2016 23rd International Conference on Pattern Recognition (ICPR)
Cancún Center, Cancún, México, December 4-8, 2016

Latent Regression Bayesian Network for Data Representation

Siqi Nie
Department of Electrical, Computer and Systems Engineering
Rensselaer Polytechnie Institute
Email: nies@rpi.edu

Yue Zhao
School of Information Engineering
Minzu University of China
Email: zhaoyueso@gmail.com

Qiang Ji
Department of Electrical, Computer and Systems Engineering
Rensselaer Polytechnie Institute
Email: qji@ecse.rpi.edu

***Abstract*—Restricted Boltzmann machines (RBMs) are widely used for data representation and feature learning in various machine learning tasks. The undirected structure of an RBM allows inference to be performed efficiently, because the latent variables are dependent on each other given the visible variables. However, we believe the correlations among latent variables are crucial for faithful data representation. Driven by this idea, we propose a counterpart of RBMs, namely latent regression Bayesian networks (LRBNs), which has a directed structure. One major difficulty of learning LRBNs is the intractable inference. To address this problem, we propose an inference method based on the conditional pseudo-likelihood that preserves the dependencies among the latent variables. For learning, we propose to employ the hard Expectation Maximization (EM) algorithm, which avoids the intractability of the traditional EM by max-out instead of sum-out to compute the data likelihood. Qualitative and quantitative evaluations of our model against state-of-the-art models and algorithms on benchmark data sets demonstrate the effectiveness of the proposed algorithm in data representation and reconstruction.**

I. Introduction

In the current "Age of Big Data", it is essential to design models and algorithms to handle large amount of unlabeled data, since labeling data is a time consuming task. To represent the patterns in unlabeled data, generative models have received increasing attention. The restricted Boltzmann machine (RBM) is one of the most successful generative models with state-of-the-art performance in many machine learning tasks. One reason behind the success of RBMs is that the special undirected structure allows inference to be performed efficiently. Compared with RBMs, their counterpart, directed generative models have been left behind, mainly due to the intractable inference. However, directed models have their own advantages. First, samples can be easily obtained by straightforward ancestral sampling without the need for Markov chain Monte Carlo (MCMC) methods. Second, there is no partition function issue since the joint distribution is obtained by multiplying all local conditional probabilities, which requires no further normalization. Last but most importantly, the latent variables are dependent on each other given the observations through the so-called "explain-away" principle. Through their inter dependency, latent variables coordinate with each other to better explain the patterns in the visible layer.

Learning directed models with many latent variables is challenging, mainly due to the intractable computation of the posterior probability, and thus conventional EM algorithm is not an option. Although Markov Chain Monte Carlo (MCMC) method is a straightforward solution, the mixing stage is often too slow. To address the problem, one common approach is the variational inference, which replaces the true posterior distribution with a tractable distribution (typically factorized) as an approximation. The mean field theory [18] assumes the latent variables to be totally independent. Some recent efforts for learning generative models have focused on feed-forward structures for inference [10, 11, 13]. To obtain the inference distribution, the KL-divergence to the true posterior distribution is minimized. In all the aforementioned inference algorithms, the approximating distribution is typically fully factorized for computational efficiency. However, the assumption of the factorized distribution sacrifices the " explain-away" effect for efficient inference, which inevitably enlarges the distance to the true posterior, and weakens the representation power of the model. This defeats a major advantage of directed graphical models.

In terms of parameter learning, one straightforward way is to maximize the log-likelihood given data, but it is typically intractable because of the log-sum-exp function with exponential terms in the summation. The Contrastive Divergence (CD) [6] approximate the gradient of parameters with samples. Alternatively, learning can be performed by maximizing a lower bound of the log-likelihood. The EM algorithm repeatedly construct a lower bound of the data log-likelihood with the current parameter for learning mixture of factor analyzers [5], probabilistic latent semantic indexing [8], probabilistic latent semantic analysis [9] and latent Dirichlet allocation (LDA) [1]. All the approximate distribution to the posterior probability discussed above can be plugged into the lower bound, resulting in a more tractable objective function.

