ITERATIVE REFINEMENT OF APPROXIMATE POSTERIOR FOR TRAINING DIRECTED BELIEF NETWORKS

R. Devon Hjelm
Mind Research Network
and the Department of Computer Science
University of New Mexico
dhjelm@mrn.org

Nebojsa Jojic
Mictrosoft Resarch
jojic@microsoft.com

Kyunghyun Cho
Courant Institute of Mathematical Sciences
and Center for Data Science
New York University
kyunghyun.cho@nyu.edu

Junyoung Chung
University of Montreal
elecegg@gmail.com

Ruslan Salakhutdinov
University of Toronto
rsalakhu@cs.toronto.edu

Vince Calhoun
Mind Research Network
and the Department of Electrical
and Computer Engineering
University of New Mexico
vcalhoun@mrn.org

ABSTRACT

Deep directed graphical models, while representing a potentially powerful class of generative representations, are challenging to train due to difficult inference. Recent advances in variational inference that make use of a inference or *recognition network* have advanced well beyond traditional variational inference and MCMC methods. While these techniques offer higher flexibility as well as simpler and faster inference, they are still limited by the choice of approximate posterior and require variance reduction. Less focus has been given to improving or *refining* the approximate posterior beyond those provided by variational inference. We show that iterative refinement of the approximate posterior can provide notable gains in maximizing the lower bound of the log likelihood, either by applying gradient descent or by using adaptive importance sampling during the E-step of a variational EM algorithm. We show our approach competes with state of the art in both continuous and binary latent variables.

1 INTRODUCTION

Deep generative models offer the capacity for rich representations of complex data while as they can capture multimodal properties of the distribution of interest and generalize better. Deep directed graphical models in particular have some advantage over their undirected cousins such as DBMs (Salakhutdinov & Hinton, 2009) as they do not suffer from learning and evaluation difficulties arising from an intractable partition function. Despite their representational power, directed graphical models are generally difficult to train, as the true posterior is generally intractable.

Methods for training variants of the Helmholtz machine (Dayan et al., 1995), such as wake-sleep (Hinton et al., 1995; Bornschein & Bengio, 2014) and neural variational inference and learning (NVIL) (Mnih & Gregor, 2014), and methods for training variational autoencoders (VAE) (Kingma & Welling, 2013) have supplanted more traditional Markov Chain Monte Carlo (MCMC) or variational inference based methods such as mean field coordinate ascent inference in training deep generative models. The gains offered by these methods are due to parameterizing the approximate posterior with a *recognition network* on top of clever learning algorithms. As opposed to traditional

methods, this general approach provides a fast and effective means of performing inference and learning which scales well to large datasets.

While advances have been clear, these methods are still limited by the choice of approximate posterior, which translates to a looseness in the lower bound. Some recent work has begun to address this limitation by either relaxing the factorial assumption (Burda et al., 2015) on the approximate posterior or through the introduction of auxiliary inference networks (Rezende & Mohamed, 2015).

I need to work on this Notably, there has been recent work focusing on refining the variational posterior by applying gradient descent to variational inference (Hoffman et al., 2013) or by applying additional MCMC transitions (Salimans, 2014). We apply two different techniques to refine the approximate posterior from the recognition network during the E-step of an EM algorithm. We first show that, given lower bound optimization can be reinterpreted in terms of a deterministic function, we can use gradient descent to refine the approximate posterior during inference. Next, we show that with binary latent variables, adaptive importance sampling (AdIS) (Karamchandani et al., 1989) can be used iteratively to achieve a tighter lower bound to the log likelihood.

2 Directed Belief Networks and Variational Inference

The term *belief network* describes a class of generative models of which the observed distribution is conditioned on a set of stochastic latent causes: $p(x) = \sum_h p(x|h)p(h)$, though in general this term is used to include models where the observed variables are not conditionally independent on the latent variables, h.

In this work, we will use the name *directed belief network* to restrict ourselves to a generative directed graphical model such that the likelihood function factorizes into a hierarchy of conditional densities: $p(x|h) = p(h_n)p(x|h_1)\prod_{l=1}^{n} p(h_l|h_{l+1})$, where each layer l is conditionally independent on the layer above with density $p(h_l|h_{l+1})$ and $p(h_n)$ is a prior distribution of the top layer. Examples include sigmoid belief networks (SBN, Neal, 1992) and deep belief networks (DBN, Hinton et al., 2006).

