Iterative Refinement of the Approximate Posterior for Training Directed Belief Networks

R Devon Hjelm
dhjelm@mrn.org

Kyunghyun Cho
kyunghyun.cho@nyu.edu

Junyoung Chung
junyoung.chung@umontreal.ca

Russ Salakhutdinov
rsalakhu@cs.toronto.edu

Vince Calhoun
vcalhoun@mrn.org

Nebojsa Jojic
jojic@microsoft.com

March 15, 2016

Abstract

Recent advances in variational inference that make use of an inference or *recognition network* for training and evaluating deep directed graphical models have advanced well beyond traditional variational inference and Markov chain Monte Carlo methods. These techniques offer higher flexibility with simpler and faster inference; yet training and evaluation still remains a challenge. We propose a method for improving the per-example approximate posterior by iterative refinement, which can provide notable gains in maximizing the variational lower bound of the log likelihood and works with both continuous and discrete latent variables. We evaluate our approach as a method of training and evaluating directed graphical models. We show that, when used for training, iterative refinement improves the variational lower bound and can also improve the log-likelihood over related methods. We also show that iterative refinement can be used to get a better estimate of the log-likelihood in any directed model trained with mean-field inference.

1 Introduction

Deep generative models have great representational capacity for modelling complex data, as they are multimodal and generalize better than models based only on deterministic processes. Deep directed models, in addition to being good models for complex and multimodal density, can be used to extract hierarchal representations. Compared to undirected graphical models such as deep Boltzmann machines (DBMs, Salakhutdinov & Hinton, 2009), directed models are easier to draw samples from and do not suffer from an intractable partition function.

Despite some advantages, directed graphical models are still difficult to train and evaluate, 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 the variational autoencoder variant (VAE, Kingma & Welling, 2013) have surpassed traditional methods in training deep generative models, such as Markov chain Monte Carlo (MCMC) (Neal, 1992) and mean-field coordinate ascent (Saul et al., 1996). These methods succeed because they use a *recognition network* for inference, allowing for fast inference and providing a means for learning which scales well to large datasets.

However, training a recognition network can be challenging, particularly when the latent variables are discrete. The most successful of these methods, VAE, relies on a re-parameterization of the latent variables to pass the learning signal to the recognition network, but this type of parameterization is not available with discrete units. Estimates for passing the gradients with discrete variables are all biased (Bengio et al., 2013), and in general offer a learning signal insufficient to train a complex recognition network.

In addition, calculating the exact log-likelihood of directed models can be computationally infeasible, as it requires a marginalization over the entire latent space. The approximate posterior offered by a recognition network can be used to estimate the likelihood, but this estimate can have high variance.

We demonstrate training and evaluating directed generative models by improving the approximate posterior by iterative refinement. The complete learning algorithm follows expectation-maximization (EM, Dempster et al., 1977;

Neal & Hinton, 1998)), where in the E-step the variational parameters of the approximate posterior are initialized using the recognition network then iteratively refined. The refinement procedure provides an asymptotically-unbiased estimate of the variational lower bound, which is tight with respect to the true posterior, works with discrete and continuous latent variables, and can be used to easily train both the recognition network and generative model during the M-step. The E-step can improve the approximate posterior of any directed model trained with mean-field inference and can give a much better lower bound and a more accurate estimate of the log-likelihood.

For our iterative refinement step, we use adaptive importance sampling (AdIS Oh & Berger, 1992), which works with binary latent variables where no good unbiased solutions exist. We demonstrate the proposed refinement procedure is effective for training directed generative models, improving the lower bound over state-of-the-art methods for a given configuration and often providing a better log-likelihood. We also demonstrate the improved posterior from refinement can greatly improve inference and the accuracy of evaluation.

2 Directed Belief Networks and Variational Inference

For the purposes of this work, a *directed belief network* is a generative directed graphical model consisting of a conditional density $p(\mathbf{x}|\mathbf{h})$ and a prior $p(\mathbf{h})$, such that the joint density can be expressed as $p(\mathbf{x}, \mathbf{h}) = p(\mathbf{x}|\mathbf{h})p(\mathbf{h})$. In particular, the conditional density factorizes into a hierarchy of conditional densities: $p(\mathbf{x}|\mathbf{h}) = p(\mathbf{x}|\mathbf{h}_1)\prod_{l=1}^{L-1} p(\mathbf{h}_l|\mathbf{h}_{l+1})$, where each layer l is conditionally independent on the layer above with density $p(\mathbf{h}_l|\mathbf{h}_{l+1})$ and $p(\mathbf{h}_L)$ is a prior distribution of the top layer. Sampling from the model can be done simply via ancestral-sampling, first sampling from the prior, then subsequently sampling from each layer until reaching the observation, $\mathbf{x}$. This latent variable structure can improve model capacity, but inference can still be intractable, as is the case in deep sigmoid belief networks (SBN, Neal, 1992), deep belief networks (DBN, Hinton et al., 2006), and other models in which each of the conditional distributions involves complex nonlinear functions.

Directed belief networks can be trained through maximizing likelihood estimation (MLE), but this is difficult due to the necessary marginalization of the joint density over the latent variables. Learning would be straightforward given the exact posterior, $p(\mathbf{h}|\mathbf{x}) = p(\mathbf{x}, \mathbf{h})/p(\mathbf{x})$, but in general this is not tractable. The posterior distribution can be iteratively approximated using MCMC sampling, which can be exact given an unbounded number of steps under certain conditions, but which is not practical in large-scale applications due to slow mixing and high computational costs.

