonic number, and K_h is the number of occurrences of the non-zero binary vector h among the columns in $\boldsymbol{Z}$. Each drawn latent dimension, $z_{\cdot k}$, is considered to be associated with some parameters, $\boldsymbol{\theta}_k$, that relate it with the observed random variables, $\boldsymbol{x}$, in the context of the postulated model likelihood function, $p(\boldsymbol{x}|\boldsymbol{z};\boldsymbol{\theta})$. These parameters are assumed to be drawn, in turn, from some base distribution, and are also inferred as part of the model training procedure.

Apart from MCMC (Griffiths and Ghahramani, 2005), inference for a latent feature model imposed an IBP prior can also be performed by means of VI. This is based on an alternative formulation of $p(\boldsymbol{Z})$ (Doshi-Velez et al., 2009b), which consists in the following equivalent hierarchical representation:

$$z_{ik} \sim \text{Bernoulli}(\pi_k)\, \forall i \tag{7}$$

$$\pi_k = \prod_{j=1}^{k} v_j \tag{8}$$

$$v_k \sim \mathrm{Beta}(\alpha, 1)\ \forall k \tag{9}$$

This equivalent hierarchical construction of the IBP prior results in a conjugate model formulation, which facilitates straightforward application of conventional VI, in a computationally elegant manner.

2.3. BBVI

Existing DGMs are a characteristic case of non-conjugate models the ELBO expression of which involves analytically intractable posterior expectations. As we discussed previously, DGMs handle this problem by resorting to stochastic optimization, where noisy gradients are formed with Monte Carlo approximation. However, such an approach must address a core problem with Monte Carlo estimates of the gradient, namely their prohibitively high variance. The solution devised to address this issue, namely SGVB, reduces the variance by means of the reparameterization trick (4). However, this solution is only amenable to models with continuous latent variables.

BBVI is an alternative to SGVB, amenable to non-conjugate probabilistic models that entail both discrete and continuous latent variables. Let us consider a probabilistic model $p(\boldsymbol{x}, \boldsymbol{z})$ and a sought variational family $q(\boldsymbol{z}; \boldsymbol{\phi})$. BBVI optimizes the ELBO (1) by relying on the "log-derivative trick" (Glynn, 1990; Williams, 1992) to obtain Monte Carlo estimates of the gradient. Specifically, by application of the identities

$$\nabla_{\boldsymbol{\phi}} q(\boldsymbol{z}; \boldsymbol{\phi}) = q(\boldsymbol{z}; \boldsymbol{\phi}) \nabla_{\boldsymbol{\phi}} \log q(\boldsymbol{z}; \boldsymbol{\phi}) \tag{10}$$

$$\mathbb{E}_{q(\boldsymbol{z};\boldsymbol{\phi})}[\nabla_{\boldsymbol{\phi}} \log q(\boldsymbol{z}; \boldsymbol{\phi})] = 0 \tag{11}$$

the gradient of the ELBO (1) reads

$$\nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\phi}) = \mathbb{E}_{q(\boldsymbol{z};\boldsymbol{\phi})}[f(\boldsymbol{z})] \tag{12}$$

where

$$f(\boldsymbol{z}) = \nabla_{\boldsymbol{\phi}} \log q(\boldsymbol{z}; \boldsymbol{\phi}) \left[\log p(\boldsymbol{x}, \boldsymbol{z}) - \log q(\boldsymbol{z}; \boldsymbol{\phi})\right] \tag{13}$$

The so-obtained Monte Carlo estimator, based on computing the posterior expectations $\mathbb{E}_{q(\boldsymbol{z};\boldsymbol{\phi})}[\cdot]$ via sampling from $q(\boldsymbol{z}; \boldsymbol{\phi})$, only requires evaluating the log-joint distribution $\log p(\boldsymbol{x}, \boldsymbol{z})$, the log-variational distribution $\log q(\boldsymbol{z}; \boldsymbol{\phi})$, and the score function $\nabla_{\boldsymbol{\phi}} \log q(\boldsymbol{z}; \boldsymbol{\phi})$, which is easy for a large class of models. However, the resulting estimator may have high variance, especially if the variational approximation $q(\boldsymbol{z}; \boldsymbol{\phi})$ is a poor fit to the actual posterior. In order to reduce the variance of the estimator, one common strategy in BBVI consists in the use of *control variates.*

A control variate is a random variable that is included in the estimator, preserving its expectation but reducing its variance. The most usual choice for control variates, which we adopt in this work, is the so-called weighted score function: Under this selection, the ELBO gradient becomes

$$\nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\phi}) = \sum_{n=1}^{N} \mathbb{E}_{q(\boldsymbol{z};\boldsymbol{\phi})}[f_n(\boldsymbol{z}) - a_n h_n(\boldsymbol{z})] \tag{14}$$

where $f_n(\cdot)$ and $h_n(\cdot)$ are the nth component of $f(\cdot)$ and $h(\cdot)$, respectively, we denote

$$h_n(\boldsymbol{z}) = \nabla_{\boldsymbol{\phi}} \log q(\boldsymbol{z}_n; \boldsymbol{\phi}) \tag{15}$$

and the constants a_n are given by (Ranganath et al., 2014)

$$a_n = \frac{\mathrm{Cov}\left(f_n(\boldsymbol{z}), h_n(\boldsymbol{z})\right)}{\mathrm{Var}\left(h_n(\boldsymbol{z})\right)} \tag{16}$$

On this basis, derivation of the sought variational posteriors is performed by utilizing the gradient expression (14) in the context of popular, off-the-shelf optimization algorithms, e.g. AdaM (Kingma and Ba, 2015) and Adagrad (Duchi et al., 2010).

