Department of Computer Science
University of Toronto
`http://learning.cs.toronto.edu`

6 King's College Rd, Toronto
M5S 3G4, Canada
fax: +1 416 978 1455

June 26, 2008

UTML TR 2008–002

Learning and Evaluating Boltzmann Machines

Ruslan Salakhutdinov
Department of Computer Science, University of Toronto

Abstract

We provide a brief overview of the variational framework for obtaining deterministic approximations or upper bounds for the log-partition function. We also review some of the Monte Carlo based methods for estimating partition functions of arbitrary Markov Random Fields. We then develop an annealed importance sampling (AIS) procedure for estimating partition functions of restricted Boltzmann machines (RBM's), semi-restricted Boltzmann machines (SRBM's), and Boltzmann machines (BM's). Our empirical results indicate that the AIS procedure provides much better estimates of the partition function than some of the popular variational-based methods. Finally, we develop a new learning algorithm for training general Boltzmann machines and show that it can be successfully applied to learning good generative models.

Learning and Evaluating Boltzmann Machines

Ruslan Salakhutdinov
Department of Computer Science, University of Toronto

1 Introduction

Undirected graphical models, also known as Markov random fields (MRF's), or general Boltzmann machines, provide a powerful tool for representing dependency structure between random variables. They have successfully been used in various application domains, including machine learning, computer vision, and statistical physics. The major limitation of undirected graphical models is the need to compute the partition function, whose role is to normalize the joint probability distribution over the set of random variables. In addition, the derivatives of the partition function are needed for parameter learning. For many problems, however, the exact calculation of the partition function or its derivatives is intractable, because it requires enumeration over an exponential number of terms.

There has been extensive research on obtaining deterministic approximations [30, 31] or deterministic upper bounds [26, 28, 5] on the log-partition function of an arbitrary discrete MRF. These methods take a variational view of estimating the log-partition function and rely critically on approximating the entropy of the undirected graphical model. Variational methods have become very popular, since they typically scale well to large applications.

There have also been many developments in the use of Monte Carlo methods for estimating the partition function, including Annealed Importance Sampling (AIS) [15], Bridged Sampling [11], Linked Importance Sampling [16], Nested Sampling [21], sequential Monte Carlo [12], and many others [14]. These methods are perceived to be computationally very demanding, so in practice, they are rarely applied to large-scale problems.

In the next section, we will describe the variational view of approximating the partition function and will review various methods that fall under this framework. For more thorough discussions on these topics refer to [27]. In section 3 we will review some Monte Carlo methods of estimating partition functions. In section 4 we will provide a brief overview of restricted Boltzmann machines (RBM's), semi-restricted Boltzmann machines (SRBM's) and Boltzmann machines (BM's), and will present a new learning algorithm for general Boltzmann Machines. We will further show how a stochastic method, Annealed Importance Sampling (AIS), can be used to efficiently estimate partition functions of these models [20]. In the experimental results section we will show that our new learning algorithm can be successfully applied to learning a good generative model of MNIST digits. We will also compare AIS to variational methods for estimating partition functions of large Boltzmann Machines, carefully trained on real data.

2 Variational Framework

2.1 Notation

Let $\mathbf{x} \in \mathcal{X}^K$ be a random vector on K variables, where each x_k takes on the values in some discrete alphabet $\mathcal{A}^m = \{0, 1, ..., m-1\}$. Let $T(\mathbf{x}) = \{t_d(\mathbf{x})\}_{d=1}^D$ be a D-dimensional vector of sufficient

statistics or potential functions, where $t_d : \mathcal{X}^K \rightarrow R$, and $\theta \in R^D$ is a vector of canonical parameters. The exponential family associated with sufficient statistics T consists of the following parametrized set of probability distributions:

$$p(\mathbf{x}; \theta) = \frac{1}{Z(\theta)} \exp\left(\theta^\top T(\mathbf{x})\right) \tag{1}$$

$$Z(\theta) = \sum_{\mathbf{x}} \exp\left(\theta^\top T(\mathbf{x})\right), \tag{2}$$

where $Z(\theta)$ is known as the partition function.

An undirected graphical model $G = (V, E)$ contains a set of vertexes V that represent random variables, and a set of undirected edges E, that represent dependencies between those random variables. Throughout this paper, we will concentrate on *pairwise* MRF's. For example, consider the following binary pairwise MRF, also known as an Ising model:

$$p(\mathbf{x}; \theta) = \frac{1}{Z(\theta)} \exp\left(\sum_{(i,j) \in E} \theta_{ij} x_i x_j + \sum_{i \in V} \theta_i x_i\right) = \frac{1}{Z(\theta)} \exp\left(\theta^\top T(\mathbf{x})\right), \tag{3}$$

where x_i is a Bernoulli random variable associated with vertex $i \in V$. This exponential representation of the Ising model is minimal. In general, we will use an overcomplete exponential representation. Let $\mathrm{I}\{x_i = s\}$ be an indicator function that is equal to 1 if $x_i = s$ and zero otherwise. For a pairwise MRF, we have:

$$p(\mathbf{x}; \theta) = \frac{1}{Z(\theta)} \exp\left(\sum_{(i,j) \in E} \sum_{s,t} \theta_{ij,st} \mathrm{I}\{x_i = s, x_j = t\} + \sum_{i \in V} \sum_s \theta_{i,s} \mathrm{I}\{x_i = s\}\right),$$

so the sufficient statistic is a collection of indicator functions. We will use the following convenient notation for the marginal probabilities: $\mu_{i;s} = p(x_i = s; \theta)$, and $\mu_{ij;st} = p(x_i = s, x_j = t; \theta)$. For the Ising model, we simply denote $\mu_i = p(x_i = 1; \theta)$, and $\mu_{ij} = p(x_i = 1, x_j = 1; \theta)$.