In this work, we propose to employ a directed model called latent regression Bayesian network (LRBN) with one layer of latent variables and one layer of visible variables. The LRBNs can be used as building blocks to construct a deep generative model, similar to RBMs. We propose to use the EM algorithm with two approximations in the inference and learning phases. First, we approximate the true poste-

978-1-5090-4846-5/16/$31.00 ©2016 IEEE

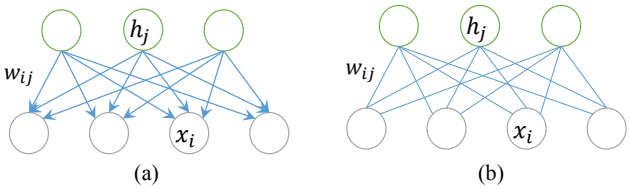


Fig. 1. Graph representation of the (a) LRBN and (b) RBM. Each link is associated with a weight parameter.

rior distribution during inference by the conditional pseudo-likelihood, which preserves to certain degree the dependencies among latent variables. Second, we approximate the data likelihood using a max-out setting during the E-step of the learning to overcome the exponential number of configurations of the latent variables. As a result, the E-step requires the maximum a posteriori (MAP) inference, which is efficiently solved based on the pseudo-likelihood. In the M-step, the problem is transferred into parameter learning with complete data, which is much easier to handle. We theoretically show that the objective function of the hard EM algorithm is a lower bound of the one of conventional EM algorithm. Experiments pn benchmark data sets demonstrate the effectiveness of the proposed algorithm in terms of data representation.

II. Latent Regression Bayesian Network

We propose to employ a directed graphical model, called latent regression Bayesian network (LRBN) [14], as shown in Fig. 1 (a). The visible variables $x \in \mathbb{B}^{n_d}$ and latent variables $h \in \mathbb{B}^{n_h}$ in LRBN are binary. Each latent variable is connected to all visible variables. We assume that the latent variables h determine the patterns in data x; therefore directed links are used to model their relationships, as in a Bayesian network.

Prior probability for latent variables is represented as a log-liner model,

$$P(h_j = 1) = \sigma(d_j)\,, \tag{1}$$

where d_j is the parameter defining the prior distribution for node h_j; $\sigma(\cdot)$ is the sigmoid function $\sigma(z) = 1/(1+e^{-z})$.

The conditional probability of a visible variable x_i given the latent variables is,

$$P(x_i = 1|h) = \sigma(\sum_j w_{ij}h_j + b_i)\,, \tag{2}$$

where w_{ij}, b_i are the parameters.

The joint probability is factorized into the product of the prior probabilities and the conditional probabilities,

$$P_\theta(x,h) = \frac{1}{\prod_j(1+\exp(d_j))} \exp\left(\sum_{i,j} w_{ij}x_ih_j + \sum_i b_ix_i + \sum_j d_jh_j - \sum_i \log\left(1+\exp(\sum_j w_{ij}h_j + b_i)\right)\right). \tag{3}$$

In this case, the model becomes a sigmoid belief network (SBN) [12] with one latent layer. If more layers are added on top, the conditional probability is defined in the same way as Eq. 2. The model is named a regression model based on the nature of the conditional probability.

As a similar model with undirected links, RBMs (Fig. 1 (b)) have been widely used in the literature for feature learning and data representation. The joint probability defined by a discrete RBM is,

$$P_{\text{RBM}}(x,h) = \frac{1}{Z}\exp\left(\sum_i b_ix_i + \sum_{i,j} w_{ij}x_ih_j + \sum_j d_jh_j\right). \tag{4}$$

Comparing Eq. 3 and 4, the additional terms in the numerator captures the correlations among them. This is the reason why $P(h|x)$ is not factorized over individual latent nodes h_j in LRBN, which is the major difference from RBM. Another advantage of Eq. 3 is that every term can be computed given the values of all variables, without the issue of the intractable partition function Z.

III. LRBN Inference

In this section, we introduce an efficient inference method for LRBN based on conditional pseudo-likelihood.