Finally, it has been very recently shown in (Ruiz et al., 2016) that the variance of the BBVI estimator can be further reduced by replacing the conventional MC sampling procedure with importance sampling. Specifically, (Ruiz et al., 2016) suggest that, instead of sampling from the variational posterior $q(\boldsymbol{z}; \boldsymbol{\phi})$ to estimate the expectations $\mathbb{E}_{q(\boldsymbol{z};\boldsymbol{\phi})}[\cdot]$, we take samples from an *overdispersed distribution* (Jørgensen, 1987) *in the same exponential family as* $q(\boldsymbol{z}; \boldsymbol{\phi})$; the resulting terms are averaged on the basis of an importance weighting scheme. All these ideas are exploited in the context of the inference algorithm of our model, outlined in the following Section.

3. Proposed Approach

As discussed in the Introduction, the ultimate goal of this work is to enable automatic, data-driven inference of the latent vector dimensionality of DGMs. To this end, we examine the efficacy of a nonparametric DGM formulation, based on the utilization of the IBP prior; this is in direct analogy to previous successful formulations of nonparametric LLV models, e.g. (Chatzis, 2012).

Specifically, let us consider a modeled dataset $X = \{\boldsymbol{x}_i\}_{i=1}^N$. The proposed IBP-DGM assumes a conditional likelihood $p(\boldsymbol{x}_i|\boldsymbol{z}_i; \boldsymbol{\theta})$, parameterized by deep neural networks, and selected similar to the case of conventional DGMs; for instance, in case of real observations, $\boldsymbol{x}_i \in \mathbb{R}^D$, a diagonal Gaussian likelihood is selected; in cases of binary observations, $\boldsymbol{x}_i \in \{0, 1\}^D$, we opt for a Bernoulli likelihood. Further, we introduce the following hierarchical prior formulation for the latent variables $\boldsymbol{z}_i$:

$$\boldsymbol{z}_i = \tilde{\boldsymbol{z}}_i \cdot \hat{\boldsymbol{z}}_i \tag{17}$$

$$p(\tilde{\boldsymbol{z}}_i) = \mathcal{N}(\tilde{\boldsymbol{z}}_i|\boldsymbol{0}, \boldsymbol{I}) \tag{18}$$

$$p(\hat{\boldsymbol{z}}_i) = \prod_{k=1}^{K\to\infty} \mathrm{Bernoulli}(\hat{z}_{ik}|\pi_k) \tag{19}$$

$$\pi_k \triangleq \prod_{j=1}^{k} v_j, \ k \in \{1, \dots, \infty\} \tag{20}$$

$$p(v_k) = \mathrm{Beta}(v_k|\alpha, 1), \ k \in \{1, \dots, \infty\} \tag{21}$$

The introduction of the binary latent variables $\hat{\boldsymbol{z}}_i$ in Eq. (17) essentially allows for the model to infer which latent features $\tilde{z}_{ik}$, $k \in \{1, \dots, K \to \infty\}$, are active for each one of the observed data $\boldsymbol{x}_i$. This way, if a latent feature, say the kth, yields drawn samples of the indicators $\hat{z}_{ik}$ that are equal to zero for every observation, $\boldsymbol{x}_i$, it will be effectively ignored by the model. This mechanism induces sparsity, essentially reducing the number of "effective" latent variables to a finite and possibly limited set of inferred latent features.

However, under the infinite dimensional setting prescribed in Eqs. (17)-(21), Bayesian inference is not feasible. For this reason, we employ a common strategy in the literature of Bayesian nonparametrics, formulated on the basis of a truncated, implicitly finite, representation of the IBP (Teh et al., 2007; Doshi-Velez et al., 2009a). That is, we fix a value $K \ll \infty$, letting the posterior over the v_k have the property $q(v_K = 0) = 1$. In other words, we set the π_k equal to zero for $k > K \; \forall i$. Note that, under this setting, our model continues to employ a full IBP prior: truncation is not imposed on the model itself, but only on the derived posterior distribution to allow for a tractable inference procedure (Doshi-Velez et al., 2009a).