2.2 Variational Framework

Following [26], the log partition function for any given parameter vector θ is given by:

$$\log Z(\theta) = \sup_{\mu \in \mathcal{M}} (\theta^\top \mu - A^*(\mu)), \tag{4}$$

where $\mathcal{M}$ is the set of mean parameters or marginals, known as the Marginal Polytope:

$$\mathcal{M} = \{\mu \in R^D \mid \exists p(\cdot) \text{ s.t. } \mu = E_p[T(\mathbf{x})]\}.$$

$A^*(\mu)$ is known as Fenchel-Legendre conjugate of $\log Z(\theta)$ and is given by:

$$A^*(\mu) = \begin{cases} -\mathcal{H}(\mu) & \text{if } \mu \in \mathcal{M} \\ +\infty & \text{otherwise} \end{cases},$$

where $\mathcal{H}(\mu)$ is the maximum entropy amongst all distributions consistent with marginals μ. Moreover, the supremum is achieved at the vector $\mu_\theta^* = E_{p(x;\theta)}[T(\mathbf{x})]$.

First, note that this variational problem is convex, since $A^*(\mu)$ is convex and the set $\mathcal{M}$ is convex. Second, it is quite simple to derive Eq. 4 (see [17, 27, 22]). Consider any distribution in the exponential family $p(\mathbf{x}; \eta)$. Using Jensen's inequality we have:

$$\log Z(\theta) = \log \sum_{\mathbf{x}} \exp\{\theta^\top T(\mathbf{x})\} = \log \sum_{\mathbf{x}} \frac{p(\mathbf{x}; \eta)}{p(\mathbf{x}; \eta)} \exp\{\theta^\top T(\mathbf{x})\} \geq \tag{5}$$

$$\geq \sum_{\mathbf{x}} p(\mathbf{x}; \eta) \log \frac{\exp\{\theta^\top T(\mathbf{x})\}}{p(\mathbf{x}; \eta)} = \sum_{\mathbf{x}} p(\mathbf{x}; \eta)\{\theta^\top T(\mathbf{x})\} + \mathcal{H}(p(\mathbf{x}; \eta)) =$$

$$= \theta^\top \mu_\eta + \mathcal{H}(p(\mathbf{x}; \eta)) = \theta^\top \mu_\eta + \mathcal{H}(\mu_\eta),$$

where $\mu_\eta = \mathrm{E}_{p(\mathbf{x};\eta)}[T(\mathbf{x})]$, and $\mathcal{H}(\mu_\eta)$ is the maximum entropy of all distributions consistent with the marginal vector μ_η. The last equality holds since $p(\mathbf{x};\eta)$ is the maximum entropy distribution consistent with μ_η, as $p(\mathbf{x};\eta)$ is in the exponential family. Clearly, the above inequality holds for any distribution Q (provided the marginals $\mu_Q \in \mathcal{M}$), and the set $\mathcal{M}$ is the set of all marginals that can be realized under some distribution (see [27, 22]). We therefore have:

$$\log Z(\theta) \geq \sup_{\mu \in \mathcal{M}} (\theta^\top \mu + \mathcal{H}(\mu)). \tag{6}$$

The bound in Eq. 5 becomes tight if and only if $p(\mathbf{x};\eta) = p(\mathbf{x};\theta)$. In this case we recover Eq. 4. Moreover, it is now clear that the supremum is attained at the vector μ_θ^*, which represents the marginals of the distribution $p(\mathbf{x};\theta)$.

There are two main difficulties associated with the above variational optimization problem. First, the marginal polytope $\mathcal{M}$ does not have an explicit representation, except for simple models, such as the tree structured graphs. One way to address this problem is to restrict optimization problem to a tractable subset $\mathcal{M}_{tract} \subseteq \mathcal{M}$. For example, optimization could be carried over a subset of "simpler" distributions, belonging to the exponential family, such as fully factorized distributions. Alternatively, one could consider the outer bound $\mathcal{M}_{outer} \supseteq \mathcal{M}$, by relaxing the set of necessary constraints that any point $\mu \in \mathcal{M}$ must satisfy.

The second major challenge comes from evaluating the entropy function $\mathcal{H}(\mu)$ – it too does not have an explicit form, with the exception of the tree graphs. A common approach to this problem is to approximate the entropy term in such a way that this approximation becomes exact on a singly-connected graph.

2.3 Mean-Field

The goal of the mean-field approach is to find a distribution Q from the class of analytically tractable distributions that best approximates the original distribution P in terms of $KL(Q||P)$. It was shown in [27] that the mean-field theory is based on variational principle of Eq. 4. Consider a set of mean parameters $\mathcal{M}_{tract} \subseteq \mathcal{M}$ that are achieved by tractable distributions for which the entropy term can be calculated exactly. In this case for any $\mu \in \mathcal{M}_{tract}$, the lower bound of Eq. 5 can be calculated exactly. The mean-field methods attempt to find the best approximation μ_{MF} which maximizes this lower bound.

As an example, consider our Ising model and let us choose a fully factorized distribution in order to approximate the original distribution. In this case we define:

$$\mathcal{M}_{tract} = \{(\mu_i, \mu_{ij}) | \mu_{ij} = \mu_i \mu_j, 0 \leq \mu_i \leq 1\}. \tag{7}$$

The entropy term of this factorized distribution is easy to compute. The mean-field objective is to maximize:

$$
\begin{aligned}
\log Z(\theta) \ \geq\ & \sup_{\mu \in \mathcal{M}_{tract}} \left(\theta^\top \mu + \mathcal{H}(\mu)\right) = \\
& \max_{\mu \in [0,1]} \left(\sum_{(i,j) \in E} \theta_{ij}\mu_i\mu_j + \sum_{i \in V} \theta_i\mu_i - \sum_{i \in V} [\mu_i \log \mu_i + (1 - \mu_i) \log (1 - \mu_i)] \right). \tag{8}
\end{aligned}
$$

The solution will provide us with the lower bound on the log-partition function. This optimization problem is equivalent to minimizing Kullback-Leibler divergence between the tractable distribution and the target distribution [27]. Furthermore, the mean-field objective may not be convex, so the mean-field updates may have multiple solutions.