Given a LRBN model with known parameters, one task of probabilistic inference is to compute the posterior probability of the latent variables given input data, i.e., computing $P(h|x)$.

The maximum a posteriori (MAP) inference, which is to find the configuration of latent variables that maximizes the posterior probability given observations,

$$h^* = \operatorname*{argmax}_h P(h|x)\,. \tag{5}$$

The MAP inference is motivated by the observation that from the data generating point of view, the variables in one latent layer take values according to the conditional probability given its upper layer. Therefore this configuration dominates all the others in explaining each data sample. In addition, the goal of feature learning is to learn a feature h that best explains x. In this regard, we only care about the most probable states of the latent variables given the observation.

Due to the facts that the total number of configurations of latent variables are exponential in the size of the latent layer, and that the latent variables are dependent on each other, direct computing $P(h|x)$ is computationally intractable.

The pseudo-likelihood replaces the conditional likelihood by a more tractable objective, i.e.,

$$P(h|x) \approx \prod_j P(h_j|h_{-j}, x)\,, \tag{6}$$

where $h_{-j} = \{h_1, \ldots, h_{j-1}, h_{j+1}, \ldots, h_{n_h}\}$ is the set of all latent variables except h_j. In this approximation, we add conditioning over additional variables.

The conditional pseudo-likelihood preserves the rich dependencies among latent variables given the input. It is a suitable approximate distribution compared to the mean field approximation or an inference network, in particular in a densely connected network, where the dependencies is an essential

property. Moreover, there is no additional parameters needed for the conditional pseudo-likelihood, unlike the variational inference.

The conditional pseudo-likelihood can be factorized into local conditional probabilities, which can be estimated in parallel. To optimize over the pseudo-likelihood, one latent variable is updated by fixing all other variables,

$$h_j^{t+1} = \operatorname*{argmax}_{h_j} P(h_j|x, h_{-j}^t)\,, \quad 1 \le j \le n_h\,, \tag{7}$$

where t denotes the t^{th} iteration.

The updating rule (Eq. 7) guarantees that the posterior probability $P(h|x)$ will only increase or stay the same after each iteration.

$$P(h_j^{t+1}, h_{-j}^t|x) \ge P(h^t|x)\,. \tag{8}$$

In general, computing the joint probability $P(x, h)$ has complexity $O(n_d n_h)$. If each latent variable is updated t times to get h^*, the overall complexity for the LRBN inference is $O(t n_d n_h^2)$, which is much lower than the $O(2^{n_h} n_d n_h)$ complexity when computing $P(h|x)$ directly.

The pseudo-likelihood based updating rule can be seen as a coordinate ascent algorithm applied to the true posterior probability $P(h|x)$, or the iterated conditional modes (ICM) as in inference of Markov random fields.

The inference method requires an initialization for the hidden variables. Different initializations will end up with different local optimal points. Multiple different initializations are used, and the best MAP configuration is used as the final result.

IV. LRBN Learning

In this section, we introduce an efficient LRBN learning method based on the hard Expectation Maximization (EM) algorithm. The conventional EM algorithm is not an option here due to the intractability of computing posterior probability in the E-step. The hard version of EM algorithm has been explored in [19] for learning a deep Gaussian mixture model. This model has a deep structure in terms of linear transformations, but only has two layers of variables.

Consider the model $P_\theta(x, h)$ defined in section 3. The goal of parameter learning is to estimate the parameters $\theta = \{w, b, d\}$ given a set of data samples $D = \{x^{(m)}\}_{m=1}^M$. The conventional maximum likelihood (ML) parameter estimation is to maximize the following objective function,

$$\theta^* = \operatorname*{argmax}_\theta \sum_m \log \sum_h P_\theta(x^{(m)}, h)\,. \tag{9}$$

The second summation in Eq. 9 is intractable due to the exponentially many configurations of h. In this work, we employ a max-out estimation of the data log-likelihood, with the following objective function,

$$\theta^* = \operatorname*{argmax}_\theta \sum_m \log \max_h P_\theta(x^{(m)}, h)\,. \tag{10}$$

Algorithm 1 Parameter learning of an LRBN with one latent layer.