Directed belief networks can be trained by maximizing log-likelihood (MLE), but this is difficult due to the necessary marginalization over the latent density. If we had the posterior $p(h|x) = p(x,h)/p(x)$, then learning would be easy, but in general this is not possible and we require some more advanced techniques.

3 Variational Inference and Learning of Directed Belief Networks

3.1 Variational Lowerbound of Directed Belief Network

Even with reasonably complex likelihood functions, MLE is not tractable in directed belief networks due to the intractable posterior inference. One solution is to use Markov Chain Monte Carlo (MCMC) sampling to iteratively approximate the posterior distribution over the latent variables, which is exact given an unbound number of steps under certain conditions. This typically suffers high variance.

Another popular method, variational inference introduces an approximate posterior $q_{\boldsymbol{\theta}}(h|x)$:

$$\begin{aligned}\log p(x) &= \log \sum_h p(x,h) = \log \sum_h q(h|x)\frac{p(x,h)}{q(h|x)} \\ &\geq \sum_h q(h|x)\log\frac{p(x,h)}{q(h|x)} = \sum_h q(h|x)\log p(x,h) + \mathcal{H}(q) = \mathcal{L}, \end{aligned} \tag{1}$$

with variational parameters $\boldsymbol{\theta}$, where $\mathcal{H}(q)$ is the entropy of the approximate posterior q.

This introduces *lower bound*, $\mathcal{L}$, of the exact likelihood. Variational inference rephrases learning as choosing the variational parameters $\boldsymbol{\theta}$ to maximize this bound, rather than choosing the latent variables directly. The bound is tight (i.e., $\mathcal{L} = \log p(x)$) when the KL divergence between the approximate and true posterior is 0 (i.e., $D_{KL}(q_{\boldsymbol{\theta}}(h|x)||p(h|x)) = 0$,) or equivalently when the

approximate matches the true posterior. Optimization then is dependent on the choice of a parametric form for the posterior that can best approximate the exact posterior, and models generally vary in choice of parameterization followed by a learning algorithm which efficiently maximizes its lower bound.

3.2 Training a Directed Belief Network

If we have a method for finding an approximate posterior, then an expectation-maximization (EM) algorithm (Dempster et al., 1977; Neal & Hinton, 1998). In the Expectation (E) step, the variational parameters parameters of the approximate posterior, $\boldsymbol{\theta}$, are chosen/updated such that the KL-divergence between the approximate posterior and true posterior is minimized. In the Maximization (M) step, the model parameters $\boldsymbol{\phi}$ are chosen/updated to maximize the likelihood function.

In general, the gradient w.r.t. the model parameters of the lower bound can be estimated using the following Monte Carlo approximation:

$$\nabla_{\boldsymbol{\phi}} \mathcal{L} \approx \frac{1}{N} \sum_{n} \nabla_{\boldsymbol{\phi}} \log p_{\boldsymbol{\phi}}(x, h^{(n)}), \tag{2}$$

where $h^{(n)} \sim q(h|x)$.

3.3 Recognition Network

Recently, it has been found that the approximate posterior inference can be done quickly by using a feed forward, often deep, *recognition network* composed of *global* variational parameters to predict the local variational parameters from the data (see, e.g., Kingma & Welling, 2013; Mnih & Gregor, 2014; Rezende et al., 2014). However, these approaches often require additional tricks to train, as the gradient of the lower bound w.r.t. the global variational parameters is difficult to compute, and most of the off-the-shelf approximations, such as Monte Carlo approximation, have high variance.

Variational autoencoders (VAE) (Kingma & Welling, 2013) make use of a clever parameterization trick so that the lower bound can be rephrased as a deterministic function with some auxilliary noise. The wake-sleep algorithm uses samples from the joint density $p(x, h)$ to optimize the variational parameters, effectively making the algorithm an optimization over two cost functions (Hinton et al., 1995). NVIL uses REINFORCE along with a baseline predicted from a deep neural network to make lower variance MCMC estimates of the gradient.

