Accurate and Conservative Estimates of MRF Log-likelihood using Reverse Annealing

Yuri Burda[1]
Fields Institute
University of Toronto

Roger B. Grosse[1]
Department of Computer Science
University of Toronto

Ruslan Salakhutdinov
Department of Computer Science
University of Toronto

Abstract

Markov random fields (MRFs) are difficult to evaluate as generative models because computing the test log-probabilities requires the intractable partition function. Annealed importance sampling (AIS) is widely used to estimate MRF partition functions, and often yields quite accurate results. However, AIS is prone to overestimate the log-likelihood with little indication that anything is wrong. We present the Reverse AIS Estimator (RAISE), a stochastic lower bound on the log-likelihood of an approximation to the original MRF model. RAISE requires only the same MCMC transition operators as standard AIS. Experimental results indicate that RAISE agrees closely with AIS log-probability estimates for RBMs, DBMs, and DBNs, but typically errs on the side of underestimating, rather than overestimating, the log-likelihood.

1 Introduction

In recent years, there has been a resurgence of interest in learning deep representations due to the impressive performance of deep neural networks across a range of tasks. Generative modeling is an appealing method of learning representations, partly because one can directly evaluate a model by measuring the probability it assigns to held-out test data. Restricted Boltzmann machines (RBMs; Smolensky, 1986) and deep Boltzmann machines (DBMs; Salakhutdinov and Hinton, 2009) are highly effective at modeling various complex visual datasets (*e.g.* Salakhutdinov and Murray, 2008; Salakhutdinov and Hinton, 2009). Unfortunately, measuring their likelihood exactly is intractable because it requires computing the partition function of a Markov random field (MRF).

Annealed importance sampling (AIS; Neal, 2001) has emerged as the state-of-the-art algorithm for estimating MRF partition functions, and is widely used to evaluate MRFs as generative models (Salakhutdinov and Murray, 2008; Theis et al., 2011). AIS is a consistent estimator of the partition function (Neal, 2001), and often performs very well in practice. However, it has a property which makes it unreliable: it tends to underestimate the partition function, which leads to overly optimistic measures of the model likelihood. In some cases, it can overestimate the log-likelihood by tens of nats (*e.g.* Grosse et al., 2013), and one cannot be sure whether impressive test log-probabilities result from a good model or a bad partition function estimator. The difficulty of evaluating likelihoods has led researchers to propose alternative generative models for which the log-likelihood can be computed exactly (Larochelle and Murray, 2011; Poon and Domingos, 2011) or lower bounded (Gregor et al., 2014; Mnih and Gregor, 2014), but RBMs and DBMs remain the state-of-the-art for modeling complex data distributions.

Bengio et al. (2013) highlighted the problem of optimistic RBM log-likelihood estimates and proposed a pessimistic estimator based on nonparametric density estimation. Unfortunately, they reported that their method tends to underestimate log-likelihoods by tens of nats on standard benchmarks, which is insufficient accuracy since the difference between competing models is often on the order of one nat.

We introduce the Reverse AIS Estimator (RAISE), an algorithm which computes conservative estimates of MRF log-likelihoods, but which achieves similar accuracy to AIS in practice. In particular, consider an approximate generative model defined as the distribution of approximate samples computed by AIS. Using importance sampling with a carefully chosen proposal distribution, RAISE computes a stochastic lower bound on the log-likelihood of the approximate model. RAISE is simple to implement, as it requires only the same MCMC transition operators as standard AIS.

We evaluated RAISE by using it to estimate test log-

Appearing in Proceedings of the 18^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2015, San Diego, CA, USA. JMLR: W&CP volume 38.

[1] Authors contributed equally

probabilities of several RBMs, DBMs, and Deep Belief Networks (DBNs). The RAISE estimates agree closely with the true log-probabilities on small RBMs where the partition function can be computed exactly. Furthermore, they agree closely with the standard AIS estimates for full-size RBMs, DBMs, and DBNs. Since one estimate is optimistic and one is pessimistic, this agreement is an encouraging sign that both estimates are close to the correct value. Our results suggest that AIS and RAISE, used in conjunction, can provide a practical way of estimating MRF test log-probabilities.

2 Background

2.1 Restricted Boltzmann Machines

While our proposed method applies to general MRFs, we use as our running example a particular type of MRF called the restricted Boltzmann machine (RBM; Smolensky, 1986). An RBM is an MRF with a bipartite structure over a set of visible units $\mathbf{v} = (v_1, \ldots, v_{N_v})$ and hidden units $\mathbf{h} = (h_1, \ldots, h_{N_h})$. In this paper, for purposes of exposition, we assume that all of the variables are binary valued. In this case, the distribution over the joint state $\{\mathbf{v}, \mathbf{h}\}$ can be written as $f(\mathbf{v}, \mathbf{h})/\mathcal{Z}$, where

$$f(\mathbf{v}, \mathbf{h}) = \exp\left(\mathbf{a}^\top \mathbf{v} + \mathbf{b}^\top \mathbf{h} + \mathbf{v}^\top \mathbf{W} \mathbf{h}\right), \qquad (1)$$

and $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{W}$ denote the visible biases, hidden biases, and weights, respectively. The weights and biases are the RBM's trainable parameters.

To train the RBM's weights and biases, one can maximize the log-probability of a set of training examples $\mathbf{v}_{\text{tr}}^{(1)}, \ldots, \mathbf{v}_{\text{tr}}^{(M_{\text{tr}})}$. Since the log-likelihood gradient is intractable to compute exactly, it is typically approximated using contrastive divergence (Hinton, 2002) or persistent contrastive divergence (Tieleman, 2008). The performance of the RBM is then measured in terms of the average log-probability of a set of test examples $\mathbf{v}_{\text{test}}^{(1)}, \ldots, \mathbf{v}_{\text{test}}^{(M_{\text{test}})}$.

It remains challenging to evaluate the probability $p(\mathbf{v}) = f(\mathbf{v})/\mathcal{Z}$ of an example. The unnormalized probability $f(\mathbf{v}) = \sum_{\mathbf{h}} f(\mathbf{v}, \mathbf{h})$ can be computed exactly since the conditional distribution factorizes over the h_j. However, $\mathcal{Z}$ is intractable to compute exactly, and must be approximated.