2.4 Bethe Approximation

A close connection between loopy belief propagation and the Bethe approximation to the log-partition function of a pairwise Markov random field was shown by [30, 31]. Let us consider a tree structured graph. In this case we know that the correct joint probability distribution can be written in terms of single and pair-wise node marginals:

$$p(x; \theta) = \prod_i \mu_{i;x_i} \prod_{i,j} \frac{\mu_{ij;x_i,x_j}}{\mu_{i;x_i} \mu_{j;x_j}}. \tag{9}$$

The entropy of this distribution has a closed-form expression and decomposes into a sum of the single node entropies and the edgewise mutual information terms between two nodes. For graphs with cycles, the entropy in general will not have this simple decomposition. Nevertheless, if we use this decomposition, we obtain the Bethe approximation to the entropy term on a graph with cycles:

$$\mathcal{H}_{Bethe}(\mu) = \sum_{i \in V} \mathcal{H}_i(\mu_i) - \sum_{(i,j) \in E} I_{ij}(\mu_{ij}), \quad \text{where} \tag{10}$$

$$\mathcal{H}_i(\mu_i) = - \sum_s \mu_{i;s} \log \mu_{i;s}, \quad I_{ij}(\mu_{ij}) = \sum_{s,t} \mu_{ij;st} \log \frac{\mu_{ij;st}}{\mu_{i;s}\mu_{j;t}} \tag{11}$$

This approximation is well-defined for any $\mu \in \mathcal{M}$. As mentioned before, it is hard to explicitly characterize the marginal polytope. Let us consider the outer bound $\mathcal{M}_{outer} \supseteq \mathcal{M}$, by relaxing the set of necessary constrains that any point $\mu \in \mathcal{M}$ must satisfy. In particular, consider the following set:

$$\mathcal{M}_{LOCAL} = \{\mu \geq 0| \sum_s \mu_{i;s} = 1, \sum_t \mu_{ij;st} = \mu_{i;s}\}. \tag{12}$$

Clearly, $\mathcal{M}_{LOCAL} \supseteq \mathcal{M}$, since any member of the marginal polytope must also satisfy local consistency constraints. Members of $\mathcal{M}_{LOCAL}$ are referred to as pseudo-marginals since they may not give rise to any valid probability distribution. The Bethe approximation to the log-partition function reduces to solving the following optimization problem:

$$\log Z_{Bethe}(\theta) = \sup_{\mu \in \mathcal{M}_{LOCAL}} (\theta^\top \mu + \mathcal{H}_{Bethe}(\mu)). \tag{13}$$

The stationary points of loopy belief propagation [30, 31] correspond to the stationary points of the above objective function. For singly-connected graphs, the Bethe approximation to the partition function becomes exact. In general, $\mathcal{H}_{Bethe}$ is not concave, so there is no guarantee that loopy belief propagation will find a global optimum. Furthermore, the Bethe approximation provides neither lower nor upper bound on the log-partition function, except for special cases [24].

2.5 Tree-Reweighted Upper Bound

A different way of approximating the entropy term, which results in obtaining an upper bound of the log-partition function, is taken by [26]. The idea is to use a convex outer bound on the marginal polytope and a concave upper bound on the entropy term.

Let us consider a pairwise MRF $G = (V, E)$. We are interested in obtaining an upper bound on the entropy $\mathcal{H}(\mu)$. Removing some of the edges from this MRF can only increase the entropy term, since removing edges corresponds to removing constraints. Consider any spanning tree $T = (V, E(T))$ of the graph G. The entropy of the joint distribution defined on this spanning tree with matched marginals $\mathcal{H}(\mu(T))$ must be larger than the entropy of the original distribution $\mathcal{H}(\mu)$. This bound must also hold for any convex combination of spanning trees.

More formally, let S be a set of spanning trees, ρ be any probability distribution over those spanning trees, and $\rho_{ij} = \sum_{T \in S} \rho(T) \mathrm{I}\{(i,j) \in E(T)\}$ be edge appearance probabilities. We therefore have the following upper bound on the entropy term:

$$
\begin{aligned}
\mathcal{H}(\mu) \leq E_\rho(H(\mu(T)) &= \sum_{T \in S} \rho(T) [\sum_{i \in V} \mathcal{H}_i(\mu_i) - \sum_{(i,j) \in E(T)} I_{ij}(\mu_{ij})] = \\
&= \sum_{i \in V} \mathcal{H}_i(\mu_i) - \sum_{(i,j) \in E} \rho_{ij} I_{ij}(\mu_{ij}) = \tilde{\mathcal{H}}(\mu).
\end{aligned}
\tag{14}
$$

Using variational approach of Eq. 4, we obtain the upper bound on the log-partition function:

$$
\log Z(\theta) = \sup_{\mu \in \mathcal{M}} (\theta^\top \mu + \mathcal{H}(\mu)) \leq \sup_{\mu \in \mathcal{M}} (\theta^\top \mu + \tilde{\mathcal{H}}(\mu)) \leq \sup_{\mu \in \mathcal{M}_{LOCAL}} (\theta^\top \mu + \tilde{\mathcal{H}}(\mu)).
\tag{15}
$$

The nice property of this optimization problem is that it is convex. Indeed, the cost function is concave, since it consists of a linear term plus a concave term. Moreover, the set $\mathcal{M}_{LOCAL}$ is convex. For any fixed ρ, Wainright et.al. [26] further derive a tree-reweighted sum-product (TRW) algorithm that efficiently solves the above optimization problem.

2.6   Other approaches

There have been numerous other approaches that either attempt to better approximate the entropy term or provide tighter outer bounds on the marginal polytope.

In particular, [28] propose to use a semi-definite outer-bound on the marginal polytope, and a Gaussian bound on the discrete entropy. The key intuition here is that the differential entropy of any continuous random vector is upper bounded by a Gaussian distribution with matched covariance.

A different approach to approximating the entropy term was suggested by [5]. Using the chain rule for the entropy we can write:

$$
\mathcal{H}(x_1, ..., x_N) = \mathcal{H}(x_1) \mathcal{H}(x_2 | x_1) ... \mathcal{H}(x_n | x_1, x_2, .., x_{N-1}).
\tag{16}
$$

Removing some of the conditioning variables cannot decrease the entropy, thus allowing us to obtain the upper bound. Note that there is a close connection between this formulation and a tree-based upper bound. Indeed, for any given spanning tree T we can calculate its entropy term as:

$$
\mathcal{H} = \sum_{i=1}^{N} \mathcal{H}(x_i | Pa_T(x_i))
\tag{17}
$$

where Pa_T are the parents of the variable x_i. This is just an instance of the bound derived from the chain rule of Eq. 16.