Input training data $X = \{x^{(m)}\}$
Output parameters θ of an LRBN

1: Initial parameters θ;
2: Initialize the states h_0 for all the latent variables using some feed-forward model (Section III);
3: **while** parameters not converging, **do**
4: Random select a minibatch of data instances $x \in X$
5: Update the corresponding h for x by maximizing the posterior probability, using current parameters,

$$h^* = \operatorname*{argmax}_h P_\theta(x, h)\,. \tag{13}$$

6: Compute the gradients using Eq. 11 and 12. Update the parameters,

$$\theta = \theta + \lambda \nabla_\theta \log P(x, h^*) \tag{14}$$

7: **end while**

Note that the max-out approximation of the data likelihood is a lower bound of the marginal likelihood. It is not equivalent to approximating $P(h|x)$ with a delta function.

With objective function Eq. 10, the learning method becomes a hard version of the EM algorithm, which iteratively fills in the latent variables and update the parameters. In the E-step, $h^* = \operatorname{argmax}_h P(x, h)$ is effectively estimated using the proposed inference method based on pseudo-likelihood and coordinate ascent. In the M-step, the problem of parameter estimation is straightforward because now we are dealing with complete data.

The gradient of the parameters in terms of one data sample x is,

$$\frac{\partial \log P(x, h)}{\partial w_{ij}} = h_j \left(x_i - P(x_i = 1|h)\right)\,, \tag{11}$$

$$\frac{\partial \log P(x, h)}{\partial b_i} = x_i - P(x_i = 1|h)\,. \tag{12}$$

In case of large data sets, it is time consuming because all the training instances are used to compute the gradient. Stochastic gradient ascent algorithm can be used to address this issue. The true gradient is approximated by the gradient at a minibatch of training samples. The proposed learning method for LRBN is summarized in Algorithm 1.

It is known that the objective function of conventional EM algorithm is a lower bound of the marginal likelihood,

$$\begin{aligned} &\sum_m \log P(x^{(m)}; \theta) \\ &\ge \sum_m \sum_{(m)} P(h^{(m)}|x^{(m)}; \theta^t) \log \frac{P(x^{(m)}, h^{(m)}; \theta)}{P(h^{(m)})|x^{(m)}; \theta^t)}\,. \end{aligned} \tag{15}$$

In this paper, we shown that the objective function of hard EM is a lower bound of the conventional EM algorithm,

$$\begin{aligned}&\sum_m \log P(x^{(m)}, h^{(m)*}; \theta) \\ \leq &\sum_m \sum_{(m)} P(h^{(m)}|x^{(m)}; \theta^t) \log \frac{P(x^{(m)}, h^{(m)}; \theta)}{P(h^{(m)})|x^{(m)}; \theta^t)} \,.\end{aligned} \tag{16}$$

where $h^{(m)*}$ is the MAP configuration for data sample $x^{(m)}$.

This inequality holds if we can prove that,

$$\frac{P(h^*|x;\theta)}{P(h|x;\theta)} \leq \frac{1}{P(h|x;\theta^t)} \,, \tag{17}$$

for any configuration h.

Eq. (17) holds under two mild assumptions:

Assumption 1: The posterior probability $P(h|x)$ changes in a small range after each parameter update,

$$\frac{P(h|x;\theta^t)}{P(h|x;\theta^{t+1})} = O(1) \,. \tag{18}$$

Assumption 2: The posterior probability $P(h|x)$ much less than 1,

$$P(h^*|x;\theta^{t+1}) \ll 1 \,. \tag{19}$$

Assumption 1 is reasonable because the parameters are updated through gradient ascent with a fixed learning rate, which means two consecutive iterations should have similar parameters. Assumption 2 is reasonable because among all the 2^{n_h} configurations of latent variables, even the largest one has a very small probability. Therefore, hard EM algorithm actually optimizes a lower bound of the objective function of the EM algorithm, which already maximizes a lower bound of the marginal likelihood.