Under this truncated variational framework, we conclude the formulation of IBP-DGM by postulating that the sought variational posteriors take on the following forms:

$$q(\tilde{\boldsymbol{z}}_i; \boldsymbol{\phi}) = \mathcal{N}(\tilde{\boldsymbol{z}}_i | \boldsymbol{\mu}(\boldsymbol{x}_i; \boldsymbol{\phi}), \mathrm{diag}\, \boldsymbol{\sigma}^2(\boldsymbol{x}_i; \boldsymbol{\phi})) \tag{22}$$

$$q(\hat{\boldsymbol{z}}_i; \boldsymbol{\phi}) = \prod_{k=1}^{K} \mathrm{Bernoulli}(\hat{z}_{ik} | \hat{\pi}_k(\boldsymbol{x}_i; \boldsymbol{\phi})) \tag{23}$$

$$q(v_k; \boldsymbol{\phi}) = \mathrm{Beta}(v_k | a_k(\boldsymbol{x}_i; \boldsymbol{\phi}), b_k(\boldsymbol{x}_i; \boldsymbol{\phi})), \; k \in \{1, \dots, K\} \tag{24}$$

Note that, in Eqs. (22)-(24), the $\boldsymbol{\mu}(\boldsymbol{x}_i; \boldsymbol{\phi})$, $\boldsymbol{\sigma}^2(\boldsymbol{x}_i; \boldsymbol{\phi})$, $\hat{\pi}_k(\boldsymbol{x}_i; \boldsymbol{\phi})$, $a_k(\boldsymbol{x}_i; \boldsymbol{\phi})$, and $b_k(\boldsymbol{x}_i; \boldsymbol{\phi})$ are parameterized by deep neural networks. Specifically, we postulate a single deep network, with separate outputs for each one of the parameterized functions.

Finally, regarding the parameters $\boldsymbol{\theta}$ of the postulated model likelihood, $p(\boldsymbol{x}|\boldsymbol{z}; \boldsymbol{\theta})$, we impose a simple spherical prior of the form:

$$p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} | \boldsymbol{0}, \boldsymbol{I}) \tag{25}$$

In addition, to facilitate computational tractability, we consider that the sought variational posterior $q(\boldsymbol{\theta})$ collapses to a single point, $\hat{\boldsymbol{\theta}}$, that essentially constitutes a point-estimate; in other words, we assume

$$q(\boldsymbol{\theta}) = \delta_{\hat{\boldsymbol{\theta}}}(\boldsymbol{\theta}) \tag{26}$$

where $\delta_{\hat{\boldsymbol{\theta}}}(\boldsymbol{\theta})$ is a distribution over $\boldsymbol{\theta}$ with all its mass concentrated on $\hat{\boldsymbol{\theta}}$.

This concludes the formulation of the proposed IBP-DGM. Inference for the proposed model is performed by resorting to BBVI, which was described in the previous Section. Specifically, the expression of the ELBO of the model becomes

$$\begin{aligned}\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}) =& \mathbb{E}\Big[\log p(\boldsymbol{x} | \tilde{\boldsymbol{z}} \cdot \hat{\boldsymbol{z}}; \boldsymbol{\theta}) + \log p(\tilde{\boldsymbol{z}}) - \log q(\tilde{\boldsymbol{z}}; \boldsymbol{\phi}) \\ &+ \log p(\hat{\boldsymbol{z}}) - \log q(\hat{\boldsymbol{z}}; \boldsymbol{\phi}) + \log p(\boldsymbol{\theta}) - \log q(\boldsymbol{\theta}) \\ &+ \sum_{k=1}^{K} \left\{ \log p(v_k) - \log q(v_k; \boldsymbol{\phi}) \right\} \Big]\end{aligned} \tag{27}$$

and is amenable to BBVI with the control variates selected as described in the previous Section. In this context, to effect the entailed approximate ELBO optimization procedure, we resort to the AdaM optimization algorithm (Kingma and Ba, 2015); we use a learning rate of 3×10^{-4}, and an exponential decay rate for the first and second moment at 0.9 and 0.999, respectively.

3.1. Semi-supervised Learning Using IBP-DGMs

Recently, it has been shown that DGMs are extremely potent in the context of semi-supervised learning tasks (Kingma et al., 2014). On this basis, in this work we evaluate the efficacy of IBP-DGM under such a semi-supervised learning setting. Indeed, when it comes to semi-supervised learning, IBP-DGM retains its formulation discussed previously, except for the introduction of a prior $p(y)$ over the labels y of the observed data points, as well as a corresponding variational posterior $q(y; \boldsymbol{\phi})$. Specifically, we assume that

$$p(y = c) = \frac{1}{C}, \ \forall c \tag{28}$$

and

$$q(y; \boldsymbol{\phi}) = \mathrm{Cat}(y|\varpi(\boldsymbol{x})) \tag{29}$$

where $\varpi(\boldsymbol{x})$ is parameterized via a deep network, and C is the total number of possible classes. In addition, the likelihood function of the model is also modified to take into account (possible) label information; we assume a likelihood function of the form $p(\boldsymbol{x}|\tilde{\boldsymbol{z}} \cdot \hat{\boldsymbol{z}}, y; \boldsymbol{\theta})$. This way, the model ELBO eventually yields the expression

$$\begin{aligned} \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}) = \mathbb{E}\big[&\log p(\boldsymbol{x}|\tilde{\boldsymbol{z}} \cdot \hat{\boldsymbol{z}}, y; \boldsymbol{\theta}) + \log p(y) - \log q(y; \boldsymbol{\phi}) \\ &+ \log p(\hat{\boldsymbol{z}}) - \log q(\hat{\boldsymbol{z}}; \boldsymbol{\phi}) + \log p(\tilde{\boldsymbol{z}}) - \log q(\tilde{\boldsymbol{z}}; \boldsymbol{\phi}) \\ &+ \log p(\boldsymbol{\theta}) - \log q(\boldsymbol{\theta}) + \sum_{k=1}^{K} \big\{ \log p(v_k) - \log q(v_k; \boldsymbol{\phi}) \big\} \big] \end{aligned} \tag{30}$$

which is amenable to BBVI. The latter can be effected via the AdaM algorithm, as discussed previously.

