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-

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 \( v = (v_1, \ldots, v_{N_v}) \) and hidden units \( h = (h_1, \ldots, h_{N_h}) \). 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 \((v, h)\) can be written as \( f(v, h) / Z \), where

\[
f(v, h) = \exp (a^\top v + b^\top h + v^\top W h), \tag{1}
\]

and \( a, b, \) and \( W \) 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 \( \{v^{(1)}_t, \ldots, v^{(M_{tr})}_t\} \). 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 \( \{v^{(1)}_{test}, \ldots, v^{(M_{test})}_{test}\} \).

It remains challenging to evaluate the probability \( p(v) = f(v) / Z \) of an example. The unnormalized probability \( f(v) = \sum_h f(v, h) \) can be computed exactly since the conditional distribution factorizes over the \( h_j \). However, \( Z \) 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 \( h_1 \) and \( h_2 \) can be written as \( f(v, h_1, h_2) / Z \), where

\[
f(v, h_1, h_2) = \exp \left( a^\top v + b_1^\top h_1 + b_2^\top h_2 + v^\top W_1 h_1 + h_1^\top W_2 h_2 \right). \tag{2}
\]

DBMs can be evaluated similarly to RBMs. The main difference is that the unnormalized probability \( f(v) = \sum_{h_1, h_2} f(v, h_1, h_2) \) 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 \( h_1 \) and \( h_2 \) is defined as the probability distribution

\[
p(v, h_1, h_2) = p_2(h_1, h_2)p_1(v | h_1), \tag{3}
\]

where \( p_2(h_1, h_2) \) is the probability distribution of an RBM, and \( p_1(v | h_1) \) is a product of independent logistic units. The unnormalized probability \( f(v) = \sum_{h_1, h_2} p_1(v | h_1)f_2(h_1, h_2) \) cannot be computed analytically, but can be approximated using importance sampling or a variational lower bound that utilizes a recognition distribution \( q(h_1 | v) \) approximating the posterior \( p(h_1 | v) \) (Hinton et al., 2006).

2.2 Partition Function Estimation

Often we have a probability distribution \( p_{tgt}(x) = f_{tgt}(x) / Z_{tgt} \) (which we call the target distribution) defined on a space \( \mathcal{X} \), where \( f_{tgt}(x) \) can be computed efficiently for a given \( x \in \mathcal{X} \), and \( Z_{tgt} \) 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 \( f_{tgt}(x) \) denotes the product of all potentials, and \( Z_{tgt} = \sum_x f_{tgt}(x) \) is the partition function of the graphical model.

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

\[
f_{tgt}(h) = p(h)p(v_{test} | h) \text{ and } Z_{tgt} = p(v_{test}). \tag{4}
\]

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 \( \hat{Z}_{tgt} \) of the partition function, i.e. \( \mathbb{E} [\hat{Z}_{tgt}] = Z_{tgt} \). Jensen’s Inequality shows that such an estimator tends to underestimate the log partition function on average:

\[
\mathbb{E} [\log \hat{Z}_{tgt}] \leq \log \mathbb{E} [\hat{Z}_{tgt}] = \log Z_{tgt}. \tag{5}
\]


Algorithm 1 Annealed Importance Sampling

\[
\text{for } i = 1 \text{ to } M \text{ do } \\
\quad x_0 \leftarrow \text{sample from } p_{\text{init}}(x)/Z_{\text{init}} \\
\quad \text{for } k = 1 \text{ to } K \text{ do } \\
\quad\quad w^{(i)} \leftarrow Z_{\text{init}} \\
\quad\quad x_k \leftarrow \text{sample from } T_k \left( x_{k-1} \right) \\
\quad \text{end for } \\
\text{end for } \\
\text{return } \hat{Z}_{\text{tgt}} = \sum_{i=1}^{M} w^{(i)}/M \\
\]

In addition, Markov’s inequality shows that it is unlikely to substantially overestimate \( \log Z_{\text{tgt}} \):

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

For these reasons, we will refer to the estimator as a stochastic lower bound on \( \log Z_{\text{tgt}} \).