2.1 Variational Lowerbound of Directed Belief Network

Another method of approximate inference, *variational inference*, introduces an approximate posterior, $q(\mathbf{h}|\mathbf{x}; \boldsymbol{\theta})$, with variational parameters, $\boldsymbol{\theta}$, such that the log-likelihood is bounded from below:

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

where $\mathcal{H}(q)$ is the entropy of the approximate posterior. Rather than directly maximizing the exact log-likelihood, variational inference rephrases inference as choosing the variational parameters, $\boldsymbol{\theta}$, to maximize the *lower bound*, $\mathcal{L}$. The bound is tight (e.g., $\mathcal{L} = \log p(\mathbf{x})$) when the KL divergence between the approximate and true posterior is zero (e.g., $D_{KL}(q(\mathbf{h}|\mathbf{x}; \boldsymbol{\theta})||p(\mathbf{h}|\mathbf{x})) = 0$), in other words when the approximate matches the true posterior. When the bound is tight, maximizing the lower bound is equivalent to maximizing the exact likelihood, and the looser the bound, the worse the approximation.

2.2 Training a Directed Belief Network

Given a method for inferring the approximate posterior, $q(\mathbf{h}|\mathbf{x}; \boldsymbol{\theta})$, learning is done by expectation-maximization (EM, Dempster et al., 1977; Neal & Hinton, 1998)). In the expectation (E) step, the variational 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 of the lower bound w.r.t. the model parameters can be estimated using the a Monte Carlo approximation:

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

where $\mathbf{h}^{(n)} \sim q(\mathbf{h}|\mathbf{x})$ is the n-th indepedent sample drawn from $q(\mathbf{h}|\mathbf{x})$.

Most of the difficulty in training directed networks lies in inferring the approximate posterior. It is often convenient to assume that the posterior distribution factorizes ($q(\mathbf{h}|\mathbf{x}) = \prod_i q(h_i|\mathbf{x})$), an approach known as *mean-field inference*. However, mean-field variational inference for directed networks is non-trivial, as the exact mean-field equations often do not have a closed form. Coordinate ascent mean field inference is expensive, though in limited settings the natural gradient can be used (Honkela et al., 2010; Hoffman et al., 2013). For sigmoid belief networks (SBNs), introducing additional variational parameters can help (Saul et al., 1996) but has very limited success in practice.

2.3 Evaluating a Directed Belief Network

While the lower bound can be used to evaluate directed models, an estimate of the log-likelihood is more informative on the model's generative capabilities. The approximate posterior can be used in an estimator for the log-likelihood:

$$\log p(\mathbf{x}) \approx \log \sum_{n=1}^{N} \frac{p(\mathbf{x}, \mathbf{h}^{(n)})}{q(\mathbf{h}^{(n)}|\mathbf{x})} - \log N, \tag{3}$$

where the estimator makes use of N independent samples, $\mathbf{h}^{(n)} \sim q(\mathbf{h}|x)$, drawn from the approximate posterior. This is a biased estimator which will on average underestimate the true log-likelihood (Bornschein & Bengio, 2014). In order to make use of this estimator, we need to draw a large number of samples ($N \approx 100,000$ in the literature for models trained on MNIST), as it has high variance.

2.4 Recognition Network

The success of variational inference relies on the choice of approximate posterior, as poor choice can result in a looser lower bound, worse learning, and less accurate log-likelihood estimation. A deep feed forward *recognition network* parameterized by *global* variational parameters, $\boldsymbol{\psi}$, has become a popular choice, as it offers fast and flexible data-dependent inference (see, e.g., Salakhutdinov & Larochelle, 2010; Kingma & Welling, 2013; Mnih & Gregor, 2014; Rezende et al., 2014). The recognition networks maps the input observations, $\mathbf{x}$, to the parameters of approximate posterior, $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}(\mathbf{x}) = f(\mathbf{x}; \boldsymbol{\psi}). \tag{4}$$

It is very common to assume the approximate posterior factorizes, $q(\mathbf{h}|\mathbf{x}; \boldsymbol{\theta}) = \prod_i q(h_i|\mathbf{x}; \boldsymbol{\theta}_i)$, such that the recognition network output is an estimate of the mean-field parameters. Generally known as a "Helmholtz machine" (Dayan et al., 1995), these approaches often require additional tricks to train, as the gradients of the lower bound from the expected reconstruction loss, $\mathbb{E}_{q(\mathbf{h}|\mathbf{x})}[\log p(\mathbf{x}, \mathbf{h})]$, do not generally back-propagate naturally through the latent variables, and most off-the-shelf approximations have high variance (Mnih & Gregor, 2014).

For Helmhotz machines with continuous latent variables, exact back-propagation is possible via re-parameterization, which leads to variational autoencoders (VAE, Kingma & Welling, 2013). However, re-parameterizations of this sort are not available with discrete latent variables, and estimating the stochastic gradient (Bengio et al., 2013) leads to a biased estimate of the lower bound which has too high of a variance to be practical. Instead, inference with discrete variables relies on creative methodology (Hinton et al., 1995; Mnih & Gregor, 2014).