4. Experiments

To exhibit the efficacy of our approach, we perform evaluation in a series of semi-supervised classification tasks. To this end, we use several benchmark datasets, namely MNIST, Rotated MNIST, MNIST+Background Images, MNIST+Random Background, Rotated MNIST+Background Images, and (Small-)NORB. In all cases, we perform model evaluation considering two alternative architectures of the deep networks parameterizing the postulated likelihood and posterior distributions. The first alternative comprises simple Dense Layer (DL) architectures. The second one is based on the Memory Network (MN) architecture recently proposed in (Li et al., 2016).

The MN architecture employs an external hierarchical memory to capture variant information at different abstraction levels trained in an unsupervised manner. Hence, it is appropriate for parameterizing generative latent variable models, such as DGMs. This kind of additional memory mechanism can help to reduce the competition between invariant feature extraction and local variant reconstruction in the context of both bottom-up inference and top-down generation; this is especially true when label information is provided (e.g., in semi-supervised learning). This way, it allows for developing DGMs with a possibly large external memory, and an attention mechanism that puts more emphasis on a select subset of the inferred latent variables, and less to the rest.

Specifically, each (deterministic) MN layer first performs computation of some low-level generative information $\boldsymbol{h}_g$ given the layer input $\boldsymbol{h}_{in}$

$$\boldsymbol{h}_g = f_g(\boldsymbol{h}_{in}; \boldsymbol{W}_g, \boldsymbol{b}_g) \tag{31}$$

where f_g is a proper nonlinear transformation, and $\boldsymbol{W}_g, \boldsymbol{b}_g$ are the weights and biases of the transformation. Subsequently, the MN layer accesses its memory mechanism to retrieve additional information that is not incorporated in the low-level information encoded by $\boldsymbol{h}_g$. To this end, the MN layer needs to compute first how to properly access its memory. This is effected via an attention mechanism, parameterized by a controlling matrix $\boldsymbol{A}$ and a bias vector $\boldsymbol{b}_a$; the corresponding control information, $\boldsymbol{h}_a$, reads

$$\boldsymbol{h}_a = f_a(\boldsymbol{h}_g; \boldsymbol{A}, \boldsymbol{b}_a) \tag{32}$$

where f_a is a proper nonlinear transformation. The so-obtained control vector, $\boldsymbol{h}_a$, is used to eventually retrieve appropriate information from the memory of the MN layer; this is parameterized as

$$\boldsymbol{h}_m = f_m(\boldsymbol{h}_a; \boldsymbol{M}) \tag{33}$$

where $\boldsymbol{M}$ is the matrix of stored memories of the considered MN layer, and f_m is a proper nonlinear transformation. The low-level generative information $\boldsymbol{h}_g$ is eventually combined with the memory information, $\boldsymbol{h}_m$, retrieved via the established attention mechanism, to obtain the final output of the MN layer. We have (Li et al., 2016)

$$\boldsymbol{h}_{out} = f_{out}(\boldsymbol{a}_{out} + \boldsymbol{b}_{out} \cdot \boldsymbol{c}_{out}) \tag{34}$$

where

$$\boldsymbol{a}_{out} = \boldsymbol{a}_1 + \boldsymbol{a}_2 \cdot \boldsymbol{h}_g + \boldsymbol{a}_3 \cdot \boldsymbol{h}_m + \boldsymbol{a}_4 \cdot \boldsymbol{h}_g \cdot \boldsymbol{h}_m \tag{35}$$

$$\boldsymbol{c}_{out} = \sigma(\boldsymbol{c}_1 + \boldsymbol{c}_2 \cdot \boldsymbol{h}_g + \boldsymbol{c}_3 \cdot \boldsymbol{h}_m + \boldsymbol{c}_4 \cdot \boldsymbol{h}_g \cdot \boldsymbol{h}_m) \tag{36}$$

$\sigma(\cdot)$ is the logistic sigmoid function, and the $\boldsymbol{b}_{out}$ and $\{\boldsymbol{a}_j, \boldsymbol{c}_j\}_{j=1}^4$ parameterize f_{out}.

In all our experiments, for simplicity and computational tractability, we use architectures comprising only one hidden layer (DL or MN), with 500 (deterministic) units. We use ReLU nonlinearities for all the postulated (deterministic) hidden units (Nair and Hinton, 2010). The used MN layers comprise 100 memory slots; that is the number of rows of matrix $\boldsymbol{A}$ in Eq. (32), or, conversely, the number of columns of the memory matrix $\boldsymbol{M}$ (Li et al., 2016). In all cases, the maximum size of the postulated latent vectors $\boldsymbol{z}$ (truncation threshold K of the variational posterior) is set to 50. Our source codes have been developed in Python, and make use of the Tensorflow[1] and Edward[2] (Tran et al., 2016) libraries. For the purpose of experimental comparisons, we have also utilized source codes from (Kingma et al., 2014)[3].