In the MRF situation, \( Z_{\text{tgt}} \) 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 \( Z_{\text{tgt}} \) by gradually changing, or “annealing,” a distribution. In particular, one must specify a sequence of \( K + 1 \) intermediate distributions \( p_k(x) = f_k(x)/Z_k \) for \( k = 0, \ldots, K \), where \( p_{\text{init}}(x) = p_0(x) \) is a tractable initial distribution, and \( p_{\text{tgt}}(x) = p_K(x) \) 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 \( \hat{Z}_{\text{tgt}} \) of \( Z_{\text{tgt}} \). 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

\[
q_{\text{fwd}}(x_0; K-1) = p_0(x_0) \prod_{k=1}^{K-1} T_k(x_k | x_{k-1}) \\
\]

\[
f_{\text{rev}}(x_0; K-1) = f_{\text{tgt}}(x_K-1) \prod_{k=1}^{K-1} \tilde{T}_k(x_k-1 | x_k), \tag{8}
\]

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

\[
p_\beta(x) = f_\beta(x)/Z(\beta) = f_{\text{init}}(x)^{1-\beta} f_{\text{tgt}}(x)\beta/Z(\beta), \tag{9}
\]

where \( 0 = \beta_0 < \beta_1 < \ldots < \beta_K = 1 \) defines the annealing schedule. Commonly, \( f_{\text{init}} \) is the uniform distribution, and (9) reduces to \( p_\beta(x) = f_{\text{tgt}}(x)^\beta/Z(\beta) \). This motivates the term “annealing”, and \( \beta \) resembles an inverse temperature parameter. As in simulated annealing, the “hotter” distributions often allow faster mixing between modes which are isolated in \( p_{\text{tgt}} \). 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 \( \beta \) 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)

\[
\begin{align*}
\text{for } i = 1 \text{ to } M \text{ do} & \quad \text{ } \\
& \quad h_K \leftarrow \text{sample from } p_{\text{tgt}}(h \mid v_{\text{test}}) \\
& \quad w(i) \leftarrow f_{\text{tgt}}(v_{\text{test}}) / Z_0 \\
\text{for } k = K - 1 \text{ to } 0 \text{ do} & \quad \text{ } \\
& \quad x_k \leftarrow \text{sample from } \hat{T}_k (\cdot \mid x_{k+1}) \\
& \quad w(i) \leftarrow w(i) f_k(x_k) / \hat{f}_{k+1}(x_k) \\
\end{align*}
\]

end for

end for

return \( \hat{p}_{\text{ann}}(v_{\text{test}}) = \sum_{i=1}^M w(i) / M \)

### 3.1 Case of Tractable Posterior

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

\[
p_{\text{fwd}}(x_0:K) = p_0(x_0) \prod_{k=1}^K \hat{T}_k (x_k \mid x_{k-1}).
\]

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

\[
p_{\text{ann}}(v_K) = \sum_{x_0, x_{K-1}, h_K} p_{\text{fwd}}(x_0:K, h_K, v_K).
\]

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

\[
q_{\text{rev}}(x_0:K-1, h_K \mid v_{\text{test}}) = p_{\text{tgt}}(h_K \mid v_{\text{test}}) \prod_{k=1}^K \hat{T}_k (x_{k-1} \mid x_k),
\]

where we identify \( v_k = v_{\text{test}} \), and \( \hat{T}_k (x \mid x') = T_k (x \mid x') p_k(x') / p_k(x) \) is the reverse transition operator for \( T_k \). Then we obtain the following identities:

\[
p_{\text{ann}}(v_{\text{test}}) = \mathbb{E}_{q_{\text{rev}}} \left[ \frac{p_{\text{fwd}}(x_0:K-1, h_K, v_{\text{test}})}{q_{\text{rev}}(x_0:K-1, h_K \mid v_{\text{test}})} \right]
\]

\[
= \mathbb{E}_{q_{\text{rev}}} \left[ \frac{p_0(x_0)}{p_{\text{tgt}}(h_K \mid v_{\text{test}})} \prod_{k=1}^K \frac{T_k (x_k \mid x_{k-1})}{T_k (x_k \mid x_{k-1})} \right]
\]

\[
= \mathbb{E}_{q_{\text{rev}}} \left[ \frac{p_0(x_0)}{p_{\text{tgt}}(h_K \mid v_{\text{test}})} \prod_{k=1}^K \frac{f_k(x_k)}{f_k(x_k)} \right]
\]

\[
= \mathbb{E}_{q_{\text{rev}}} \left[ f_{\text{rev}}(v_{\text{test}}) \frac{1}{Z_0} \prod_{k=0}^{K-1} \frac{f_k(x_k)}{f_{k+1}(x_k)} \right]
\]

\[
\triangleq \mathbb{E}_{q_{\text{rev}}} [ w ].
\]