RBMs can also be extended to deep Boltzmann machines (Salakhutdinov and Hinton, 2009) by adding one or more additional hidden layers. For instance, the joint distribution of a DBM with two hidden layers $\mathbf{h}_1$ and $\mathbf{h}_2$ can be written as $f(\mathbf{v}, \mathbf{h}_1, \mathbf{h}_2)/\mathcal{Z}$, where

$$f(\mathbf{v}, \mathbf{h}_1, \mathbf{h}_2) = \exp\left(\mathbf{a}^\top \mathbf{v} + \mathbf{b}_1^\top \mathbf{h}_1 + \mathbf{b}_2^\top \mathbf{h}_2 + \right.$$
$$\left. + \mathbf{v}^\top \mathbf{W}_1 \mathbf{h}_1 + \mathbf{h}_1^\top \mathbf{W}_2 \mathbf{h}_2\right). \qquad (2)$$

DBMs can be evaluated similarly to RBMs. The main difference is that the unnormalized probability $f(\mathbf{v}) = \sum_{\mathbf{h}_1, \mathbf{h}_2} f(\mathbf{v}, \mathbf{h}_1, \mathbf{h}_2)$ is intractable to compute exactly. However, Salakhutdinov and Hinton (2009) showed that, in practice, the mean-field approximation yields an accurate lower bound. Therefore, similarly to RBMs, the main difficulty in evaluating DBMs is estimating the partition function.

RBMs are also used as building blocks for training Deep Belief Networks (DBNs; Hinton et al., 2006). For example, a DBN with two hidden layers $\mathbf{h}_1$ and $\mathbf{h}_2$ is defined as the probability distribution

$$p(\mathbf{v}, \mathbf{h}_1, \mathbf{h}_2) = p_2(\mathbf{h}_1, \mathbf{h}_2) p_1(\mathbf{v} \,|\, \mathbf{h}_1), \qquad (3)$$

where $p_2(\mathbf{h}_1, \mathbf{h}_2)$ is the probability distribution of an RBM, and $p_1(\mathbf{v} \,|\, \mathbf{h}_1)$ is a product of independent logistic units. The unnormalized probability $f(\mathbf{v}) = \sum_{\mathbf{h}_1, \mathbf{h}_2} p_1(\mathbf{v} \,|\, \mathbf{h}_1) f_2(\mathbf{h}_1, \mathbf{h}_2)$ cannot be computed analytically, but can be approximated using importance sampling or a variational lower bound that utilizes a recognition distribution $q(\mathbf{h}_1 \,|\, \mathbf{v})$ approximating the posterior $p(\mathbf{h}_1 \,|\, \mathbf{v})$ (Hinton et al., 2006).

2.2 Partition Function Estimation

Often we have a probability distribution $p_{\text{tgt}}(\mathbf{x}) = f_{\text{tgt}}(\mathbf{x})/\mathcal{Z}_{\text{tgt}}$ (which we call the *target distribution*) defined on a space $\mathcal{X}$, where $f_{\text{tgt}}(\mathbf{x})$ can be computed efficiently for a given $\mathbf{x} \in \mathcal{X}$, and $\mathcal{Z}_{\text{tgt}}$ is an intractable normalizing constant. There are two particular cases which concern us here. First, p_{tgt} may correspond to a Markov random field (MRF), such as an RBM, where $f_{\text{tgt}}(\mathbf{x})$ denotes the product of all potentials, and $\mathcal{Z}_{\text{tgt}} = \sum_{\mathbf{x}} f_{\text{tgt}}(\mathbf{x})$ is the partition function of the graphical model.

The second case is where one has a directed graphical model with latent variables $\mathbf{h}$ and observed variables $\mathbf{v}$. Here, the joint distribution $p(\mathbf{h}, \mathbf{v}) = p(\mathbf{h}) p(\mathbf{v} \,|\, \mathbf{h})$ can be tractably computed for any particular pair $(\mathbf{h}, \mathbf{v})$. However, one often wants to compute the likelihood of a test example $p(\mathbf{v}_{\text{test}}) = \sum_{\mathbf{h}} p(\mathbf{h}, \mathbf{v}_{\text{test}})$. This can be placed in the above framework with

$$f_{\text{tgt}}(\mathbf{h}) = p(\mathbf{h}) p(\mathbf{v}_{\text{test}} \,|\, \mathbf{h}) \quad \text{and} \quad \mathcal{Z}_{\text{tgt}} = p(\mathbf{v}_{\text{test}}). \qquad (4)$$

Mathematically, the two partition function estimation problems outlined above are closely related, and the same classes of algorithms are applicable to each. However, they differ in terms of the behavior of approximate inference algorithms in the context of model selection. In particular, many algorithms, such as annealed importance sampling (Neal, 2001) and sequential Monte Carlo (del Moral et al., 2006), yield unbiased estimates $\hat{\mathcal{Z}}_{\text{tgt}}$ of the partition function, *i.e.* $\mathbb{E}[\hat{\mathcal{Z}}_{\text{tgt}}] = \mathcal{Z}_{\text{tgt}}$. Jensen's Inequality shows that such an estimator tends to underestimate the log partition function on average:

$$\mathbb{E}[\log \hat{\mathcal{Z}}_{\text{tgt}}] \leq \log \mathbb{E}[\hat{\mathcal{Z}}_{\text{tgt}}] = \log \mathcal{Z}_{\text{tgt}}. \qquad (5)$$

Algorithm 1 Annealed Importance Sampling

for $i = 1$ to M **do**
  $\mathbf{x}_0 \leftarrow$ sample from $p_0(\mathbf{x}) = f_{\text{ini}}(\mathbf{x})/\mathcal{Z}_{\text{ini}}$
  $w^{(i)} \leftarrow \mathcal{Z}_{\text{ini}}$
  for $k = 1$ to K **do**
    $w^{(i)} \leftarrow w^{(i)} \frac{f_k(\mathbf{x}_{k-1})}{f_{k-1}(\mathbf{x}_{k-1})}$
    $\mathbf{x}_k \leftarrow$ sample from $T_k(\cdot \,|\, \mathbf{x}_{k-1})$
  end for
end for
return $\hat{\mathcal{Z}}_{\text{tgt}} = \sum_{i=1}^{M} w^{(i)}/M$

In addition, Markov's inequality shows that it is unlikely to substantially overestimate $\log \mathcal{Z}_{\text{tgt}}$:

$$\Pr(\log \hat{\mathcal{Z}}_{\text{tgt}} > \log \mathcal{Z}_{\text{tgt}} + b) < e^{-b}. \tag{6}$$

For these reasons, we will refer to the estimator as a *stochastic lower bound* on $\log \mathcal{Z}_{\text{tgt}}$.

In the MRF situation, $\mathcal{Z}_{\text{tgt}}$ appears in the denominator, so underestimates of the log partition function translate into overestimates of the log-likelihood. This is problematic, since inaccurate partition function estimates can lead one to dramatically overestimate the performance of one's model. This problem has led researchers to consider alternative generative models where the likelihood can be tractably computed. By contrast, in the directed case, the partition function is the test log-probability (4), so underestimates correspond to overly conservative measures of performance. For example, the fact that sigmoid belief networks (Neal, 1992) have tractable lower (rather than upper) bounds is commonly cited as a reason to prefer them over RBMs and DBMs (*e.g.* Mnih and Gregor, 2014).