V. Experiments

In this section, we evaluate the performance of LRBN and compare against other methods on three data sets with binary variables: MNIST, Caltech 101 Silhouettes and OCR letters. The experiments will evaluate representation and reconstruction power of the proposed model.

A. Experimental protocol

We trained the LRBN model using stochastic gradient ascent algorithm with learning rate 0.002. The size of the minibatches is set to 20. For each data set, we randomly selected 100 samples from the training set to form a validation set. The joint probability on the validation set is a criterion for early stopping.

We first evaluate the MAP configuration of the latent variables through reconstruction. Reconstruction is performed as follows: given a data vector x, perform a MAP inference to get $h^* = \text{argmax}_h P(h|x)$. Then $\tilde{x} = \text{argmax}_x P(x|h^*)$ is the reconstructed data. The reconstruction error $|\tilde{x} - x|^2$ can evaluate how well the model fits the data.

The second criterion is the widely used test data log-likelihood. Directly computing probability $P(x)$ is intractable due to the exponentially many terms in the summation $P(x) = \sum_h P(x, h)$. In this work, we employ the Anneal Importance Sampling method [2]. One million samples are used to estimate the log-likelihood, and the average of ten repetitions is reported.

The reconstruction error evaluates the quality of the most probable explanation of the latent variables given observations, while the data log-likelihood evaluates the overall quality of all configurations of latent variables. They are two complementary criteria for model evaluation.

In the experiment, we compare the proposed LRBN with two learning algorithms for directed models, namely variational Bayes (VB) [4] and neural variational inference and learning (NVIL) [11], and one undirected model, namely RBM. We compare with published results if they are available. We implement the NVIL and RBM following [3, 11]. Similar log-likelihood achieved by our implementation indicates the correctness of the implementation. The code for VB is publicly available online. The size of the latent layer is set to 200, consistent with the configurations in [4, 11]. Therefore all models have the same amount of latent variables and parameters.

B. The MNIST Dataset

The first experiment is performed on the binary version of the MNIST data set. The data set consists of 70,000 handwritten digits with dimension 28×28. It is partitioned into a training set of 60,000 images and a testing set of 10,000 images.

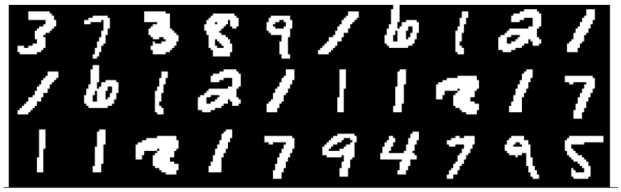
The average reconstruction errors of different learning models are reported in Table I. The MAP inference of NVIL is through the inference network. The average reconstruction error of the proposed model is 4.56 pixels, which significantly outperforms the other competing methods by at least 25 pixels. This is consistent with our objective function, indicating the most probable explanation contains most information in the input data, which is effectively captured in the proposed model. Some examples of the reconstruction are shown in Fig. 2. As can be seen, it is difficult to tell the difference between the reconstructed digits and the original ones.

TABLE I
AVERAGE RECONSTRUCTION ERRORS OF DIFFERENT METHODS ON THE MNIST DATA SET.

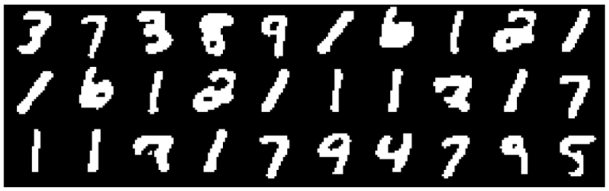
Method	Recon Error
NVIL	36.13
VB	35.71
RBM	30.71
LRBN	4.37

TABLE II
TEST DATA LOG-LIKELIHOODS OF DIFFERENT MODELS USING ON THE MNIST DATA SET.

Method	10k
NVIL	-113.1
VB	-116.9
RBM	-104.2
LRBN	-105.4



(a)



(b)

Fig. 2. Examples of the reconstruction. (a) Original digit images. (b) reconstructed by LRBN.