[1] https://www.tensorflow.org

[2] http://edwardlib.org

[3] https://github.com/dpkingma/nips14-ssl

4.1. Semi-supervised Classification Performance

Here, we discuss the performance of our approach in the task of semi-supervised classification. To provide some comparative results, apart from our method we also evaluate the M2 approach proposed in (Kingma et al., 2014), which constitutes the parametric equivalent of IBP-DGM in the context of semi-supervised learning. We perform evaluations under an experimental setup where 1% of the available training data is presented to the trained models as *labeled* training examples (randomly selected, in equal proportions from each class), while the rest is used as *unlabeled* training examples.

For the MNIST dataset, we combine the training set of $50,000$ examples with the validation set of $10,000$ examples into our available training dataset. Before each epoch, the normalized MNIST images are binarized by sampling Bernoulli distributions, similar to (Uria et al., 2014). For the considered MNIST variants (i.e., MNIST+Background Images, MNIST+Random Background, Rotated MNIST+Background Images, and Rotated MNIST), we retain the original split into $12,000$ training examples and $50,000$ test examples. Finally, regarding NORB, this set comprises $24,300$ training samples and an equal amount of test samples distributed across 5 classes (animal, human, plane, truck, car). We normalize all NORB images, following the procedure suggested in Miyato et al. (2016), using image pairs of 32×32; this results in $2,048$-dimensional inputs. We add uniform noise between 0 and 1 to each pixel value, to allow for effectively modeling them by means of Gaussian conditional likelihoods $p(\boldsymbol{x}|\tilde{\boldsymbol{z}} \cdot \hat{\boldsymbol{z}}, y; \boldsymbol{\theta})$. We normalize the NORB dataset by 256.

In Tables 1-2, we provide the obtained performance results (error rates %) of the evaluated methods under the two considered experimental scenarios. These figures are average performance results over 50 repetitions of our experiments, with different random training data splits into labeled and unlabeled subsets each time. As we observe, our approach yields a clear improvement over the competition in all cases. To examine the statistical significance of the observed performance differences, we run the Student's-t statistical significance test on the pairs of performances of our method and M2. The test rejected the null hypothesis, with p-values below 10^{-8}, in all cases.

Further, an interesting observation is that the obtained improvement of IBP-DGM over M2 is more profound in the case of the DL parameterization. We suspect this result is due to the fact that the MN parameterization introduces an attention mechanism which essentially puts more emphasis on some latent characteristics of the data, while putting less emphasis on some others. This might turn out to be more beneficial for some parametric model than for a nonparametric one, which already includes a (different sort of) mechanism for latent feature selection (retention or omission).

In the same vein, another important finding is that IBP-DGM with DL parameterization outperforms M2 with MN parameterization. This finding shows that the nonparametric Bayesian latent feature selection/reinforcement mechanisms of IBP-DGM are more potent compared to attention-driven mechanisms based on conventional neural network architectures (such as the MN architecture), which currently constitute the state-of-the-art in the literature of DGMs.

