INDIAN BUFFET PROCESS DEEP GENERATIVE MODELS FOR SEMI-SUPERVISED CLASSIFICATION

Sotirios P. Chatzis

Department of Electrical Eng., Computer Eng., and Informatics
Cyprus University of Technology
Limassol 3036, Cyprus

ABSTRACT

Deep generative models (DGMs) have brought about a major breakthrough, as well as renewed interest, in generative latent variable models. However, DGMs do not allow for performing data-driven inference of the number of latent features needed to represent the observed data. Traditional linear formulations address this issue by resorting to tools from the field of nonparametric statistics. Indeed, linear latent variable models imposed an Indian Buffet Process (IBP) prior 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.

Index Terms— Deep generative model, black-box variational inference, Indian Buffet Process prior.

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. 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, without resorting to cross-validation. To this end, 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 [1]. The IBP prior [2] is a nonparametric prior for latent feature models where 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. Its rationale consists in eventually utilizing 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 a part of the resulting inference algorithm [1].

Despite these advances, the linear assumptions of LLV models cannot be considered realistic in most real-world data modeling scenarios. As such, 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) [3, 4].

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 Variational Inference (BBVI) [5, 6].

The remainder of this paper is organized as follows: In Section 2, we briefly outline the methodological background of our approach. In Section 3, we introduce our approach and derive its inference algorithms. In Section 4, we perform a thorough experimental evaluation, using benchmark data. Finally, in the concluding Section, we briefly summarize 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):

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

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 likelihood function of the model, $\log p(\boldsymbol{x}|\boldsymbol{z};\boldsymbol{\theta})$, as well as the inferred approximate (variational) 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:

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

where the $\boldsymbol{\mu}(\boldsymbol{x};\boldsymbol{\phi})$ and $\boldsymbol{\sigma}^2(\boldsymbol{x};\boldsymbol{\phi})$ are outputs of deep neural networks, and $\operatorname{diag}\boldsymbol{\chi}$ is a diagonal matrix with $\boldsymbol{\chi}$ on its main diagonal. Under these assumptions, variational (approximate) inference is performed by 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 $\boldsymbol{\epsilon}$ [3]:

$$\boldsymbol{z} = \boldsymbol{\mu}(\boldsymbol{x};\boldsymbol{\phi}) + \boldsymbol{\sigma}(\boldsymbol{x};\boldsymbol{\phi}) \odot \boldsymbol{\epsilon}, \quad \text{with } \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) \quad (3)$$

where $\odot$ is the elementwise product between vectors. Specifically, these samples are used to approximate the intractable posterior expectations in (1), in a way that results in low-variance estimators, $\boldsymbol{\phi}$, of the sought posterior, $q(\boldsymbol{z};\boldsymbol{\phi})$ [3].

2.2. Nonparametric Modeling Using the IBP Prior

The IBP 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. Then, a formulation of the IBP that renders $p(\boldsymbol{Z})$ amenable to variational inference consists in the following hierarchical representation [7]:

$$z_{ik} \sim \text{Bernoulli}(\pi_k)\, \forall i \quad (4)$$

$$\pi_k = \prod_{j=1}^{k} v_j,\ v_k \sim \text{Beta}(\alpha, 1)\ \forall k \quad (5)$$

2.3. BBVI

BBVI is an effective means of performing variational inference for DGM variants that entail *discrete random 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" [8] to obtain Monte Carlo estimates of the gradient that reads

$$\nabla_{\boldsymbol{\phi}}\mathcal{L}(\boldsymbol{\phi}) = \mathbb{E}_{q(\boldsymbol{z};\boldsymbol{\phi})}[f(\boldsymbol{z})] \quad (6)$$

where

$$f(\boldsymbol{z}) = \nabla_{\boldsymbol{\phi}}\log q(\boldsymbol{z};\boldsymbol{\phi})\,[\log p(\boldsymbol{x},\boldsymbol{z}) - \log q(\boldsymbol{z};\boldsymbol{\phi})] \quad (7)$$