Learning difficulties aside, using a recognition network for the posterior distribution couples estimates for different data points through global variational parameters, which helps the model generalize better and avoid local-minima problems common with traditional variational inference. However, depending on the structure of the recognition

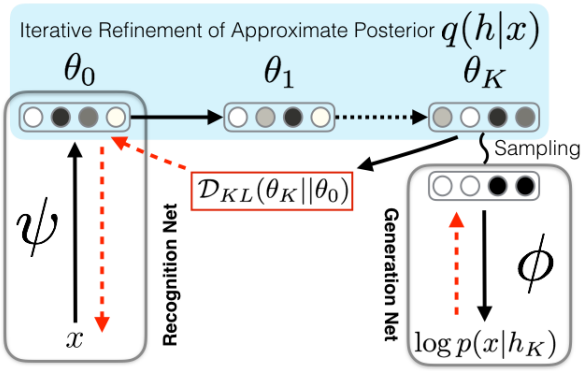

Figure 1: Iterative refinement for variational inference. An initial mean-field estimate of the approximate posterior is made through a recognition network. The mean-field parameters are then updated iteratively, maximizing the lower bound. The final approximate posterior is used to train the generative model by sampling. When IRVI is used for training, the recognition network parameters are updated using the KL divergence between the refined posterior q_k and the output of the recognition network q_0.

network, this can put restrictions on each posterior distribution, leading to poorer fit which can loosen the variational lower-bound. The posterior estimates may be inaccurate, so that the data can appear to fit the model worse than it really does. Therefore, during learning we need a sufficiently complex recognition network, the limits of which are not well understood. Our approach is to use the recognition network as a preliminary guess of the posterior, followed by iterative refinement through mean-field updates of the variational parameters. This approach, which can both speed up mean-field inference and improve on the lower bound from the approximate posterior provided by the recognition network, has been used to train deep Boltzmann machines (DBMs, Salakhutdinov & Larochelle, 2010). While straightforward in DBMs, maximizing the lower bound in directed networks with mean-field inference is challenging, and traditional mean-field requires updating each factor, θ_i, individually.

3 Iterative Refinement for Variational Inference (IRVI)

For the remainder of this paper, we will distinguish the generative parameters ϕ from the variational parameters of the recognition network, ψ, and the local parameters of the approximate posterior, $\boldsymbol{\theta}$.

3.1 Iterative Refinement of the Approximate Posterior

Iterative refinement for variational inference (IRVI) is the use of an efficient stochastic transition operator as an update for mean-field inference for maximizing the variational lower bound, using the recognition network as an initial guess of the posterior. In IRVI, we define a stochastic transition operator on the variational parameters which is a function of the observations, $\boldsymbol{\theta}_{k+1} = g(\boldsymbol{\theta}_k, \mathbf{x})$, for which the true posterior, $p(\mathbf{h}|\mathbf{x}; \boldsymbol{\theta}^\star)$, is a stationary state ($g(\boldsymbol{\theta}^\star, \mathbf{x}) = \boldsymbol{\theta}^\star$). In addition, we require the transition operator operate simultaneously on all factors of the approximate posterior: $q(\mathbf{h}|\mathbf{x}; g(\boldsymbol{\theta}, \mathbf{x})) = \prod_i q(h_i|\mathbf{x}; g_i(\boldsymbol{\theta}_i; \mathbf{x}))$.

An overview of IRVI is available in Figure 1. For the E-step, we feed the observation, $\mathbf{x}$, through the recognition network to get the initial parameters of the approximate posterior, $\boldsymbol{\theta}_0$, then apply K updates to the variational parameters, $\boldsymbol{\theta}_{k+1} = g(\boldsymbol{\theta}_k, \mathbf{x})$, iterating through K parameterizations of the approximate posterior, $q_0(\mathbf{h}|\mathbf{x}; \boldsymbol{\theta}_0), q_1(\mathbf{h}|\mathbf{x}; \boldsymbol{\theta}_1) \dots q_K(\mathbf{h}|\mathbf{x}; \boldsymbol{\theta}_K)$. Optionally, for training with the final set of parameters, $\boldsymbol{\theta}_K$, the gradients with respect to ϕ and ψ in the M-step become:

$$\nabla_\phi \mathcal{L} = \mathbb{E}_{q_K(\mathbf{h}|\mathbf{x})}[\nabla_\phi \log p(\mathbf{x}, \mathbf{h}; \phi)] \tag{5}$$

$$\nabla_\psi \mathcal{L} = -\nabla_\psi D_{KL}(q_K(\mathbf{h}|\mathbf{x})||q_0(\mathbf{h}|\mathbf{x}; \psi)). \tag{6}$$

Equation 6 is central to training with IRVI: we train the recognition network to *predict* the refined approximate posterior, improving the initialization for future inference and avoiding the need to back-propagate gradients through the

latent variables.

It is easy to demonstrate such a transition operator, $g(\boldsymbol{\theta}, \mathbf{x})$, exists. With variational autoencoders (VAE), the back-propagated gradient of the lower bound with respect to the approximate posterior is composed of individual gradients for each factor, θ_i that can be applied simultaneously. Applying the gradient directly to the variational parameters, $\boldsymbol{\theta}$, without back-propagating to the recognition network parameters, $\boldsymbol{\psi}$, yields a simple iterative refinement operator:

$$\boldsymbol{\theta}_{t+1} = g(\boldsymbol{\theta}_t, \mathbf{x}, \gamma) = \boldsymbol{\theta}_t + \gamma \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \mathbf{x}, \boldsymbol{\epsilon}), \tag{7}$$

where γ is the inference rate hyperparameter and $\boldsymbol{\epsilon}$ is auxiliary noise used in the re-parameterization.