This yields the following algorithm: generate \( M \) samples from \( q_{\text{rev}} \), 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 \( p_{\text{ann}}(v_{\text{test}}) \). Therefore, following the discussion of Section 2.2, \( \log w \) is a stochastic lower bound on \( \log p_{\text{ann}}(v_{\text{test}}) \). Furthermore, since \( p_{\text{ann}} \) converges to \( p_{\text{tgt}} \) in probability as \( K \to \infty \) (Neal, 2001), we heuristically expect RAISE to yield a conservative estimate of \( \log p_{\text{tgt}}(v_{\text{test}}) \). This is not strictly guaranteed, however; RAISE may overestimate \( \log p_{\text{tgt}}(v_{\text{test}}) \) for finite \( K \) if \( p_{\text{ann}}(v_{\text{test}}) > p_{\text{tgt}}(v_{\text{test}}) \), 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 \( p_{\text{tgt}}(h \mid v_{\text{test}}) \), 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 \( v \) agrees with that of \( p_{\text{ann}} \) in (11). We make the further (weak) assumption that \( p_0(h \mid v) \) is tractable. Let \( T^{(v)}_k \) denote an MCMC transition operator which preserves \( p_k(v, h) \), but does not change \( v \). For example, it may cycle through Gibbs updates to all variables except \( v \). The forward chain then has the following distribution:

\[
p_{\text{fwd}}(x_0:K, h_0^{K-1} \mid h_K) = p_0(x_0) \prod_{k=1}^K T_k (x_k \mid x_{k-1}) \]

\[
\prod_{k=0}^{K-1} T^{(v)}_k (h_k' \mid h_{k+1}') \]

where we identify \( h_K' = h_K \). The reverse distribution is given by:

\[
q_{\text{rev}}(x_0:K-1, h_K, h_0^{K-1} \mid v_{\text{test}}) = \]

\[
p_0(h_0' \mid v_{\text{test}}) \prod_{k=0}^{K-1} \hat{T}^{(v)}_k (h_k' \mid h_{k+1}' \mid h_k') \prod_{k=1}^K T_k (x_{k-1} \mid x_k).
\]
Yuri Burda¹, Roger B. Grosse¹, Ruslan Salakhutdinov

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
  \( h_0^{(i)} \leftarrow \text{sample from } p_0(h | v_{\text{test}}) \)
  for \( k = 1 \) to \( K \) do
    \( h_k^{(i)} \leftarrow \text{sample from } \tilde{T}_k(v_{\text{test}}) \)
    \( w_k^{(i)} \leftarrow w_k^{(i)} \cdot f_k(h_k^{(i)}, v_{\text{test}}) \)
  end for
  for \( k = K - 1 \) to 0 do
    \( x_k \leftarrow \text{sample from } \tilde{T}_k(x_k) \)
    \( w_k^{(i)} \leftarrow w_k^{(i)} \cdot f_k(x_k) \)
  end for
end for
return \( \hat{p}_{\text{ann}}(v_{\text{test}}) = \sum_{i=1}^{M} w^{(i)}/M \)

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

\[
\hat{w} = \frac{p_{\text{rewd}}(x_{0:K-1}, h_K, v_{\text{test}}, h'_{0:K-1})}{q_{\text{rev}}(x_{0:K-1}, h_K, h'_{0:K-1} | v_{\text{test}})}
\]

\[
= p_0(v_{\text{test}}) \prod_{k=0}^{K-1} f_k(x_k) \prod_{k=1}^{K} f_k(h'_k, v_{\text{test}})
\]

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 unites 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 \( p_{\text{ann}} \) 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.