We note that it is possible to achieve stronger tail bounds than (6) by combining multiple unbiased estimates in clever ways (Gogate et al., 2007).

2.3 Annealed Importance Sampling

Annealed importance sampling (AIS) is an algorithm which estimates $\mathcal{Z}_{\text{tgt}}$ by gradually changing, or "annealing," a distribution. In particular, one must specify a sequence of $K+1$ intermediate distributions $p_k(\mathbf{x}) = f_k(\mathbf{x})/\mathcal{Z}_k$ for $k = 0, \dots K$, where $p_{\text{ini}}(\mathbf{x}) = p_0(\mathbf{x})$ is a tractable initial distribution, and $p_{\text{tgt}}(\mathbf{x}) = p_K(\mathbf{x})$ is the intractable target distribution. For simplicity, assume all distributions are strictly positive on $\mathcal{X}$. For each p_k, one must also specify an MCMC transition operator T_k (e.g. Gibbs sampling) which leaves p_k invariant. AIS alternates between MCMC transitions and importance sampling updates, as shown in Algorithm 1.

The output of AIS is an unbiased estimate $\hat{\mathcal{Z}}_{\text{tgt}}$ of $\mathcal{Z}_{\text{tgt}}$. Importantly, unbiasedness is not an asymptotic property, but holds for any K (Neal, 2001; Jarzynski, 1997). Neal (2001) demonstrated this by viewing AIS as an importance sampling estimator over an extended state space. In particular, define the distributions

$$q_{\text{fwd}}(\mathbf{x}_{0:K-1}) = p_0(\mathbf{x}_0) \prod_{k=1}^{K-1} T_k(\mathbf{x}_k \,|\, \mathbf{x}_{k-1}) \tag{7}$$

$$f_{\text{rev}}(\mathbf{x}_{0:K-1}) = f_{\text{tgt}}(\mathbf{x}_{K-1}) \prod_{k=1}^{K-1} \tilde{T}_k(\mathbf{x}_{k-1} \,|\, \mathbf{x}_k), \tag{8}$$

where $\tilde{T}_k(\mathbf{x}' \,|\, \mathbf{x}) = T_k(\mathbf{x} \,|\, \mathbf{x}') p_k(\mathbf{x}')/p_k(\mathbf{x})$ is the reverse transition operator for T_k. Here, q_{fwd} represents the sequence of states generated by AIS, and f_{rev} is a fictitious (unnormalized) reverse chain which begins with an exact sample from p_{tgt} and applies the transitions in reverse order. Neal (2001) showed that the AIS weights correspond to the importance weights for f_{rev} with q_{fwd} as the proposal distribution.

The mathematical formulation of AIS leaves much flexibility for choosing intermediate distributions. The choice of distributions can have a large effect on the performance of AIS (Grosse et al., 2013), but the most common choice is to take geometric averages of the initial and target distributions:

$$p_\beta(\mathbf{x}) = f_\beta(\mathbf{x})/\mathcal{Z}(\beta) = f_{\text{ini}}(\mathbf{x})^{1-\beta} f_{\text{tgt}}(\mathbf{x})^{\beta}/\mathcal{Z}(\beta), \tag{9}$$

where $0 = \beta_0 < \beta_1 < \dots < \beta_K = 1$ defines the annealing schedule. Commonly, f_{ini} is the uniform distribution, and (9) reduces to $p_\beta(\mathbf{x}) = f_{\text{tgt}}(\mathbf{x})^\beta/\mathcal{Z}(\beta)$. This motivates the term "annealing", and β resembles an inverse temperature parameter. As in simulated annealing, the "hotter" distributions often allow faster mixing between modes which are isolated in p_{tgt}. Geometric averages are widely used because they often have a simple form; for instance, the geometric average of two RBMs is obtained by linearly averaging the weights and biases. The values of β can be spaced evenly between 0 and 1, although other schedules have been explored (Neal, 1996; Behrens et al., 2012; Calderhead and Girolami, 2009).

3 Reverse AIS Estimator

A significant difficulty in evaluating MRFs is that it is intractable to compute the partition function. Furthermore, the commonly used algorithms, such as AIS, tend to *overestimate* the log-likelihood. If we cannot hope to obtain provably accurate partition function estimates, it would be far preferable for algorithms to *underestimate*, rather than *overestimate*, the log-likelihoods. This would save us from the embarrassment of reporting unrealistically high test log-probability scores for a given dataset. In this section, we define an approximate generative model which becomes equivalent to the MRF in the limit of infinite computation. We then present a procedure for obtaining unbiased estimates of the probability of a test example (and therefore a stochastic lower bound on the test log-probability) under the approximate model.

Algorithm 2 Reverse AIS Estimator (RAISE)

for $i = 1$ to M **do**
  $\mathbf{h}_K \leftarrow$ sample from $p_{\text{tgt}}(\mathbf{h} \mid \mathbf{v}_{\text{test}})$
  $w^{(i)} \leftarrow f_{\text{tgt}}(\mathbf{v}_{\text{test}})/\mathcal{Z}_0$
  for $k = K-1$ to 0 **do**
    $\mathbf{x}_k \leftarrow$ sample from $\tilde{T}_k(\cdot \mid \mathbf{x}_{k+1})$
    $w^{(i)} \leftarrow w^{(i)} \frac{f_k(\mathbf{x}_k)}{f_{k+1}(\mathbf{x}_k)}$
  end for
end for
return $\hat{p}_{\text{ann}}(\mathbf{v}_{\text{test}}) = \sum_{i=1}^{M} w^{(i)}/M$

3.1 Case of Tractable Posterior

In this section, we denote the model state as $\mathbf{x} = (\mathbf{v}, \mathbf{h})$, with $\mathbf{v}$ observed and $\mathbf{h}$ unobserved. Let us first assume the conditional distribution $p_{\text{tgt}}(\mathbf{h} \mid \mathbf{v})$ is tractable, as is the case for RBMs. Define the following generative process, which corresponds to the sequence of transitions in AIS:

$$p_{\text{fwd}}(\mathbf{x}_{0:K}) = p_0(\mathbf{x}_0) \prod_{k=1}^{K} T_k(\mathbf{x}_k \mid \mathbf{x}_{k-1}). \tag{10}$$

By taking the final visible states of this process, we obtain a generative model (which we term the *annealing model*) which approximates $p_{\text{tgt}}(\mathbf{v})$:

$$p_{\text{ann}}(\mathbf{v}_K) = \sum_{\mathbf{x}_{0:K-1}, \mathbf{h}_K} p_{\text{fwd}}(\mathbf{x}_{0:K-1}, \mathbf{h}_K, \mathbf{v}_K). \tag{11}$$