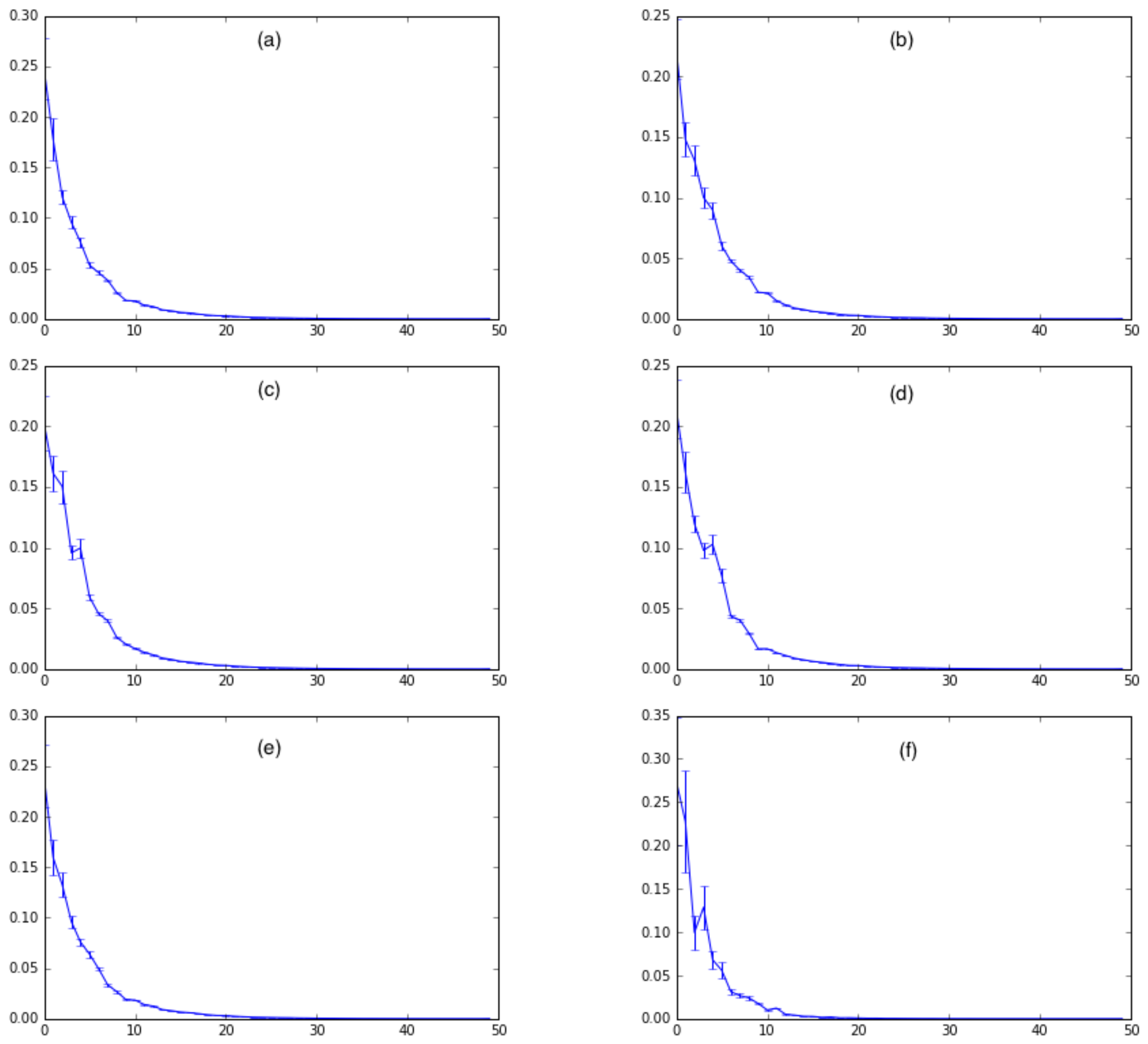

Figure 1: Posterior probabilities $q(\hat{z}_{\cdot k}; \boldsymbol{\phi})$, for each postulated latent variable $k \in \{0, \ldots, 49\}$, under the DL parameterization scheme: (a) MNIST; (b) Rotated MNIST; (c) MNIST+Background Images; (d) MNIST+Random Background; (e) Rotated MNIST+Background Images; (f) NORB.

Table 1: Semi-supervised test error (%) using the considered DL parameterization.

Method	M2	IBP-DGM
MNIST	8.10	7.85
Rotated MNIST	38.80	32.82
MNIST+Background Images	16.16	8.99
MNIST+Random Background	12.34	7.78
Rotated MNIST+Background Images	12.69	8.03
NORB	18.02	15.14

Table 2: Semi-supervised test error (%) using the considered MN parameterization.

Method	M2	IBP-DGM
MNIST	8.04	7.45
Rotated MNIST	37.29	32.80
MNIST+Background Images	9.08	7.94
MNIST+Random Background	7.87	6.85
Rotated MNIST+Background Images	8.42	7.95
NORB	15.57	14.88

4.2. Inferred Effective Latent Variable Dimensionality

In this Section, we attempt to get deeper insights into the function and the outcomes of the IBP-induced mechanisms of our model that perform effective model size inference. To this end, we examine the values of the posteriors over the latent indicators, $q(\hat{z}_{\cdot k}; \boldsymbol{\phi})$, obtained in each one of the previously considered experimental scenarios. In Figs. 1 and 2, we depict the means and error bars (over the used training data points) of these values, obtained at the end of IBP-DGM model training. As we observe, our model tends to yield high enough posterior values only for the first 10-12 latent components. In the case of MNIST and its considered variants, an additional 8-10 latent components turn out to yield non-negligible posteriors, thus allowing for being considered as “modestly” active. Hence, in all cases we observe that our model effectively retains only a fraction of the originally postulated latent variables.

Another very characteristic finding is that, in most cases, the posterior values, $q(\hat{z}_{\cdot k}; \boldsymbol{\phi})$, of the active components tend to yield higher mean values, and most importantly, higher standard deviations, in the case of the DL parameterization. In our view, this outcome vouches for our previous claims that the attention mechanisms of the MN network are actually complementary to the nonparametric feature omission/retention mechanisms of the IBP prior: Both put more emphasis on some postulated latent variables, while putting less emphasis on some others. The IBP-induced mechanism seems more potent *per se*; however, its concurrent utilization with the MN mechanism could be beneficial for the overall model. When the MN mechanism is missing, the IBP-induced mechanism is the only one that is responsible for appropriately handling the procedure of putting more or less emphasis on some latent variables for each data point; hence the higher standard deviation of the