It can be argued that better estimates can be made by simply having a better posterior. Some work has proposed to use gradient descent (Hoffman et al., 2013), auxiliary networks (Rezende & Mohamed, 2015), or MCMC transitions (Salimans, 2014) to refine inference with various posterior parameterizations.

In general, we want an iterative refinement procedure, defining a transition operator T that arrives at a better posterior, $\tilde{q}$, when applied iteratively to approximate posterior, q.

4 Iterative Refinement for Variational Inference

4.1 Iterative Refinement of Approximate Posterior

We restrict ourselves to the case where the prior and approximate posterior are factorial, that is: $p(h) = \prod_i p(h_i)$ and $\tilde{q}(h|x; \boldsymbol{\theta}) = \prod_i \tilde{q}(h_i|x; \boldsymbol{\theta})$, respectively. In order to augment our inference from those available from the parameterized approximate posterior, we wish to achieve a better posterior by taking a set number of iterative steps in the *local* variational parameters $\boldsymbol{\theta}^l$ from variation inference by applying a series of transition operators $T(\boldsymbol{\theta}^l_t, \boldsymbol{\theta}^l_{t-1}; \mathcal{L})$ parameterized by $\mathcal{L}$. These transition operators should be a function of the lower bound and maximize it as well.

After we find $\tilde{q}$, we maximize the lower bound w.r.t. the model parameters using stochastic gradient descent, using samples from the refined approximate posterior. Our loss w.r.t. the global variational parameters becomes $D_{KL}(\tilde{q}(\theta^l)||q_{\boldsymbol{\theta}})$.

4.1.1 GRADIENT DESCENT ITERATIVE REFINEMENT (GDIR)

If we can rephrase the lower bound as a deterministic function g of the local variational parameters and some optional auxiliary noise variables, ϵ, then we can improve the local variational parameters by applying iterative steps of gradient ascent:

$$\begin{aligned} \nabla_{\boldsymbol{\theta}^l} \mathcal{L} &\approx \nabla_{\boldsymbol{\theta}^l} g(\boldsymbol{\theta}^l, \epsilon) \\ \boldsymbol{\theta}^l_t &= \boldsymbol{\theta}^l_{t-1} + \gamma \nabla_{\boldsymbol{\theta}^l} g(\boldsymbol{\theta}^l, \epsilon) \end{aligned} \tag{3}$$

where γ is the "inference rate" and $\boldsymbol{\theta}^l_0$ are the initial local parameters found through the initial variational inference. The success of this approach will depend on the choice of approximation, which we will show.

Gaussian Latent Variables Let us assume that our approximate posterior and prior distributions are parameterized by multivariate normal distributions with diagonal covariance. In order to pass the gradient of the approximate log conditional to the variational parameters, we can use the re-parameterize our approximate posterior as in VAE:

$$\begin{aligned} h(\boldsymbol{\theta}^l) &= \mu + \epsilon \odot \sigma \\ \epsilon &\sim \mathcal{N}(0, 1), \end{aligned} \tag{4}$$

where $(\mu, \sigma) = \boldsymbol{\theta}^l$ are the mean and variance parameters for the posterior density and our auxiliary noise variables, ϵ, are sampled from a unit-variance, zero-mean normal distribution.

The gradient of the lower bound w.r.t. the local variational parameters can be approximated easily by Monte Carlo method:

$$\nabla_{\boldsymbol{\theta}^l} g(\boldsymbol{\theta}^l, \epsilon) = \frac{1}{N} \nabla_{\boldsymbol{\theta}^l} \left(\sum_n \log p(x, h(\boldsymbol{\theta}^l)^{(n)}) - \log \tilde{q}(h(\boldsymbol{\theta}^l)^{(n)}) \right) \tag{5}$$

Bernoulli Latent Variables Unfortunately, a suitable re-parameterization as above is not available to us with binary latent variables. In order to use gradient descent to back-propagate the gradients of the lower bound, we need to make a stronger approximation.