Suppose we are interested in estimating the probability of a test example $\mathbf{v}_{\text{test}}$. We use as a proposal distribution a reverse chain starting from $\mathbf{v}_{\text{test}}$. In the annealing metaphor, this corresponds to gradually "melting" the distribution:

$$q_{\text{rev}}(\mathbf{x}_{0:K-1}, \mathbf{h}_K \mid \mathbf{v}_{\text{test}}) = p_{\text{tgt}}(\mathbf{h}_K \mid \mathbf{v}_{\text{test}}) \prod_{k=1}^{K} \tilde{T}_k(\mathbf{x}_{k-1} \mid \mathbf{x}_k),$$

where we identify $\mathbf{v}_k = \mathbf{v}_{\text{test}}$, and $\tilde{T}_k(\mathbf{x}' \mid \mathbf{x}) = T_k(\mathbf{x} \mid \mathbf{x}') p_k(\mathbf{x}')/p_k(\mathbf{x})$ is the reverse transition operator for T_k. We then obtain the following identity:

$$\begin{aligned}
p_{\text{ann}}(\mathbf{v}_{\text{test}}) &= \mathbb{E}_{q_{\text{rev}}}\left[\frac{p_{\text{fwd}}(\mathbf{x}_{0:K-1}, \mathbf{h}_K, \mathbf{v}_{\text{test}})}{q_{\text{rev}}(\mathbf{x}_{0:K-1}, \mathbf{h}_K \mid \mathbf{v}_{\text{test}})}\right] \\
&= \mathbb{E}_{q_{\text{rev}}}\left[\frac{p_0(\mathbf{x}_0)}{p_{\text{tgt}}(\mathbf{h}_K \mid \mathbf{v}_{\text{test}})} \prod_{k=1}^{K} \frac{T_k(\mathbf{x}_k \mid \mathbf{x}_{k-1})}{\tilde{T}_k(\mathbf{x}_{k-1} \mid \mathbf{x}_k)}\right] \\
&= \mathbb{E}_{q_{\text{rev}}}\left[\frac{p_0(\mathbf{x}_0)}{p_{\text{tgt}}(\mathbf{h}_K \mid \mathbf{v}_{\text{test}})} \prod_{k=1}^{K} \frac{f_k(\mathbf{x}_k)}{f_k(\mathbf{x}_{k-1})}\right] \\
&= \mathbb{E}_{q_{\text{rev}}}\left[\frac{p_0(\mathbf{x}_0)}{p_{\text{tgt}}(\mathbf{h}_K \mid \mathbf{v}_{\text{test}})} \frac{f_{\text{tgt}}(\mathbf{x}_K)}{f_0(\mathbf{x}_0)} \prod_{k=0}^{K-1} \frac{f_k(\mathbf{x}_k)}{f_{k+1}(\mathbf{x}_k)}\right] \\
&= \mathbb{E}_{q_{\text{rev}}}\left[\frac{f_K(\mathbf{v}_{\text{test}})}{\mathcal{Z}_0} \prod_{k=0}^{K-1} \frac{f_k(\mathbf{x}_k)}{f_{k+1}(\mathbf{x}_k)}\right] \\
&\triangleq \mathbb{E}_{q_{\text{rev}}}[w].
\end{aligned} \tag{12}$$

This yields the following algorithm: generate M samples from q_{rev}, and average the values w defined in (12). There is no need to store the full chains, since the weights can be updated online. We refer to this algorithm as the Reverse AIS Estimator, or RAISE. The full algorithm is given in Algorithm 2. We note that RAISE is straightforward to implement, as it requires only the same MCMC transition operators as standard AIS.

Our derivation (12) mirrors the derivation of AIS by Neal (2001). The difference is that in AIS, the reverse chain is merely hypothetical; in RAISE, the reverse chain is simulated, and it is the forward chain which is hypothetical.

By (12), the weights w are an unbiased estimator of the probability $p_{\text{ann}}(\mathbf{v}_{\text{test}})$. Therefore, following the discussion of Section 2.2, $\log w$ is a stochastic lower bound on $\log p_{\text{ann}}(\mathbf{v}_{\text{test}})$. Furthermore, since p_{ann} converges to p_{tgt} in probability as $K \to \infty$ (Neal, 2001), we would heuristically expect RAISE to yield a conservative estimate of $\log p_{\text{tgt}}(\mathbf{v}_{\text{test}})$. This is not strictly guaranteed, however; RAISE may overestimate $\log p_{\text{tgt}}(\mathbf{v}_{\text{test}})$ for finite K if $p_{\text{ann}}(\mathbf{v}_{\text{test}}) > p_{\text{tgt}}(\mathbf{v}_{\text{test}})$, which is possible if the AIS approximation somehow attenuates pathologies in the original MRF. (One such example is described in Section 5.1.) However, since RAISE is a stochastic lower bound on the log-probabilities under the annealing model, we can strictly rule out the possibility of RAISE reporting unrealistically high test log-probabilities for a given dataset, a situation frequently observed with AIS.

3.2 Extension to Intractable Posterior Distributions

Because Algorithm 2 begins with an exact sample from the conditional distribution $p_{\text{tgt}}(\mathbf{h} \mid \mathbf{v}_{\text{test}})$, it requires that this distribution be tractable. However, many models of interest, such as DBMs, have intractable posterior distributions. To deal with this case, we augment the forward chain with an additional heating step, such that the conditional distribution in the final step is tractable, but the distribution over $\mathbf{v}$ agrees with that of p_{ann} in (11). We make the further (weak) assumption that $p_0(\mathbf{h} \mid \mathbf{v})$ is tractable. Let $T_k^{(\mathbf{v})}$ denote an MCMC transition operator which preserves $p_k(\mathbf{v}, \mathbf{h})$, but does not change $\mathbf{v}$. For example, it may cycle through Gibbs updates to all variables except $\mathbf{v}$. The forward chain then has the following distribution:

$$\begin{aligned}
p_{\text{fwd}}(\mathbf{x}_{0:K}, \mathbf{h}'_{0:K-1}) = p_0(\mathbf{x}_0) &\prod_{k=1}^{K} T_k(\mathbf{x}_k \mid \mathbf{x}_{k-1}) \\
&\prod_{k=0}^{K-1} T_k^{(\mathbf{v}_K)}(\mathbf{h}'_k \mid \mathbf{h}'_{k+1}),
\end{aligned}$$

where we identify $\mathbf{h}'_K = \mathbf{h}_K$. The reverse distribution is given by:

$$\begin{aligned}
&q_{\text{rev}}(\mathbf{x}_{0:K-1}, \mathbf{h}_K, \mathbf{h}'_{0:K-1} \mid \mathbf{v}_{\text{test}}) = \\
&p_0(\mathbf{h}'_0 \mid \mathbf{v}_{\text{test}}) \prod_{k=0}^{K-1} \tilde{T}_k^{(\mathbf{v}_{\text{test}})}(\mathbf{h}'_{k+1} \mid \mathbf{h}'_k) \prod_{k=1}^{K} \tilde{T}_k(\mathbf{x}_{k-1} \mid \mathbf{x}_k).
\end{aligned}$$