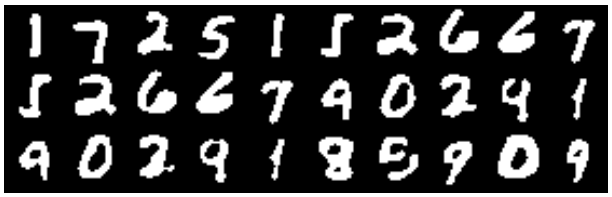


Fig. 3. Random samples from the generative model on the MNIST data set.

In Table II we report the average log-likelihood on the test set. With the same dimensionality, LRBN achieves a log-likelihood of -103.5 nats, outperforming VB and NVIL, and similar to RBM. Even though our objective function does not explicitly maximize the log-likelihood, the learned model achieves comparable performance compared with state-of-the-art learning methods, which indicates that the proposed method is also effective in capturing the distribution of the training data. Some samples from the generative model are given in Fig. 3.

C. The Caltech 101 Silhouettes Dataset

The second experiment is performed on the Caltech 101 Silhouettes data set. The data set contains 6364 training images and 2307 testing images. The dimension of the image is 28×28. Each image in the data set includes a polygon outline of the object in the scene.

The reconstruction error is reported in Table III. The proposed learning method obtains a reconstruction error of 5.95 pixels, outperforming all the competing methods by a large margin, indicating the effectiveness of the max-out approximation.

The test data log-likelihood is reported in Table IV. LRBN achieves a test log-likelihood of -103.5 nats. With the same dimensionality, LRBN learned with hard EM outperforms the one learned by VB and NVIL. The improvement is over 24 nats. Moreover, compared to an RBM, our model also achieves better performance, indicating the importance of the underlying dependency of the latent variables. Examples are shown in Fig. 4.

TABLE III
AVERAGE RECONSTRUCTION ERRORS OF DIFFERENT METHODS ON THE CALTECH 101 SILHOUETTES DATA SET.

Method	Recon Error
NVIL	29.78
VB	30.75
RBM	32.47
LRBN	5.95

TABLE IV
TEST DATA LOG-LIKELIHOOD OF DIFFERENT METHODS ON THE CALTECH 101 SILHOUETTES DATA SET.

Method	Log-prob
NVIL	127.9
VB	-136.8
RBM	-107.8
LRBN	-103.5

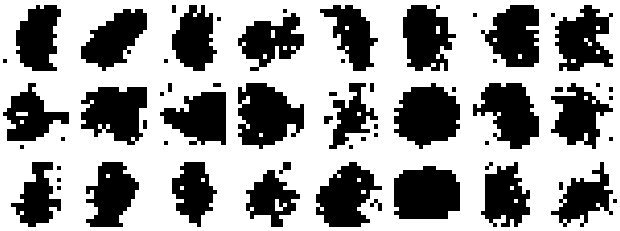


Fig. 4. Random samples from LRBN on Caltech 101 Silhouettes data set.

D. The OCR Letters Dataset

The last experiment is performed on the OCR letters data set, which contains 42,152 training images and 10,000 testing images of English letters. The images have the dimensionality of 16×8.

The reconstruction error is reported in Table V. The proposed method shows superior performance compared to all the competing methods. The average reconstruction error on the test set is 2.15 pixels, which is at least 12 pixels better than the other methods.

The test data log-likelihood is reported in Table VI. Our model obtains a log-likelihood -38.7, which outperforms all the competing algorithms.

Samples from the LRBN are shown in Fig. 5. We display the samples of letter 'g'. For the same letter, the learned

TABLE V
AVERAGE RECONSTRUCTION ERRORS OF DIFFERENT METHODS ON THE OCR LETTERS DATA SET.

Method	Recon Error
NVIL	15.42
VB	14.37
RBM	16.83
LRBN	2.15

TABLE VI
TEST DATA LOG-LIKELIHOOD ON THE OCR LETTERS DATA SET.

Method	Log-prob
NVIL	-47.2
VB	-48.2
RBM	-40.8
LRBN	-38.7

model is able to capture the different handwriting styles, while preserving the key information.