There has been some recent work into passing gradients through binary latent variables (Bengio et al., 2013). Instead of approximating the partial derivatives w.r.t. the latent variables in back-propagation, we instead push the centers of the Bernoulli distribution, avoiding sampling overall. Our lower bound becomes:

$$\mathcal{L} \approx g(\boldsymbol{\theta}^l) = \log p(x|\mu) + \langle \log p(h) \rangle_{\tilde{q}} + \mathcal{H}(\tilde{q}), \tag{6}$$

where $\log p(x|\mu)$ is the log-output of the generation network using the variational parameters $\mu = \text{sigmoid}(z)$ as input. However, this approximation is likely highly biased, and becomes worse as μ moves away from $(0, 1)$, so we would expect this approximation to be best when the entropy is low.

4.1.2 ADAPTIVE IMPORTANCE SAMPLING ITERATIVE REFINEMENT (ADSIR)

As an alternative and to address the issues with binary latent variables, we use *adaptive importance sampling* (Karamchandani et al., 1989) to iteratively refine the Bernoulli centers at each inference step:

$$\begin{aligned} h^{(n)} &\sim \text{Bernoulli}(\mu_t) \\ w^{(n)} &= p(x, h^{(n)}) / q(h^{(n)}|x) \\ \tilde{w}^{(n)} &= w^{(n)} / \sum_n w^{(n)} \\ \mu_{t+1} &= \sum_n \tilde{w}^{(n)} h^{(n)} \end{aligned} \tag{7}$$

This will clearly lead to a biased estimate of the lower bound. But we will show that it works very well in practice.

5 RELATED WORKS

This needs work Our work combines elements found in variational inference and MCMC methods. In spirit, it is closest to the hybrid MCMC for variational inference (Salimans, 2014), but uses a much MCMC transition than the hamiltonian MC used there, while the AdIS method is also applicable to binary units.

Recent work on stochastic variational inference Hoffman et al. (2013) also uses gradient descent to improve on the variational inference algorithm. However, the exact parameterization of the posterior is very limited and the method cannot be used with complex approximate posteriors.

Normalizing flows for VAE (Rezende & Mohamed, 2015) make use of auxiliary networks to improve the approximate posterior. In doing so, they are able to escape the factorial condition that GDIR and AdISIR have. But they require additional parameters to improve the posterior.

6 EXPERIMENTS

6.1 SETTINGS

We test GDIR and AdISIR on MNIST, using the binarized dataset found in (Salakhutdinov & Murray, 2008), with a standard train, validation, and test split of 50k, 10k, and 10k samples. Unfortunately, recent papers on VAE (Mnih & Gregor, 2014; Salimans, 2014) have used different, non-standard splits, or actively sample from continuous MNIST (Burda et al., 2015), making it difficult to compete. We however make sure that our results are from the most conservative setting (having the least amount of training examples), thereby making any comparison as meaningful as possible.

All models used a recognition network, for comparison to similar models, with the number of parameters and deterministic layers (if present) to match the generation network.

All models were trained with early stopping by recording best validation, and none of our models used any of the regularizations available to us, such as dropout, weight noise, or L2-norm regularization. All of our models were trained for at most 500 epochs, and each took no more than 24 hours to complete, best validation was typically reached much faster.

6.2 CONTINUOUS VARIABLES

For the continuous latent variable case we used the same network structure as in (Kingma & Welling, 2013; Salimans, 2014), with 200 continuous latent variables and a generation and recognition network with 500 deterministic softplus units. Each of our models were trained using 20 samples during inference using GDIR to refine output of the recognition network and 20 samples to approximate the lower bound during the M-step. Optimal parameters were found using the rmsprop algorithm (Hinton, 2012) with an initial learning rate of 1×10^{-4} with a batch size of 100. We are currently running validation exps to pick new hps

Our results with continuous variables are presented in table 6.2. For comparison are (approximate) VAE results They just have plots, no numbers from (Kingma & Welling, 2013), normalizing flow (Rezende & Mohamed, 2015), Hamiltonian MCMC variational inference (Salimans, 2014) with the number of leapfrog steps specified.

We observe faster convergence and better lower bound when using more gradient steps. I need to add these tables

GDIR shows notable improvement over VAE, while showing comparable performance to Hamiltonian MCMC, despite conservative training. Need to read more on normalizing flow, but it's unclear the parameterization they use, number of latent layers, and whether they are reporting NLL or lower bound. In addition they have more parameters in their auxiliary inference network, so this needs to be addressed.