Then, 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

$$\nabla_{\boldsymbol{\phi}}\mathcal{L}(\boldsymbol{\phi}) = \sum_{n=1}^{N} \mathbb{E}_{q(\boldsymbol{z};\boldsymbol{\phi})}[f_n(\boldsymbol{z}) - a_n h_n(\boldsymbol{z})] \quad (8)$$

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

$$h_n(\boldsymbol{z}) = \nabla_{\boldsymbol{\phi}}\log q(\boldsymbol{z}_n;\boldsymbol{\phi}) \quad (9)$$

and the constants a_n are given by [5]

$$a_n = \frac{\text{Cov}\,(f_n(\boldsymbol{z}), h_n(\boldsymbol{z}))}{\text{Var}\,(h_n(\boldsymbol{z}))} \quad (10)$$

On this basis, derivation of the sought variational posteriors is performed by utilizing the gradient expression (8) in the context of off-the-shelf stochastic gradient optimizers. Specifically, in this work we utilize AdaM [9].

3. PROPOSED APPROACH

Let us consider the dataset $X = \{\boldsymbol{x}_i\}_{i=1}^{N}$. The proposed IBP-DGM model 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 $\boldsymbol{z}_i$:

$$\boldsymbol{z}_i = \tilde{\boldsymbol{z}}_i \cdot \hat{\boldsymbol{z}}_i \tag{11}$$

$$p(\tilde{\boldsymbol{z}}_i) = \mathcal{N}(\tilde{\boldsymbol{z}}_i|\boldsymbol{0}, \boldsymbol{I}) \tag{12}$$

$$p(\hat{\boldsymbol{z}}_i) = \prod_{k=1}^{K\to\infty} \text{Bernoulli}(\hat{z}_{ik}|\pi_k) \tag{13}$$

$$\pi_k \triangleq \prod_{j=1}^{k} v_j,\ k \in \{1,\ldots,\infty\} \tag{14}$$

$$p(v_k) = \text{Beta}(v_k|\alpha, 1),\ k \in \{1,\ldots,\infty\} \tag{15}$$

The introduction of the binary latent variables $\hat{\boldsymbol{z}}_i$ in Eq. (11) essentially allows for the model to infer which latent features $\tilde{z}_{ik},\ k \in \{1,\ldots,K\to\infty\}$, 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.

Under the infinite dimensional setting prescribed in Eqs. (11)-(15), 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 [7]. 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 $k > K\ \forall i$. We then postulate:

$$q(\tilde{\boldsymbol{z}}_i; \boldsymbol{\phi}) = \mathcal{N}(\tilde{\boldsymbol{z}}_i|\boldsymbol{\mu}(\boldsymbol{x}_i; \boldsymbol{\phi}), \text{diag}\ \boldsymbol{\sigma}^2(\boldsymbol{x}_i; \boldsymbol{\phi})) \tag{16}$$

$$q(\hat{\boldsymbol{z}}_i; \boldsymbol{\phi}) = \prod_{k=1}^{K} \text{Bernoulli}(\hat{z}_{ik}|\hat{\pi}_k(\boldsymbol{x}_i; \boldsymbol{\phi})) \tag{17}$$

$$q(v_k; \boldsymbol{\phi}) = \text{Beta}(v_k|a_k(\boldsymbol{x}_i; \boldsymbol{\phi}), b_k(\boldsymbol{x}_i; \boldsymbol{\phi})),\ k \in \{1,\ldots,K\} \tag{18}$$

Here, 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. Finally, we impose a simple spherical prior over the likelihood parameters $\boldsymbol{\theta}$:

$$p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{0}, \sigma_\theta^2 \boldsymbol{I}) \tag{19}$$