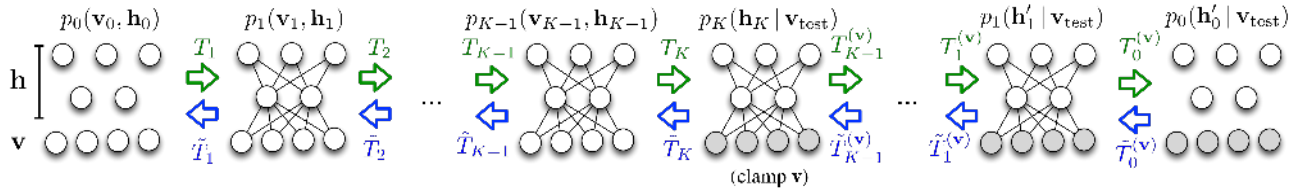


Figure 1: A schematic of RAISE for intractable distributions, applied to DBMs. Green: generative model. Blue: proposal distribution. At the top is shown which distribution the variables at each step are meant to approximate.

Algorithm 3 RAISE with intractable posterior

for $i = 1$ to M **do**
  $\mathbf{h}'_0 \leftarrow$ sample from $p_0(\mathbf{h} \mid \mathbf{v}_{\text{test}})$
  $w^{(i)} \leftarrow p_0(\mathbf{v}_{\text{test}})$
  for $k = 1$ to K **do**
    $\mathbf{h}'_k \leftarrow$ sample from $\tilde{T}_k^{(\mathbf{v}_{\text{test}})}(\cdot \mid \mathbf{h}'_{k-1})$
    $w^{(i)} \leftarrow w^{(i)} \frac{f_k(\mathbf{h}'_k, \mathbf{v}_{\text{test}})}{f_{k-1}(\mathbf{h}'_k, \mathbf{v}_{\text{test}})}$
  end for
  for $k = K-1$ to 0 **do**
    $\mathbf{x}_k \leftarrow$ sample from $\tilde{T}_k(\cdot \mid \mathbf{x}_{k+1})$
    $w^{(i)} \leftarrow w^{(i)} \frac{f_k(\mathbf{x}_k)}{f_{k+1}(\mathbf{x}_k)}$
  end for
end for
return $\hat{p}_{\text{ann}}(\mathbf{v}_{\text{test}}) = \sum_{i=1}^M w^{(i)}/M$

The unbiased estimator is derived similarly to that of Section 3:

$$w \triangleq \frac{p_{\text{fwd}}(\mathbf{x}_{0:K-1}, \mathbf{h}_K, \mathbf{v}_{\text{test}}, \mathbf{h}'_{0:K-1})}{q_{\text{rev}}(\mathbf{x}_{0:K-1}, \mathbf{h}_K, \mathbf{h}'_{0:K-1} \mid \mathbf{v}_{\text{test}})} \tag{13}$$
$$= p_0(\mathbf{v}_{\text{test}}) \prod_{k=0}^{K-1} \frac{f_k(\mathbf{x}_k)}{f_{k+1}(\mathbf{x}_k)} \prod_{k=1}^{K} \frac{f_k(\mathbf{h}'_k, \mathbf{v}_{\text{test}})}{f_{k-1}(\mathbf{h}'_k, \mathbf{v}_{\text{test}})}$$

The full algorithm is shown in Algorithm 3, and a schematic for the case of DBMs is shown in Figure 1.

3.3 Interpretation as Unrolling

Hinton et al. (2006) showed that the Gibbs sampling procedure for a binary RBM could be interpreted as generating from an infinitely deep sigmoid belief net with shared weights. They used this insight to derive a greedy training procedure for Deep Belief Nets (DBNs), where one unties the weights of a single layer at a time. Furthermore, they observed that one could perform approximate inference in the belief net using the transpose of the generative weights to compute a variational approximation.

We note that, for RBMs, RAISE can similarly be viewed as a form of unrolling: the annealed generative model p_{ann} can be viewed as a belief net with $K+1$ layers. Furthermore, the RAISE proposal distribution can be viewed as using the transpose of the weights to perform approximate inference. (The difference from approximate inference in DBNs is that RAISE samples the units rather than using the mean-field approximation).

This interpretation of RAISE suggests a method of applying it to DBNs. The generative model is obtained by unrolling the RBM on top of the directed layers as shown in Figure 2. The proposal distribution uses the transposes of the DBN weights for each of the directed layers. The rest is the same as the ordinary RAISE for the unrolled part of the model.

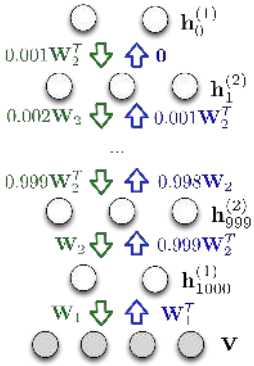


Figure 2: RAISE applied to a DBN unrolled into a very deep sigmoid belief net, for $K = 1000$ intermediate distributions. **Green:** generative model. **Blue:** proposal distribution.

4 Variance Reduction using Control Variates

One of the virtues of log-likelihood estimation using AIS is its speed: the partition function need only be estimated once. RAISE, unfortunately, must be run separately for every test example. We would therefore prefer to compute the RAISE estimate for only a small number of test examples. Unfortunately, subsampling the test examples introduces a significant source of variability: as different test examples can have wildly different log-likelihoods[2], the estimate of the average log-likelihood can vary significantly depending which batch of examples is selected. We attenuate this variability using the method of control variates (Ross, 2006), a variance reduction technique which has also been applied to black-box variational inference (Ranganath et al., 2014).