Recently, [23, 22] proposed a cutting plane algorithm that solves for an upper bound of the log-partition function of Eq. 4 by incorporating a set of valid constraints that are violated by the current pseudo-marginals. The idea is to use cycle inequalities to obtain a tighter outer bound on the marginal polytope. Consider a binary MRF. Suppose we start at node i with $x_i = 0$, traverse the cycle, and return back to node i. Clearly, the number of times an assignment changes, can only be even. Mathematically, this constraint is written as follows: for any cycle C and any $F \subseteq C$ such that $|F|$ is odd we have: $\sum_{(i,j) \in C \backslash F} \mathrm{I}\{x_i \neq x_j\} + \sum_{(i,j) \in F} \mathrm{I}\{x_i = x_j\} \geq 1$. Since this is true for any valid assignment, it also holds in expectation, providing us with the cycle inequalities constraints:

$$
\sum_{(i,j) \in C \backslash F} (\mu_{ij;01} + \mu_{ij;10}) + \sum_{(i,j) \in F} (\mu_{ij;00} + \mu_{ij;11}) \geq 1
\tag{18}
$$

The cutting plane algorithm starts with the loose outer bound $\mathcal{M}_{LOCAL}$, proceeds by solving for an upper bound of the log-partition function of Eq. 4, finds violated cycle inequalities by the current pseudo-marginals, and adds them as valid constraints, thus providing a tighter outer bound on the marginal polytope.

3 Stochastic Methods for Estimating Partition Functions

Throughout the remaining sections of this paper we will make use of the following canonical form of distributions:

$$p(\mathbf{x};\theta) = \frac{1}{Z(\theta)} \exp\left(-E(x;\theta)\right) \tag{19}$$

where $E(x;\theta)$ is the energy function which depends on parameter vector θ. If we define $E(x;\theta) = -\theta^\top T(\mathbf{x})$, then we recover the canonical form of the exponential family associated with sufficient statistics T, as defined in section 2.1.

3.1 Simple Importance Sampling (SIS)

Suppose we have two distributions defined on some space $\mathcal{X}$ with probability density functions $p_A(\mathbf{x}) = p_A^*(\mathbf{x})/Z_A$ and $p_B(\mathbf{x}) = p_B^*(\mathbf{x})/Z_B$, where $p^*(\cdot)$ denotes the unnormalized probability density. Let Ω_A and Ω_B be the support sets of p_A and p_B respectively. One way of estimating the ratio of normalizing constants is to use a simple importance sampling (SIS) method. We use the following identity, assuming that $\Omega_B \subseteq \Omega_A$, i.e. $p_A(\mathbf{x}) \neq 0$ whenever $p_B(\mathbf{x}) \neq 0$:

$$\frac{Z_B}{Z_A} = \frac{\int p_B^*(\mathbf{x}) \mathrm{d}\mathbf{x}}{Z_A} = \int \frac{p_B^*(\mathbf{x})}{p_A^*(\mathbf{x})} p_A(\mathbf{x}) \mathrm{d}\mathbf{x} = \mathrm{E}_{p_A}\left[\frac{p_B^*(\mathbf{x})}{p_A^*(\mathbf{x})}\right].$$

Assuming we can draw independent samples from p_A, the unbiased estimate of the ratio of partition functions can be obtained by using a simple Monte Carlo approximation:

$$\frac{Z_B}{Z_A} \approx \frac{1}{M} \sum_{i=1}^{M} \frac{p_B^*(\mathbf{x}^{(i)})}{p_A^*(\mathbf{x}^{(i)})} \equiv \frac{1}{M} \sum_{i=1}^{M} w^{(i)} = \hat{r}_{\text{SIS}}, \tag{20}$$

where $\mathbf{x}^{(i)} \sim p_A$. If we choose $p_A(\mathbf{x})$ to be a tractable distribution for which we can compute Z_A analytically, we obtain an unbiased estimate of the partition function Z_B. However, if p_A and p_B are not close enough, the estimator $\hat{r}_{\text{SIS}}$ will be very poor. In high-dimensional spaces, the variance of an estimator $\hat{r}_{\text{SIS}}$ will be very large, or possibly infinite (see [10], chapter 29), unless p_A is a near-perfect approximation to p_B.

3.2 Annealed Importance Sampling (AIS)

Suppose that we can define a sequence of intermediate probability distributions: $p_0, ..., p_K$, with $p_0 = p_A$ and $p_K = p_B$, which satisfy the following conditions:

C1 $p_k(\mathbf{x}) \neq 0$ whenever $p_{k+1}(\mathbf{x}) \neq 0$.

C2 We must be able to easily evaluate the unnormalized probability $p_k^*(\mathbf{x})$, $\forall \mathbf{x} \in \mathcal{X}$, $k = 0, ..., K$.

C3 For each $k = 1, ..., K-1$, we must be able to draw a sample $\mathbf{x}'$ given $\mathbf{x}$ using a Markov chain transition operator $T_k(\mathbf{x}'; \mathbf{x})$ that leaves $p_k(\mathbf{x})$ invariant:

$$\int T_k(\mathbf{x}'; \mathbf{x}) p_k(\mathbf{x}) \mathrm{d}\mathbf{x} = p_k(\mathbf{x}'). \tag{21}$$

C4 We must be able to draw (preferably independent) samples from p_A.