In addition, to facilitate computational efficiency, we consider that the sought variational posterior $q(\boldsymbol{\theta})$ collapses to a single point, $\hat{\boldsymbol{\theta}}$, that essentially constitutes a point-estimate; i.e., $q(\boldsymbol{\theta}) = \delta_{\hat{\boldsymbol{\theta}}}(\boldsymbol{\theta})$, 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 IBP-DGM model. Even though IBP-DGM is a generative model, we can use it to perform semi-supervised learning. To this end, we only need to modify the likelihood function so as to take into account (possible) label information. Specifically, we postulate a different class-conditional likelihood function for each class label, y, of the form $p(\boldsymbol{x}|\tilde{\boldsymbol{z}} \cdot \hat{\boldsymbol{z}}, y; \boldsymbol{\theta})$; this is employed for all the labeled training data points belonging to the corresponding class. On the other hand, we continue to use the likelihood function $p(\boldsymbol{x}|\tilde{\boldsymbol{z}} \cdot \hat{\boldsymbol{z}}; \boldsymbol{\theta})$ for the available unlabeled data points. Finally, we also need to introduce a prior $p(y)$ over the labels y of the observed data, as well as a corresponding variational posterior $q(y; \boldsymbol{\phi})$. We have

$$p(y = c) = \frac{1}{C},\ \forall c \tag{20}$$

and

$$q(y; \boldsymbol{\phi}) = \text{Cat}(y|\varpi(\boldsymbol{x})) \tag{21}$$

where $\varpi(\boldsymbol{x})$ is parameterized via a deep network, and C is the total number of classes.

Then, variational inference is performed by resorting to BBVI, as described in the previous Section. The ELBO expression of the model reads:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\phi}; \boldsymbol{\theta}) =& \mathbb{E}_{q(\tilde{\boldsymbol{z}}; \boldsymbol{\phi})}[p(\tilde{\boldsymbol{z}}) - \log q(\tilde{\boldsymbol{z}}; \boldsymbol{\phi})] \\ &+ \mathbb{E}_{q(\hat{\boldsymbol{z}}; \boldsymbol{\phi})}[p(\hat{\boldsymbol{z}}) - \log q(\hat{\boldsymbol{z}}; \boldsymbol{\phi})] \\ &+ \mathbb{E}_{q(y; \boldsymbol{\phi})}[p(y) - \log q(y; \boldsymbol{\phi})] \\ &+ \sum_{\boldsymbol{x}_i : y_i = y} \mathbb{E}_{q(\tilde{\boldsymbol{z}}, \hat{\boldsymbol{z}}; \boldsymbol{\phi})}[p(\boldsymbol{x}_i|\tilde{\boldsymbol{z}}_i \cdot \hat{\boldsymbol{z}}_i, y; \boldsymbol{\theta})] \\ &+ \sum_{\boldsymbol{x}_i : y_i = \emptyset} \mathbb{E}_{q(\tilde{\boldsymbol{z}}, \hat{\boldsymbol{z}}; \boldsymbol{\phi})}[p(\boldsymbol{x}_i|\tilde{\boldsymbol{z}}_i \cdot \hat{\boldsymbol{z}}_i; \boldsymbol{\theta})]\end{aligned} \tag{22}$$

4. EXPERIMENTS

To exhibit the efficacy of our approach, we perform evaluation using the MNIST, Rotated MNIST+Background Images, MNIST+Background Images, MNIST+Random Background, Rotated MNIST, and (Small-)NORB benchmarks.[1] 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. To provide some comparative results, apart from our method we also evaluate the M2 approach proposed in [12], which constitutes the parametric equivalent of IBP-DGM in the context of semi-supervised learning. We consider two alternative architectures of the deep networks parameterizing the postulated likelihood and posterior distributions of IBP-DGM. The first alternative comprises simple Dense Layer (DL) architectures. The second one is based on the Memory Network (MN) architecture recently proposed in [13]. This employs an external hierarchical memory to capture variant information at different abstraction levels trained in an unsupervised manner.

[1] Before each epoch, the normalized MNIST images are binarized by sampling Bernoulli distributions, similar to [10]. We normalize all NORB images following the procedure suggested in [11]; we add uniform noise between 0 and 1 to each pixel value, to allow for effectively modeling them by means of Gaussian conditional likelihoods, $p(\boldsymbol{x}|\tilde{\boldsymbol{z}} \cdot \hat{\boldsymbol{z}}, y; \boldsymbol{\theta})$ and $p(\boldsymbol{x}|\tilde{\boldsymbol{z}} \cdot \hat{\boldsymbol{z}}; \boldsymbol{\theta})$.