This gradient-descent iterative refinement (GDIR) is very straightforward with continuous latent variables as with VAE. However, GDIR with discrete units suffers the same shortcomings as when passing the gradients directly, so a better transition operator is needed.

3.2 Adaptive Importance Sampling Iterative Refinement (AIR)

Adaptive importance sampling (AdIS, Oh & Berger, 1992) provides a general approach to iteratively refining the local variational parameters. Adaptive importance sampling "recenters" the proposed distribution using a weighted average of M independently drawn samples from the approximate posterior, $\mathbf{h}^{(m)} \sim q(\mathbf{h}|\mathbf{x})$, each sample weighted by an importance weight:

$$\begin{aligned} w^{(m)} &= p(\mathbf{x}, \mathbf{h}^{(m)}) / q(\mathbf{h}^{(m)}|\mathbf{x}) \\ \tilde{w}^{(m)} &= w^{(m)} / \sum_n w^{(m)}, \end{aligned} \tag{8}$$

where $w^{(m)}$ and $\tilde{w}^{(m)}$ are the unnormalized and normalized weights for the m-th sample.

For binary stochastic variables with the Bernoulli centers, $\boldsymbol{\mu} = \boldsymbol{\theta}$, for each update or *inference step*, k, we draw M samples from the Bernoulli distribution, then move the centers using the importance weights:

$$\begin{aligned} \mathbf{h}^{(m)} &\sim \text{Bernoulli}(\boldsymbol{\mu}_k) \\ \boldsymbol{\mu}_{k+1} = g(\boldsymbol{\mu}_k, \mathbf{x}, \gamma) &= (1-\gamma)\boldsymbol{\mu}_k + \gamma \sum_{m=1}^{M} \tilde{w}^{(m)} \mathbf{h}^{(m)}, \end{aligned} \tag{9}$$

where γ is the inference rate or $(1-\gamma)$ can be thought of as the adaptive "damping" rate. As required for IRVI, the true posterior is a stationary state, as the unnormalized importance weights become $p(\mathbf{x}, \mathbf{h}^{(m)})/p(\mathbf{h}^{(m)}|\mathbf{x}) = p(\mathbf{x})$, making the weights uniform across samples, so that $g(\boldsymbol{\mu}_k, \mathbf{x}) = (1-\gamma)\boldsymbol{\mu}_k + \gamma\boldsymbol{\mu}_k = \boldsymbol{\mu}_k$. This is an asymptotically unbiased estimate of the lower bound that works very well in practice with a finite number of samples. This approach should work with any discrete parametric distribution, and although AIR should be applicable with continuous Gaussian variables, which have both mean and covariance, we were able to make reasonable gains using gradient-descent iterative refinement (GDIR), leaving applying AIR to continuous latent variables for future work.

3.3 Algorithm and Complexity

The general IRVI algorithm follows Algorithm 1. While iterative refinement may reduce the variance of stochastic gradient estimates and speed up learning, it comes at a computational cost, as each update is K times more expensive than fixed approximations. However, in addition to potential learning benefits, IRVI can also improve the approximate posterior of an already trained Helmholtz machine (VAE or otherwise) at test, independent on how the model was trained. Our implementation is available at https://github.com/rdevon/IRVI.

4 Related Work

Iterative refinement for variational inference (IRVI) is a hybrid method, combining concepts from variational inference and MCMC. In spirit, it is closest to the refinement procedure of hybrid MCMC for variational inference (HVI, Salimans et al., 2015) and normalizing flows for VAE (NF, Rezende & Mohamed, 2015). HVI uses Hamiltonian MCMC

Algorithm 1 IRVI

Require: A generative model $p(\mathbf{x}, \mathbf{h}) = p(\mathbf{x}|\mathbf{h})p(\mathbf{h})$ and a recognition network $\boldsymbol{\theta}_0 = f(\mathbf{x}; \boldsymbol{\psi})$
Require: Number of iterations, K
Require: (For AIR) Number of adaptive samples, M
Require: transition operator $g(\boldsymbol{\theta}, \mathbf{x}, \gamma)$ and inference rate γ.
(E-step for training or test)
Compute $\boldsymbol{\theta}_0 = f(\mathbf{x}; \boldsymbol{\psi})$ for $q_0(\mathbf{h}|\mathbf{x}; \boldsymbol{\theta}_0)$
for k=1:K **do**
(For AIR) Draw M samples from $q_k(\mathbf{h}|\mathbf{x})$ and compute normalized importance weights $\tilde{w}^{(m)}$
$\boldsymbol{\theta}_k = g(\boldsymbol{\theta}_{k-1}, \mathbf{x})$
end for
(M-step for training only)
$\Delta\boldsymbol{\phi} \propto \mathbb{E}_{q_K(\mathbf{h}|\mathbf{x})}[\nabla_{\boldsymbol{\phi}} \log p(\mathbf{x}, \mathbf{h}; \boldsymbol{\phi})]$
$\Delta\boldsymbol{\psi} \propto -\nabla_{\boldsymbol{\psi}} D_{KL}(q_K(\mathbf{h}|\mathbf{x})||q_0(\mathbf{h}|\mathbf{x}; \boldsymbol{\psi}))$