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, 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

\[\text{This effect can be counterintuitively large due to different complexities of different categories; e.g., for the mnistCD25-500 RBM, the average log-likelihood of handwritten digits “1” was 56.6 nats higher than the average log-likelihood of digits “8”.}\]
and highly correlated with \( Y \), then for any scalar \( \alpha \),
\[
\frac{1}{n} \sum_{i=1}^{n} (Y_i - \alpha X_i) + \frac{\alpha}{N} \sum_{i=1}^{N} X_i
\]
is an unbiased estimator of \( E[Y] \) with variance
\[
\frac{\text{Var}[Y - \alpha X]}{n} + \frac{\alpha^2 \text{Var}[X]}{N} + 2\alpha \frac{\text{Cov}[Y - \alpha X, X]}{n}.
\]

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 \( N \) be much larger than \( n \); we set \( n = 100 \) and \( N \) 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.

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 \( \beta \). We tested two choices of initial distribution \( p_{\text{ini}} \): 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.

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, \( \log f(v) \), 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).\footnote{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 \( K = 100,000 \).}

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

```plaintext
<table>
<thead>
<tr>
<th>Model</th>
<th>exact</th>
<th>CSL</th>
<th>RAISE</th>
<th>AIS</th>
<th>gap</th>
<th>RAISE</th>
<th>AIS</th>
<th>gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>mnistCD1-20</td>
<td>-164.50</td>
<td>-152.13</td>
<td>-150.58</td>
<td>-150.04</td>
<td>0.54</td>
<td>-150.17</td>
<td>-150.10</td>
<td>0.07</td>
</tr>
<tr>
<td>mnistPCD-20</td>
<td>-150.11</td>
<td>-152.13</td>
<td>-150.58</td>
<td>-150.04</td>
<td>0.54</td>
<td>-150.17</td>
<td>-150.10</td>
<td>0.07</td>
</tr>
<tr>
<td>mnistCD1-500</td>
<td>-566.91</td>
<td>-150.78</td>
<td>-106.52</td>
<td>44.26</td>
<td>-124.77</td>
<td>-124.09</td>
<td>0.68</td>
<td></td>
</tr>
<tr>
<td>mnistPCD-500</td>
<td>-138.76</td>
<td>-101.07</td>
<td>-99.99</td>
<td>1.08</td>
<td>-101.26</td>
<td>-101.28</td>
<td>-0.02</td>
<td></td>
</tr>
<tr>
<td>mnistCD25-500</td>
<td>-145.26</td>
<td>-88.51</td>
<td>-86.42</td>
<td>2.09</td>
<td>-86.39</td>
<td>-86.35</td>
<td>0.04</td>
<td></td>
</tr>
<tr>
<td>omniPCD-1000</td>
<td>-144.25</td>
<td>-100.47</td>
<td>-100.45</td>
<td>0.02</td>
<td>-100.46</td>
<td>-100.46</td>
<td>0.00</td>
<td></td>
</tr>
</tbody>
</table>
```

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

AIS estimates
Estimate of average log probability on the test dataset

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

<table>
<thead>
<tr>
<th>Model</th>
<th>RAISE</th>
<th>uniform</th>
<th>AIS gap</th>
<th>RAISE</th>
<th>AIS</th>
<th>gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNIST DBM</td>
<td>-85.69</td>
<td>-85.72</td>
<td>-0.03</td>
<td>-85.74</td>
<td>-85.67</td>
<td>0.07</td>
</tr>
<tr>
<td>Omniglot DBM</td>
<td>-104.48</td>
<td>-110.86</td>
<td>-6.38</td>
<td>-102.64</td>
<td>-103.27</td>
<td>-0.63</td>
</tr>
<tr>
<td>MNIST DBN</td>
<td>-84.67</td>
<td>-84.49</td>
<td>0.18</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Omniglot DBN</td>
<td>-100.78</td>
<td>-100.45</td>
<td>0.33</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

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 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 $f(v) = \sum_{h_1, h_2} p(v | h_1) f(h_1)$ we used importance sampling $f(v) = \mathbb{E}_q \left( \frac{f(v, h_2)}{q(h_2 | v)} \right)$ with 500 samples, where $q$ was the proposal distribution $q$ was the mean-field approximation to the conditional distribution $p(h_2 | v)$. The term $f(v, h_2)$ was computed by summing out $h_1$ analytically, which is efficient because the conditional distribution factorizes.$^5$

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 estimators.

Acknowledgements

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