In all our experiments, for simplicity and computational efficiency, 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 [14]. Initialization of the network parameters is performed by adopting a Glorot-style uniform initialization scheme [15]. The used MN layers comprise 100 memory slots; that is the number of rows of matrix $\boldsymbol{A}$, or, conversely, the number of columns of the memory matrix $\boldsymbol{M}$ [13]. In all cases, the maximum size of the postulated latent vectors $\boldsymbol{z}$ (truncation threshold K of the variational posterior) is set to 50.[2] Our source codes have been developed in Python, and make use of the Tensorflow library [16].

In Tables 1 and 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.

Another 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 or less emphasis on some latent characteristics of the data. 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 retention or omission.

Note also that IBP-DGM requires similar computational time to generate one prediction compared to the competition. Turning to the training 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 one M2 network; 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 one M2 network 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.

Finally, it is interesting to 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. As we have observed, our model tends to yield high enough posterior values only for the first 10-12 latent components. 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: When both mechanisms are used, they tend to reinforce each other. This results in a lower standard deviation for the $q(\hat{z}_{\cdot k};\boldsymbol{\phi})$ values of the active components across the training data points.

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

5. CONCLUSIONS

In this paper, we addressed the problem of performing inference over the latent variable dimensionality of DGMs. To this end, we devised a nonparametric formulation of DGMs, obtained by utilizing the IBP prior. We performed inference for the so-derived IBP-DGM model by resorting to the BBVI inference scheme. As we showed, our approach is quite effective in terms of inferring the latent variable dimensionality, and yields competitive classification performance. Remarkably, the observed modeling and predictive performance improvement did not come at the cost of extra computational overheads. Our future work will focus on extending DGMs so as to model heteroscedastic data, e.g. [17, 18], as well as data with temporal dynamics of unknown order, e.g. [19].

[2] In each case, the prior variance σ_θ^2 of the model parameters $\boldsymbol{\theta}$ is heuristically selected among the alternative values $\{10^{-3}, 10^{-2}, 10^{-1}\}$, with the aim of maximizing out-of-sample predictive performance. To execute AdaM, 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.

6. REFERENCES

[1] Sotirios P. Chatzis, "Nonparametric Bayesian multitask collaborative filtering," in *Proc. ACM CIKM*, 2013.

[2] T. Griffiths and Z. Ghahramani, "Infinite latent feature models and the Indian buffet process," Tech. Rep. TR 2005-001, Gatsby Computational Neuroscience Unit, 2005.

[3] D. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. ICLR*, 2014.

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

[5] Rajesh Ranganath, Sean Gerrish, and David M. Blei, "Black box variational inference," in *Proc. AISTATS*, 2014.

[6] Francisco J. R. Ruiz, Michalis K. Titsias, and David M. Blei, "Overdispersed black-box variational inference," in *Proc. UAI*, 2016.

[7] Finale Doshi-Velez, Kurt Miller, Jurgen Van Gael, and Yee Whye Teh, "Variational inference for the Indian Buffet Process," in *Proc. AISTATS*, 2009.

[8] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[9] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.

[10] B. Uria, I. Murray, and H. Larochelle, "A deep and tractable density estimator," in *Proc. ICML*, 2014.

[11] T. Miyato, S. Maeda, M. Koyama, K. Nakae, and S. Ishii, "Distributional smoothing with virtual adversarial training," in *Proc. ICLR*, 2016.

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

[13] Chongxuan Li, Jun Zhu, and Bo Zhang, "Learning to generate with memory," in *Proc. ICML*, 2016.

[14] Vinod Nair and Geoffrey E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. ICML*, 2010.

[15] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. AISTATS*, 2010.

[16] Martín Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, Software available from tensorflow.org.

[17] Emmanouil A. Platanios and Sotirios P. Chatzis, "Gaussian process-mixture conditional heteroscedasticity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 5, pp. 888–900, 2014.

[18] Sotirios P. Chatzis and Y. Demiris, "Nonparametric mixtures of Gaussian processes with power-law behavior," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, pp. 1862–1871, Dec. 2012.

[19] Sotirios P. Chatzis and Yiannis Demiris, "The infinite-order conditional random field model for sequential data modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1523–1534, 2013.