Model	lower bound	NLL
VAE	99	
$GBVI_{20}$	92.83	90.51
HVI_5[1]	90.86	87.16
HVI_{10}[1]	87.60	85.56
$DLGM + NF$	TODO	results in paper are confusing

Table 1: Lower bounds and NLL for various continuous latent variable models and training algorithms.

Model	lower bound	NLL
Wake-sleep	120.7	116.3
NVIL	113.1	
AVI_{20}	114.08	110.13
$AVI_{20}(240)$	107.03	103.10
$MPVI_{20}$		
$MPVI_{20}(240)$	122.67	103.32

Table 2: Test lower bounds and NLL for various Bernoulli latent variable models training algorithms.

6.3 Bernoulli Variables

For binary latent variables, we used two structures to compare to wake-sleep and NVIL. For comparison to wake-sleep, we used a shallow feed forward network with 200 output neurons for the binary latent variables for the posterior and no deterministic units for the posterior nor conditional. As NVIL uses an additional feed forward network for baseline prediction, we used an deterministic layer of 240 softplus units in the recognition and generation networks to roughly match the total number of parameters. Parameters were adjusted using Adam (Kingma & Adam, 2014) with an initial learning rate of 3×10^{-4} with a batch size of 10 as per validation. Much of these hps will change this week, most likely to match continuous above

We are clearly beating NVIL and wake-sleep. Though with no extra deterministic layers we are also very close. The story with number of steps is not as strong as above. We can reach the same lower bound with only one step. But it's MUCH faster to use more steps. I will work on these figures. In addition, I am working on code / exps with 2 layers of latent variables: I have a feeling the difference there will be more pronounced.

7 Conclusion

TBD

Acknowledgments

Microsoft, Mind Research Grants, PIBBS maybe

TODO

References

Bengio, Yoshua, Léonard, Nicholas, and Courville, Aaron. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Bornschein, Jörg and Bengio, Yoshua. Reweighted wake-sleep. *arXiv preprint arXiv:1406.2751*, 2014.

Burda, Yuri, Grosse, Roger, and Salakhutdinov, Ruslan. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

Dayan, Peter, Hinton, Geoffrey E, Neal, Radford M, and Zemel, Richard S. The helmholtz machine. *Neural computation*, 7(5):889–904, 1995.

Dempster, Arthur P, Laird, Nan M, and Rubin, Donald B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

Hinton, Geoffrey. Neural networks for machine learning. Coursera, video lectures, 2012.

Hinton, Geoffrey, Dayan, Peter, Frey, Brendan, and Neal, Radford. The "wake-sleep" algorithm for unsupervised neural networks. *Science*, 268(5214):1158–1161, 1995.

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

Hoffman, Matthew, Blei, David, Wang, Chong, and Paisley, John. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

Karamchandani, A, Bjerager, P, and Cornell, CA. Adaptive importance sampling. In *Structural Safety and Reliability*, pp. 855–862. ASCE, 1989.

Kingma, Diederik and Welling, Max. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kingma, DP and Adam, J Ba. A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Mnih, Andriy and Gregor, Karol. Neural variational inference and learning in belief networks. *arXiv preprint arXiv:1402.0030*, 2014.

Neal, Radford M. Connectionist learning of belief networks. *Artificial intelligence*, 56(1):71–113, 1992.

Neal, Radford M and Hinton, Geoffrey E. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pp. 355–368. Springer, 1998.

Rezende, Danilo Jimenez and Mohamed, Shakir. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.

Rezende, Danilo Jimenez, Mohamed, Shakir, and Wierstra, Daan. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

Salakhutdinov, Ruslan and Hinton, Geoffrey E. Deep boltzmann machines. In *International Conference on Artificial Intelligence and Statistics*, pp. 448–455, 2009.

Salakhutdinov, Ruslan and Murray, Iain. On the quantitative analysis of deep belief networks. In *Proceedings of the 25th international conference on Machine learning*, pp. 872–879. ACM, 2008.

Salimans, Tim. Markov chain monte carlo and variational inference: Bridging the gap. *arXiv preprint arXiv:1410.6460*, 2014.