If $Y_1, \ldots, Y_n$ are independent samples of a random variable Y, then the sample average $\frac{1}{n}\sum_{i=1}^n Y_i$ is an unbiased estimator of $\mathbb{E}[Y]$ with variance $\text{Var}[Y]/n$. If X is another random variable (which ideally is both cheap to compute

[2]This effect can be counterintuitively large due to different complexities of different categories; *e.g.*, for the mnistCD25-500 RBM, the average log-likelihood of handwritten digits "1" was 56.6 nats higher than the average log-likelihood of digits "8".

and highly correlated with Y), then for any scalar α,

$$\frac{1}{n}\sum_{i=1}^{n}(Y_i - \alpha X_i) + \frac{\alpha}{N}\sum_{i=1}^{N} X_i \tag{14}$$

is an unbiased estimator of $\mathbb{E}[Y]$ with variance

$$\frac{\mathrm{Var}[Y - \alpha X]}{n} + \alpha^2 \frac{\mathrm{Var}[X]}{N} + 2\alpha \frac{\mathrm{Cov}[Y - \alpha X, X]}{n}.$$

In our experiments, Y is the RAISE estimate of the log-probability of a test example, and X is the (exact or estimated) log unnormalized probability under the original MRF. Since the unnormalized probability under the MRF is significantly easier to evaluate than the log-probability under the annealing model, we can let N to be much larger than n; we set $n = 100$ and let N be the total number of test examples. Since the annealing model is an approximation to the MRF, the two models should assign similar log-probabilities, so we set $\alpha = 1$. Hence we expect the variance of $Y - X$ to be smaller than the variance of Y, and thus (14) to have a significantly smaller variance than the sample average. Empirically, we have found that $Y - X$ has significantly smaller variance than Y, even when the number of intermediate distributions is relatively small.

5 Experimental Results

We have evaluated RAISE on several MRFs to determine if its log-probability estimates are both accurate and conservative. We compared our estimates against those obtained from standard AIS. We also compared against the exact log-probabilities of small models for which the partition function can be computed exactly (Salakhutdinov and Murray, 2008). AIS is expected to overestimate the true log-probabilities while RAISE is expected to underestimate them. Hence, a close agreement between the two estimators would be a strong indication of accurate estimates.

We considered two datasets: (1) the MNIST handwritten digit dataset (LeCun et al., 1998), which has long served as a benchmark for both classification and density modeling, and (2) the Omniglot dataset (Lake et al., 2013), which contains images of handwritten characters across many world alphabets.[3]

Both AIS and RAISE can be used with any sequence of intermediate distributions. For simplicity, in all of our experiments, we used the geometric averages path (9) with linear spacing of the parameter β. We tested two choices of initial distribution p_{ini}: the uniform distribution, and the data base rate (DBR) distribution (Salakhutdinov and Murray, 2008), where all units are independent, all hidden units are uniform, and the visible biases are set to match the average pixel values in the training set. In all cases, our MCMC transition operator was Gibbs sampling.

[3]We used the standard split of MNIST into 60,000 training and 10,000 test examples and a random split of Omniglot into 24,345 training and 8,070 test examples. In both cases, the inputs are 28×28 binary images.

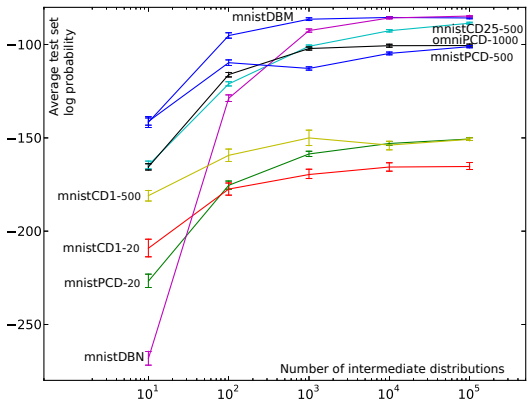


Figure 3: RAISE estimates of average test log-probabilities using uniform p_{ini}. The log-probability estimates tend to increase with the number of intermediate distributions, suggesting that RAISE is a conservative estimator.

We estimated the log-probabilities of a random sample of 100 examples from the test set using RAISE and used the method of control variates (Sec. 4) to estimate the average log-probabilities on the full test dataset. For RBM experiments, the control variate was the RBM log unnormalized probability, $\log f(\mathbf{v})$, whereas for DBMs and DBNs, we used an estimate based on simple importance sampling as described below. For each of the 100 test examples, RAISE was run with 50 independent chains, while the AIS partition function estimates used 5,000 chains; this closely matched the computation time per intermediate distribution between the two methods. Each method required about 1.5 hours with the largest number of intermediate distributions ($K = 100{,}000$).

5.1 Restricted Boltzmann Machines

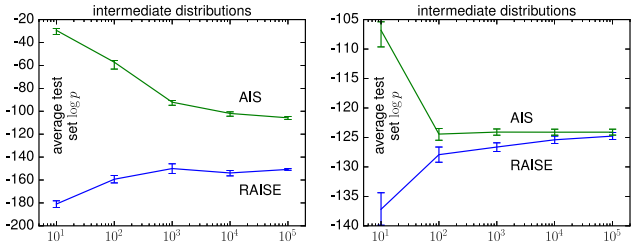
We considered models trained using two algorithms: contrastive divergence (CD; Hinton, 2002) with both 1 and 25 CD steps, and persistent contrastive divergence (PCD; Tieleman, 2008). We will refer to the RBMs by the dataset, training algorithm, and the number of hidden units. For example, "mnistCD25-500" denotes an RBM with 500 hidden units, trained on MNIST using 25 CD steps. The MNIST trained RBMs are the same ones evaluated by Grosse et al. (2013). We also provide comparisons to the Conservative Sampling-based Log-likelihood (CSL) estimator of Bengio et al. (2013).[4]

Figure 3 shows the average RAISE test log-probability estimates for all of the RBMs as a function of the number of intermediate distributions. In all of these examples, as expected, the estimated log-probabilities tended to increase

[4]The number of chains and number of Gibbs steps for CSL were chosen to match the total number of Gibbs steps required by RAISE and AIS for $K = 100{,}000$.

			uniform			data base rates		
Model	exact	CSL	RAISE	AIS	gap	RAISE	AIS	gap
mnistCD1-20	-164.50	-185.74	-165.33	-164.51	0.82	-164.11	-164.50	-0.39
mnistPCD-20	-150.11	-152.13	-150.58	-150.04	0.54	-150.17	-150.10	0.07
mnistCD1-500	—	-566.91	-150.78	-106.52	44.26	-124.77	-124.09	0.68
mnistPCD-500	—	-138.76	-101.07	-99.99	1.08	-101.26	-101.28	-0.02
mnistCD25-500	—	-145.26	-88.51	-86.42	2.09	-86.39	-86.35	0.04
omniPCD-1000	—	-144.25	-100.47	-100.45	0.02	-100.46	-100.46	0.00

Table 1: RAISE and AIS average test log-probabilities using 100,000 intermediate distributions and both choices of p_{ini}. **CSL:** the estimator of Bengio et al. (2013). **gap:** the difference AIS − RAISE



(a) uniform (b) data base rates

Figure 4: AIS and RAISE estimates of mnistCD1-500 average test log-probabilities have a significant gap when annealing from a uniform initial distribution. However, they agree closely when annealing from the data base rates.