The transition operators $T_k(\mathbf{x}'; \mathbf{x})$ represent the probability density of transitioning from state $\mathbf{x}$ to $\mathbf{x}'$. Constructing a suitable sequence of intermediate probability distributions will depend on the problem. One general way to define this sequence is to set:

$$p_k(\mathbf{x}) \propto p_A^*(\mathbf{x})^{1-\beta_k} p_B^*(\mathbf{x})^{\beta_k}, \tag{22}$$

with $0 = \beta_0 < \beta_1 < ... < \beta_K = 1$ chosen by the user. Once the sequence of intermediate distributions has been defined we have:

Annealed Importance Sampling (AIS) run:

1. Generate $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_K$ as follows:

   - Sample $\mathbf{x}_1$ from $p_A = p_0$.
   - Sample $\mathbf{x}_2$ given $\mathbf{x}_1$ using $T_1(\mathbf{x}_1; \mathbf{x}_2)$.
   - ...
   - Sample $\mathbf{x}_K$ given $\mathbf{x}_{K-1}$ using $T_{K-1}(\mathbf{x}_K; \mathbf{x}_{K-1})$.

2. Set

$$w_{AIS}^{(i)} = \frac{p_1^*(\mathbf{x}_1)}{p_0^*(\mathbf{x}_1)} \frac{p_2^*(\mathbf{x}_2)}{p_1^*(\mathbf{x}_2)} ... \frac{p_{K-1}^*(\mathbf{x}_{K-1})}{p_{K-2}^*(\mathbf{x}_{K-1})} \frac{p_K^*(\mathbf{x}_K)}{p_{K-1}^*(\mathbf{x}_K)}.$$

Note that there is no need to compute the normalizing constants of any intermediate distributions. After performing M runs of AIS, the importance weights $w_{AIS}^{(i)}$ can be substituted into Eq. 20 to obtain an estimate of the ratio of partition functions:

$$\frac{Z_B}{Z_A} \approx \frac{1}{M} \sum_{i=1}^{M} w_{AIS}^{(i)} = \hat{r}_{\text{AIS}}. \tag{23}$$

It is shown in [15, 16] that for sufficiently large number of intermediate distributions K, the variance of $\hat{r}_{\text{AIS}}$ will be proportional to $1/MK$. Provided K is kept large, the total amount of computation can be split in any way between the number of intermediate distributions K and the number of annealing runs M without adversely affecting the accuracy of the estimator. If samples drawn from p_A are independent, the AIS runs can be used to obtain the variance of the estimate $\hat{r}_{\text{AIS}}$:

$$\text{Var}(\hat{r}_{\text{AIS}}) = \frac{1}{M} \text{Var}(w_{AIS}^{(i)}) \approx \frac{\hat{s}^2}{M} = \hat{\sigma}^2, \tag{24}$$

where $\hat{s}^2$ is estimated simply from the sample variance of the importance weights.

The intuition behind AIS is the following. Consider the following identity:

$$\frac{Z_K}{Z_0} = \frac{Z_1}{Z_0} \frac{Z_2}{Z_1} ... \frac{Z_K}{Z_{K-1}} \tag{25}$$

Provided the two intermediate distributions p_k and p_{k+1} are close enough, a simple importance sampler can be used to estimate each ratio Z_{k+1}/Z_k:

$$\frac{Z_{k+1}}{Z_k} \approx \frac{1}{M} \sum_{i=1}^{M} \frac{p_{k+1}^*(\mathbf{x}^{(i)})}{p_k^*(\mathbf{x}^{(i)})}, \qquad \text{where} \quad \mathbf{x}^{(i)} \sim p_k$$

These ratios can then be used to estimate $\frac{Z_K}{Z_0} = \prod_{k=0}^{K-1} \frac{Z_{k+1}}{Z_k}$. The problem with this approach is that, except for p_0, we typically cannot easily draw exact samples from intermediate distributions p_k. We

could resort to Markov chain methods, but then it is hard to determine when the Markov chain has converged to the desired distribution.

A remarkable fact shown by [15, 9] is that the estimate of Z_K/Z_0 will be exactly unbiased if each ratio Z_{k+1}/Z_k is estimated using $M = 1$, and a sample $\mathbf{x}^{(i)}$ is obtained by using Markov chain starting at the previous sample. The proof of this fact relies on the observation that the AIS procedure is just a simple importance sampling defined on the extended state space $X = (x_1, x_2, ..., x_K)$. Indeed, consider the unnormalized joint distribution defined by the AIS procedure:

$$Q^*(X) = Z_0 p_0(x_1) \prod_{k=1}^{K-1} T_k(x_{k+1}; x_k). \tag{26}$$

We can view $Q(X)$ as a proposal distribution for the target distribution $P(X)$ on the extended space X. This target distribution is defined by the reverse AIS procedure:

$$P^*(X) = Z_K p_K(x_K) \prod_{k=1}^{K-1} \hat{T}_k(x_k; x_{k+1}), \tag{27}$$

where $\hat{T}$ are the reverse transition operators:

$$\hat{T}_k(x'; x) = T_k(x; x') \frac{p_k(x')}{p_k(x)} \tag{28}$$

If T_k is reversible then $\hat{T}_k$ are the same as T_k. Due to invariance of p_k with respect to T_k (Eq. 21), the reverse transition operators are valid transition probabilities, which ensures that the marginal distribution over x_K is correct. From Eq. 20, the importance weight can then be found as:

$$w = \frac{P^*(X)}{Q^*(X)} = \frac{Z_K p_K(x_K) \prod_{i=1}^{K-1} \hat{T}_k(x_k; x_{k+1})}{Z_0 p_0(x_1) \prod_{k=1}^{K-1} T_k(x_{k+1}; x_k)} = \frac{p_K^*(x_K)}{p_0^*(x_1)} \prod_{k=1}^{K-1} \frac{p_k^*(x_k)}{p_k^*(x_{k+1})} = \prod_{k=1}^{K} \frac{p_k^*(x_k)}{p_{k-1}^*(x_k)}, \tag{29}$$

which are the weights provided by the AIS algorithm. Observe that the Markov transition operators do not necessarily need to be ergodic. In particular, if we were to choose dumb transition operators that do nothing, $T_k(x'; x) = \delta(x' - x)$ for all k, we recover the original simple importance sampling procedure.