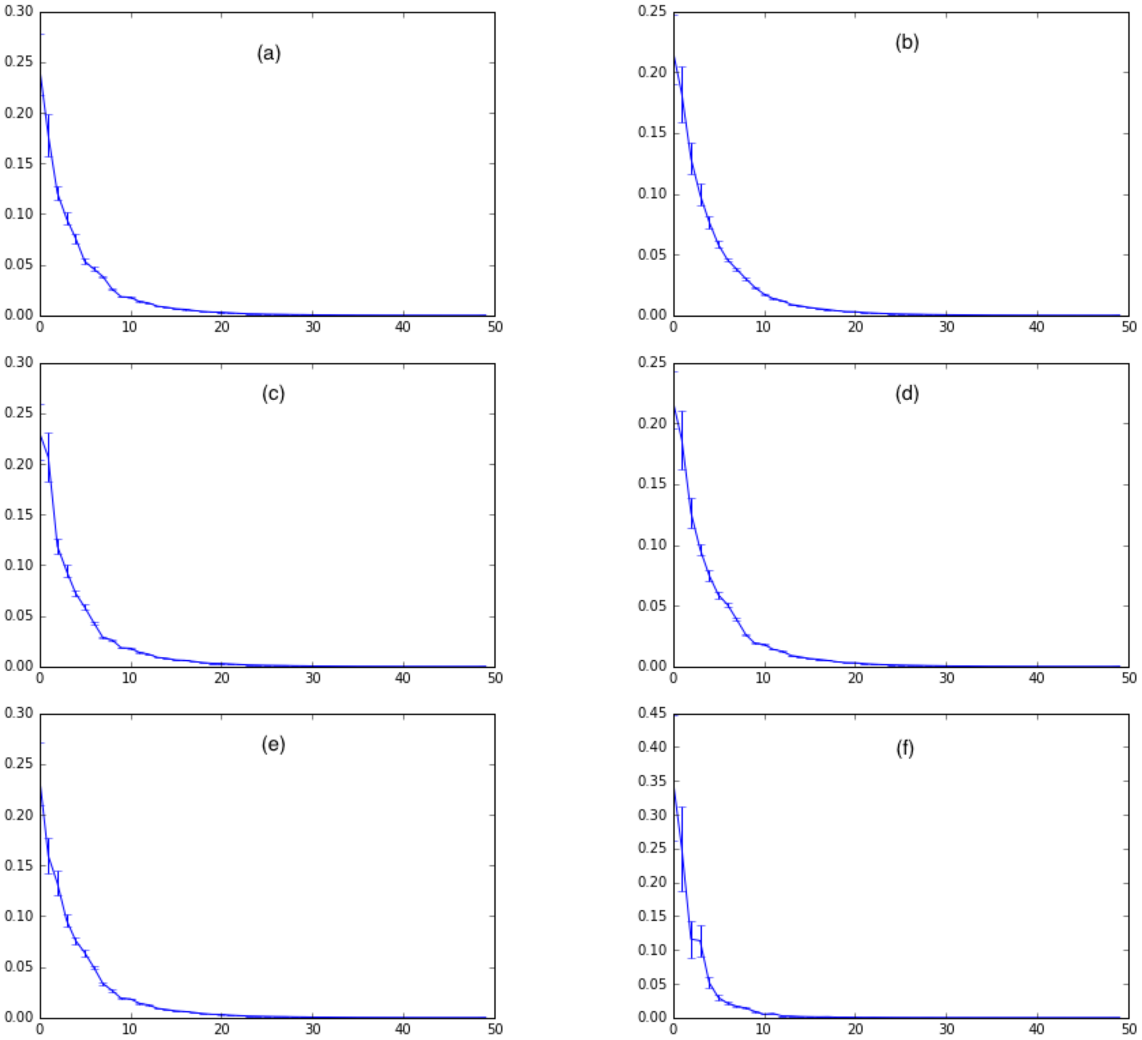

Figure 2: Posterior probabilities $q(\hat{z}_{\cdot k}; \boldsymbol{\phi})$, for each postulated latent variable $k \in \{0, \ldots, 49\}$, under the MN parameterization scheme: (a) MNIST; (b) Rotated MNIST; (c) MNIST+Background Images; (d) MNIST+Random Background; (e) Rotated MNIST+Background Images; (f) NORB.

$q(\hat{z}_{\cdot k}; \boldsymbol{\phi})$ values of the active components across the training data points in the case of the DL network parameterization (depicted in Fig. 1).

4.3. Computational Complexity

Finally, we briefly discuss the computational complexity of our approach, and how it compares to M2. We begin with prediction generation, where efficiency constitutes a significant aspect that affects the efficacy of an approach. As we have observed, IBP-DGM requires similar computational time to generate one prediction compared to the competition. Indeed, this was well-expected given the formulation of our method, since the extra posteriors [Eqs. (23)-(24)] our model introduces are amortized via deep network parameterizations; this results in fast (and highly parallelizable) feedforward computations, hence eventually inducing low computational overheads. Turning to the inference algorithm of our approach, we can report the following quite interesting finding: When using the DL parameterization, IBP-DGM requires approximately 4 times more algorithm epochs to converge compared to M2; this is the case for all the considered benchmarks. On the other hand, when using the MN parameterization, both approaches require similar numbers of epochs to converge; this is approximately 4 times more epochs compared to M2 with DL parameterization. Our interpretation of this finding is that the introduction of a mechanism that puts less or more emphasis on some latent features requires that model training proceeds more slowly. We underline though that, since this is an offline procedure, the resulting improved predictive performance is worth the entailed extra number of algorithm epochs.

5. Conclusions

In this paper, we addressed the problem of performing inference over the latent variable dimensionality of DGMs. To this end, we drew inspiration from analogous efforts in the context of traditional LLV models. Specifically, we devised a nonparametric formulation of DGMs, effected by postulating a hierarchical graphical model driven by appropriate imposition of an IBP prior. We performed inference for the so-derived IBP-DGM model by resorting to the BBVI inference scheme.