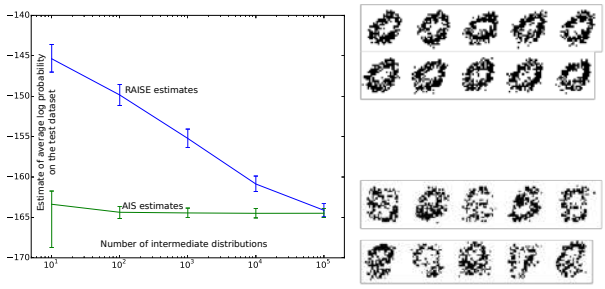


Figure 5: The mnistCD1-20 RBM, where we observed RAISE to overestimate the RBM's test log-probabilities. **Left:** Average test log-probability estimates as a function of K. **Top right:** 10 independent samples from the RBM. **Bottom right:** 10 independent samples from the annealing model p_{ann} with 10 intermediate distributions. The p_{ann} samples, while poor, show greater diversity compared to the RBM samples, consistent with p_{ann} better matching the data distribution.

with the number of intermediate distributions, consistent with RAISE being a conservative log-probability estimator.

Table 1 shows the final average test log-probability estimates obtained using CSL as well as both RAISE and AIS with 100,000 intermediate distributions. In all of the trials using the DBR initial distribution, the estimates of AIS and RAISE agreed to within 1 nat, and in many cases, to within 0.1 nats. The CSL estimator, on the other hand, underestimated $\log p_{\text{tgt}}$ by tens of nats in almost all cases, which is insufficient accuracy since well-trained models often differ by only a few nats.

We observed that the DBR initial distribution gave consistently better agreement between the two methods compared with the uniform distribution, consistent with the results of Salakhutdinov and Murray (2008). The largest discrepancy, 44.26 nats, was for mnistCD1-500 with uniform p_{ini}; with DBR, the two methods differed by only 0.68. Figure 4 plots both estimates as a function of the number of initial distributions. In the uniform case, one might not notice the inaccuracy only by running AIS, as the AIS estimates may appear to level off. One could be tricked into reporting results that are tens of nats too high! By contrast, when both methods are run in conjunction, the inaccuracy of at least one of the methods becomes obvious.

As discussed in Section 3.1, RAISE is a stochastic lower bound on the log-likelihood of the annealing model p_{ann}, but not necessarily of the RBM itself. When p_{ann} is a good approximation to the RBM, RAISE gives a conservative estimate of the RBM log-likelihood. However, it is possible for RAISE to overestimate the RBM log-likelihood if p_{ann} models the data distribution better than the RBM itself, for instance if the approximation attenuates pathologies of the RBM. We observed a single instance of this in our RBM experiments: the mnistCD1-20 RBM, with the data base rate initialization. As shown in Figure 5, the RAISE estimates exceeded the AIS estimates for small K, and declined as K was increased. Since RAISE gives a stochastic lower bound on $\log p_{\text{ann}}$ and AIS gives a stochastic upper bound on $\log p_{\text{tgt}}$, this inversion implies that p_{ann} significantly outperformed the RBM itself. Indeed, the RBM (mistakenly) assigned 93% of its probability mass to a single hidden configuration, while the RAISE model spreads its probability mass among more diverse configurations.

In all of our other RBM experiments, the AIS and RAISE estimates with DBR initialization and $K = 100{,}000$ agreed to within 0.1 nats. Figure 6 shows one such case, for an RBM trained on the challenging Omniglot dataset.

Overall, the RAISE and AIS estimates using DBR initialization agreed closely in all cases, and RAISE gave conservative estimates in all but one case, suggesting that RAISE typically gives accurate and conservative estimates of RBM test log-probabilities.

5.2 Deep Boltzmann Machines

We used RAISE to estimate the average test log-probabilities of two DBM models trained on MNIST and Omniglot. The MNIST DBM has 2 hidden layers of size 500 and 1000, and the Omniglot DBM has 2 hidden layers

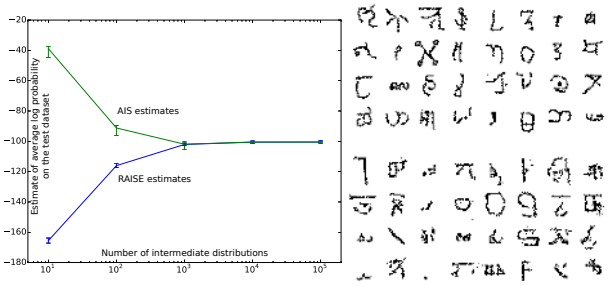


Figure 6: **Left:** AIS and RAISE estimates of omniPCD-1000 RBM average test log-probabilities with annealing from a uniform initial distribution **Top right:** 32 training samples from Omniglot training set **Bottom right:** 32 independent samples from the omniPCD-1000 RAISE model with 100,000 intermediate distributions.

	uniform			data base rates		
Model	RAISE	AIS	gap	RAISE	AIS	gap
MNIST DBM	-85.69	-85.72	-0.03	-85.74	-85.67	0.07
Omniglot DBM	-104.48	-110.86	-6.38	-102.64	-103.27	-0.63
MNIST DBN	-84.67	-84.49	0.18	—	—	—
Omniglot DBN	-100.78	-100.45	0.33	—	—	—

Table 2: Test log-probability estimates for deep models with $K = 100{,}000$. **gap:** the difference AIS − RAISE

each of size 1000. As with RBMs, we ran RAISE on 100 random test examples and used the DBM log unnormalized probability, $\log f(\mathbf{v})$, as a control variate. To obtain estimates of the DBM unnormalized probability $f(\mathbf{v}) = \sum_{\mathbf{h}_1,\mathbf{h}_2} f(\mathbf{v},\mathbf{h}_1,\mathbf{h}_2)$ we used simple importance sampling $f(\mathbf{v}) = \mathbb{E}_q\left(\frac{f(\mathbf{v},\mathbf{h}_2)}{q(\mathbf{h}_2 \mid \mathbf{v})}\right)$ with 500 samples, where the proposal distribution q was the mean-field approximation to the conditional distribution $p(\mathbf{h}_2 \mid \mathbf{v})$. The term $f(\mathbf{v},\mathbf{h}_2)$ was computed by summing out $\mathbf{h}_1$ analytically, which is efficient because the conditional distribution factorizes.[5]

We compared the RAISE estimates to those obtained using AIS. All results for $K = 100{,}000$ are shown in Table 2, and the estimates for the MNIST DBN are plotted as a function of K in Figure 7. All estimates for the MNIST DBM with $K = 100{,}000$ agreed quite closely, which is a piece of evidence in favor of the accuracy of the estimates. Furthermore, RAISE provided conservative estimates of log-probabilities for small K, in contrast with AIS, which gave overly optimistic estimates. For the Omniglot DBM, RAISE overestimated the DBM log-probabilities by at least 6 nats, implying that the annealing model fit the data distribution better than the DBM, analogously to the case of the mnistCD1-20 RBM discussed in Section 5.1. This shows that RAISE does not completely eliminate the possibility of overestimating an MRF's test log-probabilities.