3.3 Bridge Sampling

Suppose that in addition to having two distributions $p_A(\mathbf{x}) = p_A^*(\mathbf{x})/Z_A$ and $p_B(\mathbf{x}) = p_B^*(\mathbf{x})/Z_B$, we also have a "bridge distribution" $p_{A,B}(\mathbf{x}) = p_{A,B}^*(\mathbf{x})/Z_*$, such that it is overlapped with both p_A and p_B. We can then use a simple importance sampling procedure to separately estimate Z_*/Z_A and Z_*/Z_B to obtain:

$$\begin{aligned} \frac{Z_B}{Z_A} &= \frac{Z_*/Z_A}{Z_*/Z_B} = \mathrm{E}_{p_A}\left[\frac{p_{A,B}^*(\mathbf{x})}{p_A^*(\mathbf{x})}\right] \bigg/ \mathrm{E}_{p_B}\left[\frac{p_{A,B}^*(\mathbf{x})}{p_B^*(\mathbf{x})}\right]. \\ &\approx \frac{1}{M_A} \sum_{i=1}^{M_A} \frac{p_{A,B}^*(\mathbf{x}^{(0,i)})}{p_A^*(\mathbf{x}^{(0,i)})} \bigg/ \frac{1}{M_B} \sum_{i=1}^{M_B} \frac{p_{A,B}^*(\mathbf{x}^{(1,i)})}{p_B^*(\mathbf{x}^{(1,i)})} = \hat{r}_{Bridged}, \end{aligned} \tag{30}$$

where $\mathbf{x}^{(0,i)} \sim p_A$ and $\mathbf{x}^{(1,i)} \sim p_B$. Compare this estimator to the SIS estimator of Eq. 20. With SIS, draws from P_A are used as proposals for P_B, whereas with bridged sampling, draws from both P_A and P_B are used as proposals for $P_{A,B}$. The distribution $P_{A,B}$ acts as a "connecting bridge" between our

two distributions [11, 4]. One simple choice is to use a geometric bridge: $P_{A,B} = \sqrt{P_A P_B}$, in which case we get:

$$\frac{Z_B}{Z_A} = \mathrm{E}_{p_A}\left[\sqrt{\frac{p_B^*(\mathbf{x})}{p_A^*(\mathbf{x})}}\right] \Big/ \mathrm{E}_{p_B}\left[\sqrt{\frac{p_A^*(\mathbf{x})}{p_B^*(\mathbf{x})}}\right]. \tag{31}$$

The square root helps to control the magnitude of the importance weights and ensures that both $\sqrt{p_A^*/p_B^*}$ and $\sqrt{p_B^*/p_A^*}$ are square integrable with respect to p_B and p_A respectively. Other choices for bridge distributions are discussed in more detail in [11, 1].

The advantage of bridge sampling over SIS is that it uses a much weaker requirement on the support of the proposal distribution p_A. The bridge sampling estimate, that uses for example a geometric bridge, will converge to the correct estimate provided that $\Omega_A \bigcap \Omega_B \neq \emptyset$, i.e. there is some region that has non-zero probability under both p_A and p_B. The SIS, on the other hand, uses a much stronger assumption that $\Omega_B \subseteq \Omega_A$. Consider the following intuitive example, which we borrow from [16]. Let $p_A^*(x) = \mathrm{I}\{x \in (0,3)\}$ and $p_B^*(x) = \mathrm{I}\{x \in (2,4)\}$, so that $Z_A = 3$ and $Z_B = 2$. The bridged sampling that uses $p_{A,B}^*(x) = \mathrm{I}\{x \in (2,3)\}$ converges to the correct estimate, since the numerator converges to 1/3 and the denominator converges to 1/2. SIS, on the other hand, will converge to the wrong value of 1/3.

If however, p_A and p_B are not close enough, the bridge sampling estimator will be poor. We could resort to the same idea of defining a set of intermediate distributions as in Eq. 25, and estimate each ratio using bridge sampling. However, there are two problems with this approach. First, the estimator r_{Bridge} is biased, and so we cannot average over multiple independent runs to get a better estimator. Second, as previously discussed, it is not clear how we can draw samples from the intermediate distributions.

Neal [16] has recently developed a method called Linked Importance Sampling (LIS) that combines the ideas of AIS and bridge sampling to produce an unbiased estimate of the ratio of partition functions without the need to draw exact samples from the intermediate distributions. The idea of LIS is that each intermediate distribution p_k is "linked" to the next distribution p_{k+1} by a single linked state, which is chosen using the bridge distribution $p_{k,k+1}$. Just as AIS, the LIS procedure, can be viewed as a simple importance sampling defined on the extended state space, thus producing an unbiased estimator.

Furthermore, as pointed out in [16, 2], since we can view both AIS and LIS as simple importance sampling procedures defined on the extended state space, we can obtain a bridged sampling estimate by combining both forward and reverse estimators, as for example in Eq. 31, leading to the bridged AIS and bridged LIS estimators.

4   Learning Boltzmann Machines

In this section we will provide a brief overview of restricted Boltzmann machines (RBM's), semi-restricted Boltzmann machines (SRBM's), and Boltzmann machines (BM's). We will focus on showing how AIS can be used to estimate partition functions of these models. However, we could just as easily apply LIS, or bridged versions of AIS and LIS, which could potentially provide better estimates. This we leave for future study. We will further present a new learning algorithm for general Boltzmann machines.