to extend the posterior along a directed graph corresponding to a periodic solution to a Hamiltonian with the use of auxiliary momentum variables. HVI is the same complexity as IRVI, as it requires computing the updated lower bound at every iteration. NF extends the posterior with invertible transformations with a Jacobian that is reasonably easy to compute. Despite the extra complexity involved with computation of the Jacobian and inverse, NF is typically cheaper than both IRVI and HVI, though a comparison to the added complexity of backpropagating through the parameterized refinement is difficult to quantify. While IRVI refines the posterior by adjusting the mean-field parameters, HVI and NF depart from the mean-field approximation in whole by adjusting the form of the approximate posterior, are fundamentally deeper latent models, and thus should provide superior solutions to IRVI when applied to mean-field. However, they come with two limitations compared to IRVI. First, they rely on the VAE re-parameterization to work, and thus cannot be applied to discrete variables. Second, their refinement cannot be applied to already-trained models to improve the posterior, as their refinements are parameterized and learned during training.

GDIR also shares similarities with stochastic variational inference (SVI, Hoffman et al., 2013), which also uses the natural gradient to improve on the variational inference algorithm. SVI, however, requires global latent variables with carefully chosen relationships with the local latent variables (our equivalent of $\mathbf{h}$) to do inference.

Neural variational inference and learning (NVIL, Mnih & Gregor, 2014) and re-weighted wake-sleep (RWS, Bornschein & Bengio, 2014), like adaptive importance sampling iterative refinement (AIR), offer arguably the best solution for inference and learning in directed belief networks with discrete variables, though NVIL is biased, and RWS (like AIR) is asymptotically so. While NVIL generally works, the variance is still not low enough to be practical, and convergence takes *much* longer in terms of epochs and wall clock time than AIR, despite lower complexity. RWS is very successful at training SBNs, but its complex objective function means that convergence properties cannot be proven.

AIR also shares similarities with RWS in the use of importance sampling, as with importance-weighted autoencoders (IWAE, Burda et al., 2015) and recent work on stochastic feed-forward networks (SFFN, Tang & Salakhutdinov, 2013; Raiko et al., 2014). Iterative refinement, however, takes a distinctly orthogonal approach, using importance sampling to achieve a better posterior through refinement. While the end result may be a reduction of variance during training, none of these methods can refine the posterior further at test.

Finally, IRVI is meant to be a general approach to improving the approximate posterior within the limits of mean-field inference. It can be used to improve the inference of many of the models above and improve the accuracy of evaluation.

5 Experiments

5.1 Settings

We evaluate iterative refinement for variational inference (IRVI), using adaptive importance sampling iterative refinement (AIR) as the mean-field update, for both training and evaluating directed belief networks. We train and test on the following benchmarks: the binarized MNIST handwritten digit dataset Salakhutdinov & Murray (2008) with a standard train, validation, and test split of 50k/10k/10k and the Caltech-101 Silhouettes dataset with a split

of $4100/2264/2307$. We centered the MNIST and Caltech datasets by subtracting the mean-image over the training set when used as input to the recognition network. We also train additional models using our implementation of the re-weighted wake-sleep algorithm (RWS, Bornschein & Bengio, 2014), the state-of-the-art for many configurations of directed belief networks with discrete variables on these datasets for comparison and to demonstrate improving the approximate posteriors with refinement. With our experiments, we show that 1) iterative refinement can train a variety of directed models as well or better than existing methods, 2) the gains from refinement improves the approximate posterior, and can be applied to models trained by other algorithms, and 3) iterative refinement can be used to improve a model with a relatively simple approximate posterior.

Models were trained using the RMSprop algorithm (Hinton, 2012) with a batch size of 100 and early stopping by recorded best variational lowerbound on the validation dataset. For AIR, 20 mean-field updates or "inference steps" ($K = 20$), 20 adaptive samples ($M = 20$), and an adaptive damping rate, γ, of 0.9 were used during inference, chosen from validation in initial experiments. 20 posterior samples ($N = 20$) were used for model parameter updates for both AIR and RWS. All models were trained for 500 epochs and were fine-tuned for an additional 500 epochs with a decaying learning rate and the SGD learning algorithm.

As inference and learning with deep directed models with discrete latent variables is the most difficult, we use a generative model composed of a) a factorized Bermoulli prior as with sigmoid belief networks (SBNs) or b) an autoregressive prior, as in published MNIST results with deep autoregressive networks (DARN, Gregor et al., 2013):

$$\begin{aligned} &\text{a)} \quad p(\mathbf{h}) = \prod_i p(h_i); P(h_i = 1) = \sigma(b_i) \\ &\text{b)} \quad P(h_i = 1) = \sigma(\sum_{j=0}^{i-1} (W_r^{i,j<i} h_{j<i}) + b_i), \end{aligned} \tag{10}$$

where σ is the sigmoid ($\sigma(x) = 1/(1 + \exp(-x))$) function, W_r is a lower-triangular square matrix, and $\mathbf{b}$ is the bias vector.