5.3 Deep Belief Networks

In our final set of experiments, we used RAISE to estimate the average test log-probabilities of DBNs trained on

[5]Previous work (*e.g.* Salakhutdinov and Hinton, 2009) estimated $\log f(\mathbf{v})$ using the mean-field lower bound. We found importance sampling to give more accurate results in the context of AIS. However, it made less difference for RAISE, where the log unnormalized probabilities are merely used as a control variate.

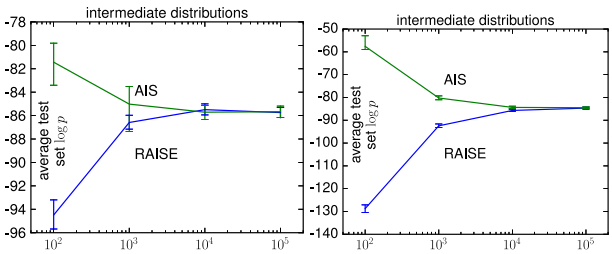


Figure 7: Average test log-probability estimates for MNIST models as a function of K. **Left:** the DBM. **Right:** the DBN.

MNIST and Omniglot. The MNIST DBN had two hidden layers of size 500 and 2000, and the Omniglot DBN had two hidden layers each of size 1000. For the initial distribution p_0 we used the uniform distribution, as the DBR distribution is not defined for DBNs. To obtain estimates of DBN unnormalized probabilities $f(\mathbf{v}) = \sum_{\mathbf{h}_1} p(\mathbf{v} \mid \mathbf{h}_1) f(\mathbf{h}_1)$ we used importance sampling $f(\mathbf{v}) = \mathbb{E}_q\left(\frac{p(\mathbf{v} \mid \mathbf{h}_1) f(\mathbf{h}_1)}{q(\mathbf{h}_1 \mid \mathbf{v})}\right)$ with 500 samples, where q was the DBN recognition distribution (Hinton et al., 2006).

All results for $K = 100{,}000$ are shown in Table 2, and Figure 7 shows the estimates for the MNIST DBN as a function of K. For both DBNs, RAISE and AIS agreed to within 1 nat for $K = 100{,}000$, and RAISE gave conservative log-probability estimates for all values of K.

5.4 Summary

Between our RBM, DBM, and DBN experiments, we compared 10 different models using both uniform and data base rate initial distributions. In all but two cases (the mnistCD1-20 RBM and the Omniglot DBN), RAISE gave estimates at or below the smallest log-probability estimates produced by AIS, suggesting that RAISE typically gives conservative estimates. Furthermore, in all but one case (the Omniglot DBM), the final RAISE estimate agreed with the lowest AIS estimate to within 1 nat, suggesting that it is typically accurate.

6 Conclusion

In this paper, we presented RAISE, a stochastic lower bound on the log-likelihood of an approximation to an MRF model. Our experimental results show that RAISE typically produces accurate, yet conservative, estimates of log-probabilities for RBMs, DBMs, and DBNs. More importantly, by using RAISE and AIS in conjunction, one can judge the accuracy of one's results by measuring the agreement of the two estiatmators.

Acknowledgements

This research was supported by NSERC, Google, and Samsung.

References

Gundula Behrens, Nial Friel, and Merrilee Hurn. Tuning tempered transitions. *Statistics and Computing*, 22:65–78, 2012.

Y. Bengio, L. Yao, and K. Cho. Bounding the test log-likelihood of generative models. arXiv:1311.6184, 2013.

Ben Calderhead and Mark Girolami. Estimating Bayes factors via thermodynamic integration and population MCMC. *Computational Statistics and Data Analysis*, 53(12):4028–4045, 2009.

P. del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Methodology)*, 68(3):411–436, 2006.

V. Gogate, B. Bidyuk, and R. Dechter. Studies in lower bounding probability of evidence using the Markov inequality. In *Conference on Uncertainty in AI*, 2007.

K. Gregor, I. Danihelka, A. Mnih, C. Blundell, and D. Wierstra. Deep autoregressive networks. In *Int'l Conf. on Machine Learning*, 2014.

R. B. Grosse, C. J. Maddison, and R. Salakhutdinov. Annealing between distributions by averaging moments. In *Neural Information Processing Systems*, 2013.

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

Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 18(7):1527–1554, 2006.

Christopher Jarzynski. Equilibrium free-energy differences from nonequilibrium measurements: A master-equation approach. *Physical Review E*, 56:5018–5035, 1997.

Brenden M Lake, Ruslan Salakhutdinov, and Josh Tenenbaum. One-shot learning by inverting a compositional causal process. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2526–2534. Curran Associates, Inc., 2013.

H. Larochelle and I. Murray. The neural autoregressive distribution estimator. In *Artificial Intelligence and Statistics*, 2011.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

A. Mnih and K. Gregor. Neural variational inference and learning in belief networks. In *Int'l Conf. on Machine Learning*, 2014.

R. M. Neal. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, 2001.

Radford Neal. Sampling from multimodal distributions using tempered transitions. *Statistics and Computing*, 6: 353–366, 1996.

Radford M. Neal. Connectionist learning of belief networks. *Artificial Intelligence*, 1992.

H. Poon and P. Domingos. Sum-product networks: a new deep architecture. In *Uncertainty in Artificial Intelligence*, 2011.

R. Ranganath, S. Gerrish, and D. M. Blei. Black box variational inference. In *Artificial Intelligence and Statistics*, 2014.

S. M. Ross. *Simulation*. Academic Press, 2006.

Ruslan Salakhutdinov and Geoffrey E. Hinton. Deep Boltzmann machines. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2009.

Ruslan Salakhutdinov and Ian Murray. On the quantitative analysis of deep belief networks. In *Int'l Conf. on Machine Learning*, pages 6424–6429, 2008.

P. Smolensky. Information processing in dynamical systems: foundations of harmony theory. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. MIT Press, 1986.

Lucas Theis, Sebastian Gerwinn, Fabian Sinz, and Matthias Bethge. In all likelihood, deep belief is not enough. *Journal of Machine Learning Research*, 12:3071–3096, 2011.

Tijmen Tieleman. Training restricted Boltzmann machines using approximations to the likelihood gradient. In *Intl. Conf. on Machine Learning*, 2008.