4.1   Restricted Boltzmann Machines (RBM's)

A restricted Boltzmann machine is a particular type of MRF that has a two-layer architecture, in which the visible, binary stochastic units $\mathbf{v} \in \{0,1\}^D$ are connected to hidden binary stochastic units $\mathbf{h} \in \{0,1\}^P$ (see Fig. 1, left panel). The energy of the state $\{\mathbf{v}, \mathbf{h}\}$ is:

$$E(\mathbf{v}, \mathbf{h}; \theta) = -\sum_{i=1}^{D}\sum_{j=1}^{P} W_{ij} v_i h_j - \sum_{i=1}^{D} b_i v_i - \sum_{j=1}^{P} a_j h_j, \tag{32}$$

where $\theta = \{W, \mathbf{b}, \mathbf{a}\}$ are the model parameters: W_{ij} represents the symmetric interaction term between visible unit i and hidden unit j; b_i and a_j are the bias terms. The probability that the model assigns to a visible vector $\mathbf{v}$ is:

$$p(\mathbf{v}; \theta) = \frac{1}{Z(\theta)} \sum_h \exp\left(-E(\mathbf{v}, \mathbf{h}; \theta)\right). \tag{33}$$

The conditional distributions over hidden units $\mathbf{h}$ and visible vector $\mathbf{v}$ are given by logistic functions:

$$p(\mathbf{h}|\mathbf{v}; \theta) = \prod_j p(h_j|\mathbf{v}; \theta), \quad p(\mathbf{v}|\mathbf{h}; \theta) = \prod_i p(v_i|\mathbf{h}; \theta),$$

$$p(h_j = 1|\mathbf{v}; \theta) = \sigma(\sum_i W_{ij} v_i + a_j), \quad p(v_i = 1|\mathbf{h}; \theta) = \sigma(\sum_j W_{ij} h_j + b_i), \tag{34}$$

where $\sigma(x) = 1/(1 + \exp(-x))$. The derivative of the log-likelihood with respect to the model parameter W can be obtained from Eq. 33:

$$\frac{\partial \log p(\mathbf{v}; \theta)}{\partial W} = \mathrm{E}_{P_{\mathrm{data}}}[\mathbf{v}\mathbf{h}^\top] - \mathrm{E}_{P_{\mathrm{model}}}[\mathbf{v}\mathbf{h}^\top], \tag{35}$$

where $\mathrm{E}_{P_{\mathrm{data}}}[\cdot]$ denotes an expectation with respect to the data distribution $P_{\mathrm{data}}(\mathbf{h}, \mathbf{v}; \theta) = p(\mathbf{h}|\mathbf{v}; \theta) P_{\mathrm{data}}(\mathbf{v})$, with $P_{\mathrm{data}}(\mathbf{v})$ representing the empirical distribution, and $\mathrm{E}_{P_{\mathrm{model}}}[\cdot]$ is an expectation with respect to the distribution defined by the model. The learning rule for the biases is just a simplified version of Eq. 35. The expectation $\mathrm{E}_{P_{\mathrm{model}}}[\cdot]$ cannot be computed analytically.

4.1.1 Stochastic Approximation Procedure (SAP)

Stochastic Approximation Procedure (SAP) [25, 32, 33, 13] uses MCMC methods to stochastically approximate the second term of Eq. 35. The idea behind SAP is straightforward. Let θ_t and X^t be the current parameter and the state. Then X^t and θ_t are updated sequentially as follows. Given X^t, a new state X^{t+1} is sampled from the transition operator $T_{\theta_t}(X^{t+1}; X^t)$ that leaves p_{θ_t} invariant. A new parameter θ_{t+1} is then obtained by replacing the second term of Eq. 35 by the expectation with respect to X^{t+1}. In practice, we typically maintain a set of M sample points $X^t = \{\mathbf{x}^{t,1}, ..., \mathbf{x}^{t,M}\}$. The algorithm proceeds as follows:

Stochastic Approximation Procedure (SAP):

1. Initialize θ_0 and $X^0 = \{\mathbf{x}^{0,1}, ..., \mathbf{x}^{0,M}\}$.

2. For t=0 to T

    (a) For $i = 1, ..., M$ sample $\mathbf{x}^{t+1,i}$ given $\mathbf{x}^{t,i}$ using transition operator $T_{\theta_t}(\mathbf{x}^{t+1,i}; \mathbf{x}^{t,i})$.

    (b) Update $\theta_{t+1} = \theta_t + \alpha_t F(\theta_t, X^{t+1})$.

    (c) Decrease α_t.

For an RBM model, the state is $\mathbf{x} = \{\tilde{\mathbf{v}}, \tilde{\mathbf{h}}\}$, the transition kernel T_θ is defined by the blocked Gibbs updates (Eqs. 34), and

$$F(W_t, X^{t+1}) = \mathrm{E}_{P_{\mathrm{data}}}[\mathbf{v}\mathbf{h}^\top] - \frac{1}{M} \sum_{m=1}^M [\tilde{\mathbf{v}}^{t+1,m}(\tilde{\mathbf{h}}^{t+1,m})^\top], \tag{36}$$

$$F(\mathbf{a}_t, X^{t+1}) = \mathrm{E}_{P_{\mathrm{data}}}[\mathbf{h}] - \frac{1}{M} \sum_{m=1}^M [(\tilde{\mathbf{h}}^{t+1,m})],$$

$$F(\mathbf{b}_t, X^{t+1}) = \mathrm{E}_{P_{\mathrm{data}}}[\mathbf{v}] - \frac{1}{M} \sum_{m=1}^M [\tilde{\mathbf{v}}^{t+1,m}].$$
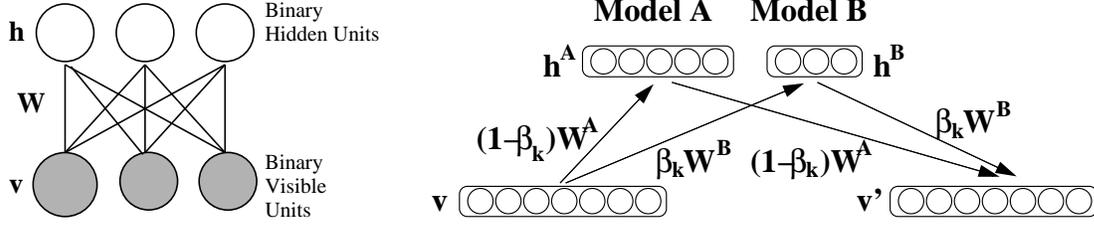
Figure 1: **Left:** Restricted Boltzmann machine. The top layer represents a vector of stochastic binary units **h** and the bottom layer represents a vector of stochastic binary visible variables **v**. **Right:** The Gibbs transition operator $T_k(\mathbf{v}'; \mathbf{v})$ leaves $p_k(\mathbf{v})$ invariant when estimating the ratio of partition functions Z_B/Z_A.