For our experiments, we use conditional and approximate posterior densities that follow Bernoulli distributions:

$$P(h_{i,l} = 1 | \mathbf{h}_{l+1}) = \sigma(W_l i, : \cdot \mathbf{h}_{l+1} + b_{i,l}), \tag{11}$$

where W_l is a weight matrix between the l and $l + 1$ layers. As in Gregor et al. (2013) when training on MNIST, we do not use autoregression on the observations, $\mathbf{x}$, and use a fully factorized approximate posterior.

5.2 Training and Density Estimation

We evaluate AIR for training SBNs with one, two, and three layers of 200 hidden units and DARN with 200 and 500 hidden units, comparing against our implementation of RWS. All models were tested using $100,000$ posterior samples to estimate the lower bounds and log-likelihoods, as is consistent with the literature (Gregor et al., 2013; Bornschein & Bengio, 2014). We evaluate AIR at test, calculating the lower bound and approximate log-likelihoods using the unrefined approximate posterior from the recognition network, then refining the approximate posterior using AIR with an inference rate of 0.01, with the K and M selected depending on the model configuration.

When training SBNs with AIR and RWS, we used a completely deterministic network for the approximate posterior. For example, for a 2-layer SBN, the approximate posterior factors into the approximate posteriors for the top and the bottom hidden layers, and the initial parameters for the top layer, $\boldsymbol{\theta}_0^{(2)}$ are a function of the initial parameters for the first layer, $\boldsymbol{\theta}_0^{(1)}$:

$$\begin{aligned} q_0(\mathbf{h}_1, \mathbf{h}_2 | \mathbf{x}; \boldsymbol{\theta}_0^{(1)}, \boldsymbol{\theta}_0^{(2)}) &= q_0(\mathbf{h}_1 | \mathbf{x}; \boldsymbol{\theta}_0^{(1)}) q(\mathbf{h}_2 | \mathbf{x}; \boldsymbol{\theta}_0^{(2)}) \\ \boldsymbol{\theta}_0^{(1)} = f_1(\mathbf{x}; \boldsymbol{\psi}_1) \quad \boldsymbol{\theta}_0^{(2)} &= f_2(\boldsymbol{\theta}_0^{(1)}; \boldsymbol{\psi}_2). \end{aligned} \tag{12}$$

For DARN, we trained two different configurations on MNIST: one with 500 stochastic units and an additional hyperbolic tangent deterministic layer with 500 units in both the generative and recognition networks, and another with 200 stochastic units with a 500 hyperbolic tangent deterministic layer in the generative network only. We only used the smaller 200 stochastic unit version of DARN when training on the Caltech-101 silhouettes dataset.

The results of our experiments with the MNIST and Caltech-101 silhouettes datasets trained with AIR and RWS are in Table 1, along with various other methods for training SBNs and other generative models for reference. For most

Table 1: Results for adaptive importance sampling iterative refinement (AIR) and reweighted wake-sleep (RWS) for a variety of model configurations. Additional sigmoid belief networks (SBNs) trained with neural variational inference and learning from †Mnih & Gregor (2014), as well as other generative models quoted from §Gregor et al. (2013) and ‡Bornschein & Bengio (2014). AIR is trained with 20 inference steps and adaptive samples ($K = 20, M = 20$) in training (*3 layer SBN was trained with 50 steps with a inference rate of 0.05). Values for negative lower bound and log-likelihood estimates for both the unrefined approximate posterior ($\times$) and the approximate posterior refined by AIR ($\surd$) are provided.

		MNIST				Caltech-101 Silhouettes			
	Model	$\leq -\log p(x)$		$\approx -\log p(x)$		$\leq -\log p(x)$		$\approx -\log p(x)$	
	Refined at test with AIR?	$\times$	$\surd$	$\times$	$\surd$	$\times$	$\surd$	$\times$	$\surd$
SBN	NVIL (200) †	113.1							
	NVIL (200-200) †	99.8							
	NVIL (200-200-200) †	96.7							
	RWS (200)	115.13	107.87	102.51	102.00	166.50	154.85	121.38	118.63
	RWS (200-200)	107.05	97.97	93.82	92.83	145.57	125.66	112.86	107.20
	RWS (200-200-200)	104.60	96.46	92.00	91.02	141.46	120.58	110.57	104.54
	AIR (200)	117.31	107.41	101.47	100.92	158.08	135.79	120.23	116.61
	AIR (200-200)	107.66	98.25	93.68	92.90	149.14	118.76	116.35	106.94
	AIR (200-200-200)	108.97	98.81	93.24	92.56*	140.34	115.30	112.11	104.36
DARN	RWS (200)	96.53	89.87	86.91	86.21	139.51	122.43	113.69	109.73
	RWS (500)	94.63	88.62	85.40	84.71				
	AIR (200)	95.83	88.64	86.43	85.89	140.45	118.91	115.36	109.76
	AIR (500)	93.92	88.37	85.91	85.46				
	DARN (500) §			84.11					
	RWS NADE / NADE ‡			85.23				104.3	
	DBN ‡	86.22		84.55					

model configurations, AIR and RWS perform comparably, though RWS appears to do better in log-likelihood estimates for some configurations. In addition, refinement with AIR at test greatly improves the log-likelihood estimates for both models, and the log-likelihood estimates become much more similar, and even approach SOTA with Caltech-101 silhouettes with 3-layer SBNs.