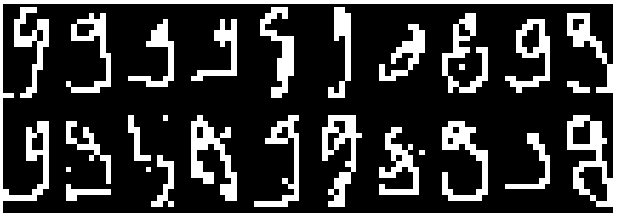


Fig. 5. Random samples from LRBN on the OCR letters data set.

VI. CONCLUSION

In this work, we introduce a directed model based on the latent regression Bayesian network to explicitly capture the dependencies among the latent variables for data representation. We introduce an efficient inference method based on pseudo-likelihood and coordinate ascent. A hard EM learning method is proposed for efficient parameter learning. The proposed inference method solve the inference intractability, while preserving the dependencies among latent variables. We theoretically and empirically compare different models and learning methods. We point out that the latent variables in regression Bayesian network have strong dependencies, which can better explain the patterns in the input layer. Experiments on benchmark data sets shows the proposed model significantly outperforms the existing models in data reconstruction and achieves comparable performance for data representation.

ACKNOWLEDGMENT

The work described in this paper is supported in part by an award from the National Science Foundation under the grant number IIS 1539012.

REFERENCES

[1] David Blei, Andrew Ng, and Michael Jordan. Latent dirichlet allocation. *the Journal of Machine Learning Research*, 3:993–1022, 2003.

[2] David Carlson, Ya-Ping Hsieh, Edo Collins, Lawrence Carin, and Volkan Cevher. Stochastic spectral descent for discrete graphical models. *IEEE Journal of Selected Topics in Signal Processing*.

[3] KyungHyun Cho, Tapani Raiko, and Alexander Ilin. Enhanced gradient for training restricted Boltzmann machines. *Neural Computation*, 25(3):805–831, 2013.

[4] Zhe Gan, Ricardo Henao, David Carlson, and Lawrence Carin. Learning deep sigmoid belief networks with data augmentation. *International Conference on Artificial Intelligence and Statistics*, 2015.

[5] Zoubin Ghahramani and Geoffrey E Hinton. The EM algorithm for mixtures of factor analyzers. Technical report, Technical Report CRG-TR-96-1, University of Toronto, 1996.

[6] Geoffrey Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14 (8):1771–1800, 2002.

[7] Geoffrey Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

[8] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 50–57. ACM, 1999.

[9] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc., 1999.

[10] Diederik Kingma and Max Welling. Auto-encoding variational Bayes. *In Proceedings of the International Conference on Learning Representations*, 2014.

[11] Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. *In Proceedings of the 31st International Conference on Machine Learning*, 2014.

[12] Radford Neal. Connectionist learning of belief networks. *Artificial Intelligence*, 56(1):71–113, 1992.

[13] Danilo Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1278–1286, 2014.

[14] Frank Rijmen. Bayesian networks with a logistic regression model for the conditional probabilities. *International Journal of Approximate Reasoning*, 48(2):659–666, 2008.

[15] Ruslan Salakhutdinov and Geoffrey E Hinton. Deep Boltzmann machines. In *International Conference on Artificial Intelligence and Statistics*, pages 448–455, 2009.

[16] Ruslan Salakhutdinov and Hugo Larochelle. Efficient learning of deep Boltzmann machines. In *International Conference on Artificial Intelligence and Statistics*, pages 693–700, 2010.

[17] Ruslan Salakhutdinov and Iain Murray. On the quantitative analysis of deep belief networks. In *Proceedings of the 25th International Conference on Machine Learning*, pages 872–879. ACM, 2008.

[18] Lawrence Saul, Tommi Jaakkola, and Michael Jordan. Mean field theory for sigmoid belief networks. *Journal of Artificial Intelligence Research*, 4(61):76, 1996.

[19] Aaron van den Oord and Benjamin Schrauwen. Factoring variations in natural images with deep Gaussian mixture models. In *Advances in Neural Information Processing Systems*, pages 3518–3526, 2014.