To evaluate the efficacy of the proposed approach, we applied it to semi-supervised data classification tasks, where existing DGMs have been shown to yield state-of-the-art performance; in all cases, we used benchmark datasets. As we showed, our approach is quite effective in terms of inferring the latent variable dimensionality. Indeed, we empirically found that our model retains only a fraction of the initially postulated large number of latent features. In addition, we observed that our method yields competitive classification performance compared to existing DGMs; this fact vouches for the utility of performing inference over the latent vector dimensionality of DGMs.

Our future work in this line of research focuses on extensions of our approach to DGM formulations suitable for modeling data with temporal dynamics and interdependencies, as well as to the related problem of modeling heteroscedastic time-series.

Acknowledgment

References

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. In *Proc. ICLR*, 2016.

S. Chatzis, D. Kosmopoulos, and T. Varvarigou. Signal modeling and classification using a robust latent space model based on t distributions. *IEEE Trans. Signal Processing*, 56 (3):949–963, March 2008.

Sotirios P. Chatzis. A coupled Indian buffet process model for collaborative filtering. In *Journal of Machine Learning Research: Workshop and Conference Proceedings*, volume 25: ACML 2012, pages 65–79, 2012.

Sotirios P. Chatzis. Nonparametric Bayesian multitask collaborative filtering. In *Proc. ACM CIKM*, 2013.

F. Doshi-Velez and Z. Ghahramani. Correlated non-parametric latent feature models. In *Proc. UAI*, 2009.

Finale Doshi-Velez, Kurt Miller, Jurgen Van Gael, and Yee Whye Teh. Variational inference for the Indian Buffet Process. In *Proc. AISTATS*, 2009a.

Finale Doshi-Velez, Kurt T. Miller, Jurgen Van Gael, and Yee Whye Teh. Variational inference for the Indian buffet process. Technical Report CBL-2009-001, Computational and Biological Learning Laboratory, Department of Engineering, University of Cambridge, 2009b.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2010.

S. J. Gershman and N. D. Goodman. Amortized inference in probabilistic reasoning. In *Proc. Annual Conference of the Cognitive Science Society*, 2014.

S.J. Gershman, P.I. Frazier, and D.M. Blei. Distance dependent infinite latent feature models. *IEEE Trans. Pattern Anal. and Mach. Intell.*, 37(2):334–345, 2015.

Z. Ghahramani and G.E. Hinton. The EM algorithm for mixtures of factor analyzers. Technical Report CRGTR- 96-1, Department of Computer Science, University of Toronto, Toronto, Canada, M5S 1A4, 1997.

P. W. Glynn. Likelihood ratio gradient estimation for stochastic systems. *Communications of the ACM*, 33(10):75–84, 1990.

T. Griffiths and Z. Ghahramani. Infinite latent feature models and the Indian buffet process. Technical Report TR 2005-001, Gatsby Computational Neuroscience Unit, 2005.

T. Jaakkola and M.I. Jordan. Bayesian parameter estimation via variational methods. *Statistics and Computing*, 10:25–37, 2000.

B. Jørgensen. Exponential dispersion models. *Journal of the Royal Statistical Society, Series B (Methodological)*, 49(2):127–162, 1987.

D. Kingma and M. Welling. Auto-encoding variational Bayes. In *Proc. ICLR*, 2014.

D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling. Semi-supervised learning with deep generative models. In *Proc. NIPS*, 2014.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. ICLR*, 2015.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 512:436–444, 2015.

Chongxuan Li, Jun Zhu, and Bo Zhang. Learning to generate with memory. In *Proc. ICML*, 2016.

Tsung-I Lin, Geoffrey J. McLachlan, and Sharon X. Lee. Extending mixtures of factor models using the restricted multivariate skew-normal distribution. *Journal of Multivariate Analysis*, 143:398–413, 2016.

G. McLachlan and D. Peel. *Finite Mixture Models.* Wiley Series in Probability and Statistics, New York, 2000.

E. Meeds, Z. Ghahramani, R. Neal, and S. Roweis. Modeling dyadic data with binary latent factors. In *Proc. NIPS*, 2006.

T. Miyato, S.-i. Maeda, M. Koyama, K. Nakae, and S. Ishii. Distributional smoothing with virtual adversarial training. In *Proc. ICLR*, 2016.

A. Montanari and C. Viroli. A skew-normal factor model for the analysis of student satisfaction towards university courses. *J. Appl. Statist.*, 37:473–487, 2010.

Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted Boltzmann machines. In *Proc. ICML*, 2010.

Rajesh Ranganath, Sean Gerrish, and David M. Blei. Black box variational inference. In *Proc. AISTATS*, 2014.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proc. ICML*, 2014.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proc. ICML*, 2015.

Francisco J. R. Ruiz, Michalis K. Titsias, and David M. Blei. Overdispersed black-box variational inference. In *Proc. UAI*, 2016.

Y. W. Teh, D. Gorur, and Z. Ghahramani. Stick-breaking construction for the Indian buffet process. In *Proc. AISTATS*, 2007.

M.E. Tipping and C.M. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2):443–482, 1999.

Dustin Tran, David M. Blei, Alp Kucukelbir, Adji Dieng, Maja Rudolph, and Dawen Liang. Edward: A library for probabilistic modeling, inference, and criticism, 2016. URL http://edwardlib.org.

B. Uria, I. Murray, and H. Larochelle. A deep and tractable density estimator. In *Proc. ICML*, 2014.

R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256, 1992.