We also tested our log-likelihood estimates against the exact log-likelihood (by marginalized over the joint) of smaller single-layer SBNs with 20 stochastic units. The exact log-likelihood was -127.474 and our estimate with the unrefined approximate was -127.51 and -127.48 with 100 refinement steps. Overall, this result is consistent with those of Table 1, that iterative refinement improves the accuracy of log-likelihood estimates.

5.3 Posterior Improvement

In order to visualize the improvements due to refinement and to demonstrate IRVI as a general means of improvement for directed models at test, we generate N samples from the approximate posterior without ($\mathbf{h}_0^{(n)} \sim q_0(\mathbf{h}|\mathbf{x}; \boldsymbol{\theta}_0)$) and with refinement ($\mathbf{h}_K^{(n)} \sim q(\mathbf{h}|\mathbf{x}; \boldsymbol{\theta}_K)$), from a single-layer SBN with 20 stochastic units originally trained with RWS. We then use the samples from the approximate posterior to compute the expected conditional likelihood or average reconstruction: $\frac{1}{N}\sum_{n=1}^{N} p(\mathbf{x}|\mathbf{h}^{(n)})$. We used a restricted model with a lower number of stochastic units to demonstrate that refinement also works well with simple models, where the recognition network is more likely to "average" over latent configurations, giving a misleading evaluation of the model's generative capability.

We also refine the approximate posterior of a simplified version of the recognition network of a single-layer SBN with 200 units trained with RWS. We simplified the approximate posterior by first computing $\boldsymbol{\theta}_0(x)$, then randomly setting 80% of the variational parameters to 0.5.

Figure 2 shows improvement from refinement for 25 digits from the MNIST test dataset, where the samples chosen were those of which the expected reconstruction error of the original test sample was the most improved. The digits generated from the refined posterior are of higher quality, and in many cases the correct digit class is revealed.

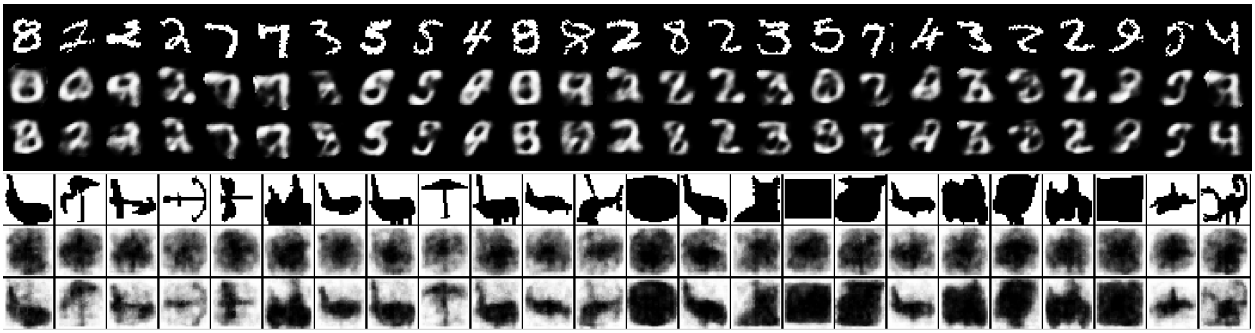

Figure 2: Top: Average reconstructions, $1/N \sum_{n=1}^{N} p(\mathbf{x}|\mathbf{h}^{(n)})$, for $\mathbf{h}^{(n)}$ sampled from the output of the recognition network, $q_0(\mathbf{h}|\mathbf{x};\boldsymbol{\theta}_0)$ (middle row) against those sampled from the refined posterior, $q_K(\mathbf{h}|\mathbf{x};\boldsymbol{\theta}_K)$ (bottom row) for $K = 20$ with a model trained on MNIST. Top row is ground truth. We used a smaller model with 20 latent variables to demonstrate that refinement can help even simple models fit the data well. Among the digits whose reconstruction changes the most, many changes correctly reveal the identity of the digit. Bottom: Average reconstructions for a single-layer model with 200 trained on Caltech-101 silhouettes. Instead of using the posterior from the recognition network, we derived a simpler version, setting 80% of the variational parameters from the recognition network to 0.5, then applied iterative refinement. Despite starting at a relatively poor of the posterior, refinement is able to retrieve much structure of the ground truth. This indicates that refinement can be used to do inference, starting with a simple approximate posterior.

This shows that, in many case where the recognition network indicates that the generative model cannot model a test sample correctly, refinement can more accurately reveal the model's capacity. With the simplified approximate posterior, refinement is able to retrieve most of the shape of images from the Caltech-101 silhouettes, despite only starting with 20% of the original parameters from the recognition network. This indicates that the work of inference need not all be done via a complex recognition network: iterative refinement can be used to aid in inference with a relatively simple approximate posterior.

5.4 Classification

Inference via a recognition network is not only a means of training a directed belief network, but also is a means of latent analysis. The best classification rates are achieved by supervised models, but purely predictive models do not teach us about the data.

In order to further demonstrate the importance of a quality posterior, we evaluate the unrefined ($\boldsymbol{\theta}_0(\mathbf{x})$) and the refined ($\boldsymbol{\theta}_K(\mathbf{x})$) parameters of the approximate posterior as input to a classifier. The unrefined parameters are from a model with 200 stochastic units trained with RWS on the MNIST dataset. For a classifier, we used a FFN with a single hidden layer with 100 sigmoid units, and did not back-propagate the gradients through the recognition network. As a baseline, we also trained a FFN using the raw observations, $\mathbf{x}$, as input with two layers with 200 and 100 sigmoid units each. Classifiers were trained for 200 epochs with a learning rate of 0.0001, 10% dropout rate, 0.0002 L2 weight decay. For the refined posterior, 20 mean-field updates with AIR were used. This experiment was repeated 10 times. On the MNIST test set, the baseline network achieved an error of $3.63 \pm 0.04\%$, the unrefined network, $5.61 \pm 0.06\%$, and the refined network, $5.20 \pm 0.03\%$, for a gain of $0.42 \pm 0.06\%$.

5.5 Iterative Refinement with Continuous Variables

For continuous latent variables, rather than use the re-parameterization to pass gradients to the global variational parameters, $\boldsymbol{\psi}$, as in VAE, we can apply the gradients iteratively at the local parameters, $\boldsymbol{\theta}$, improving the initial estimate provided by $q_0(\mathbf{h}|\mathbf{x};\boldsymbol{\psi})$. The iterative transition follows the stochastic gradient of the lower bound w.r.t. $\boldsymbol{\theta}$, which can be approximated easily by Equation 7. In the limit of $K = 0$, we do not arrive at VAE, as the gradients are never passed through the approximate posterior during learning. However, as the complete computational graph involves a series of differentiable variables, $\boldsymbol{\theta}_k$, in addition to auxiliary noise, it is possible to pass gradients through

Table 2: Lower bounds and NLL for various continuous latent variable models and training algorithms along with the corresponding VAE estimates. We use 200 latent Gaussian variables. †From Salimans et al. (2015). §From Rezende & Mohamed (2015). ‡From Burda et al. (2015).

Model	$\leq$ -log p(x)	$\approx$ -log p(x)
VAE	94.48	89.31
VAE (w/ refinement)	90.57	88.53
$\text{GDIR}_{50,20}$	90.60	88.54
VAE†	94.18	88.95
HVI_1†	91.70	88.08
HVI_8†	88.30	85.51
VAE §	89.9	
DLGM+NF_{80} §	**85.1**	
VAE‡		86.35
IWAE ($K = 50$)‡		**84.78**

GDIR to the recognition network parameters, ψ, during learning, though we do not here.

For continuous latent variables, we used the same network structure as in (Kingma & Welling, 2013; Salimans et al., 2015). Results for GDIR are presented in Table 2 for the MNIST dataset, and included for comparison are methods for learning non-factorial latent distributions for Gaussian variables and the corresponding result for VAE, the baseline.

Though GDIR can improve the posterior in VAE, our results show that VAE is at an upper-bound for learning with a factorized posterior on the MNIST dataset. Further improvements on this dataset must be made by using a non-factorized posterior. GDIR may still provide improvement for training models with other datasets, and we leave this for future work.

6 Conclusion

We have introduced iterative refinement for variational inference (IRVI), a simple, yet effective and flexible, approach for training and evaluating directed belief networks that works by improving the approximate posterior from a recognition network. We demonstrate IRVI using adaptive importance sampling iterative refinement (AIR), which uses importance sampling at each iterative step and works with discrete or continuous units, and show that it can both be used to effectively train deep directed graphical models and to greatly improve evaluation at test, independent of how the model was originally trained.

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.

Gregor, Karol, Danihelka, Ivo, Mnih, Andriy, Blundell, Charles, and Wierstra, Daan. Deep autoregressive networks. *arXiv preprint arXiv:1310.8499*, 2013.

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.

Honkela, Antti, Raiko, Tapani, Kuusela, Mikael, Tornio, Matti, and Karhunen, Juha. Approximate riemannian conjugate gradient learning for fixed-form variational bayes. *The Journal of Machine Learning Research*, 11:3235–3268, 2010.

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

Mnih, Andriy and Gregor, Karol. Neural variational inference and learning in belief networks. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 1791–1799, 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.

Oh, Man-Suk and Berger, James O. Adaptive importance sampling in monte carlo integration. *Journal of Statistical Computation and Simulation*, 41(3-4):143–168, 1992.

Raiko, Tapani, Berglund, Mathias, Alain, Guillaume, and Dinh, Laurent. Techniques for learning binary stochastic feedforward neural networks. *arXiv preprint arXiv:1406.2989*, 2014.

Rezende, Danilo J, Mohamed, Shakir, and Wierstra, Daan. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 1278–1286, 2014.

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

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

Salakhutdinov, Ruslan and Larochelle, Hugo. Efficient learning of deep boltzmann machines. In *International Conference on Artificial Intelligence and Statistics*, pp. 693–700, 2010.

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, Kingma, Diederik, and Welling, Max. Markov chain monte carlo and variational inference: Bridging the gap. In Blei, David and Bach, Francis (eds.), *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pp. 1218–1226. JMLR Workshop and Conference Proceedings, 2015. URL `http://jmlr.org/proceedings/papers/v37/salimans15.pdf`.

Saul, Lawrence K, Jaakkola, Tommi, and Jordan, Michael I. Mean field theory for sigmoid belief networks. *Journal of artificial intelligence research*, 4(1):61–76, 1996.

Tang, Yichuan and Salakhutdinov, Ruslan R. Learning stochastic feedforward neural networks. In *Advances in Neural Information Processing Systems*, pp. 530–538, 2013.