SAP belongs to the class of well-studied stochastic approximation algorithms of Robbins-Monro type [32, 33, 19]. The proof of convergence of these algorithms relies on the following basic decomposition. Let $S(\theta) = \frac{\partial \log p(\mathbf{v};\theta)}{\partial \theta}$, then:

$$\theta_{t+1} = \theta_t + \alpha_t S(\theta_t) + \alpha_t(F(\theta_t, X^{t+1}) - S(\theta_t)) = \theta_t + \alpha_t S(\theta_t) + \alpha_t \epsilon_{t+1}. \tag{37}$$

The first term is the discretization of the ordinary differential equation $\dot{\theta} = S(\theta)$. The algorithm is therefore a perturbation of this discretization with the noise term ϵ. The proof then proceeds by showing that the noise term is not too large.

Precise sufficient conditions that ensure almost sure convergence to an asymptotically stable point of $\dot{\theta} = S(\theta)$ are given in [32, 33]. One necessary condition requires the learning rate to decrease with time, i.e. $\sum_{t=0}^{\infty} \alpha_t = \infty$ and $\sum_{t=0}^{\infty} \alpha_t^2 < \infty$. This condition can for example be satisfied simply by setting $\alpha_t = 1/t$. Other conditions basically ensure that the speed of convergence of the Markov chain, govern by the transition operator T_θ, does not decrease too fast as θ tends to infinity, and that the noise term ϵ in the update of Eq. 37 is bounded.

Typically, in practice, the sequence $|\theta_t|$ is bounded[1], and the Markov chain, governed by the transition kernel T_θ, is ergodic. Together with the condition on the learning rate, this ensures almost sure convergence of SAP to an asymptotically stable point of $\dot{\theta} = S(\theta)$. When applied to learning RBM's, [25] shows that this stochastic approximation algorithm, also termed Persistent Contrastive Divergence, performs quite well compared to Contrastive Divergence learning [7].

4.1.2   Estimating the Ratio of Partition Functions of two RBM's

Suppose we have two RBM's with parameter values $\theta_A = \{W^A, \mathbf{b}^A, \mathbf{a}^A\}$ and $\theta_B = \{W^B, \mathbf{b}^B, \mathbf{a}^B\}$ that define probability distributions p_A and p_B over $\mathcal{V} \in \{0, 1\}^D$. Each RBM can have a different number of hidden units $\mathbf{h}^A \in \{0, 1\}^{P_A}$ and $\mathbf{h}^B \in \{0, 1\}^{P_B}$. We could define intermediate distributions using Eq. 22. However, sampling from these distributions would be much harder than from an RBM. Instead we introduce the following sequence of distributions for $k = 0, ..., K$ [20]:

$$p_k(\mathbf{v}) = \frac{p_k^*(\mathbf{v})}{Z_k} = \frac{1}{Z_k} \sum_{\mathbf{h}} \exp\left(-E_k(\mathbf{v}, \mathbf{h})\right), \tag{38}$$

where $\mathbf{h} = \{\mathbf{h}^A, \mathbf{h}^B\}$, and the energy function is given by:

$$E_k(\mathbf{v}, \mathbf{h}) = (1 - \beta_k)E(\mathbf{v}, \mathbf{h}^A; \theta_A) + \beta_k E(\mathbf{v}, \mathbf{h}^B; \theta_B), \tag{39}$$

with $0 = \beta_0 < \beta_1 < ... < \beta_K = 1$. For $i = 0$, we have $\beta_0 = 0$ and so $p_0 = p_A$. Similarly, for $i = K$, we have $p_K = p_B$. For the intermediate values of k, we will have some interpolation between p_A and p_B.

---

[1]θ_t belongs to some compact set, which ensures its boundedness. In fact, we could always define a procedure that projects θ_t onto some compact set.

11

Let us now define a Markov chain transition operator $T_k(\mathbf{v}'; \mathbf{v})$ that leaves $p_k(\mathbf{v})$ invariant. Using Eqs. 38, 39, it is straightforward to derive a block Gibbs sampler. The conditional distributions are given by logistic functions:

$$p(h_j^A = 1|\mathbf{v}) = \sigma\left((1 - \beta_k)(\sum_i W_{ij}^A v_i + a_j^A)\right), \tag{40}$$

$$p(h_j^B = 1|\mathbf{v}) = \sigma\left(\beta_k(\sum_i W_{ij}^B v_i + a_j^B)\right), \tag{41}$$

$$p(v_i' = 1|\mathbf{h}) = \sigma\left((1 - \beta_k)(\sum_j W_{ij}^A h_j^A + b_i^A) + \beta_k(\sum_j W_{ij}^B h_j^B + b_i^B)\right). \tag{42}$$

Given $\mathbf{v}$, Eqs. 40, 41 are used to stochastically activate hidden units $\mathbf{h}^A$ and $\mathbf{h}^B$. Eq. 42 is then used to draw a new sample $\mathbf{v}'$ as shown in Fig. 1 (right panel). Due to the special structure of RBM's, the cost of summing out $\mathbf{h}$ is linear in the number of hidden units. We can therefore easily evaluate:

$$
\begin{aligned}
p_k^*(\mathbf{v}) &= \sum_{\mathbf{h}^A, \mathbf{h}^B} e^{(1-\beta_k)E(\mathbf{v},\mathbf{h}^A;\theta_A) + \beta_k E(\mathbf{v},\mathbf{h}^B;\theta_B)} \\
&= e^{(1-\beta_k)\sum_i b_i^A v_i} \prod_{j=1}^{P_A}(1 + e^{(1-\beta_k)(\sum_i W_{ij}^A v_i + a_j^A)}) \times e^{\beta_k \sum_i b_i^B v_i} \prod_{j=1}^{P_B}(1 + e^{\beta_k(\sum_i W_{ij}^B v_i + a_j^B)}).
\end{aligned}
$$

We will assume that the parameter values of each RBM are bounded in which case $p(\mathbf{v}) > 0$ for all $\mathbf{v} \in \mathcal{V}$. This will ensure that condition C1 of the AIS procedure is always satisfied. We have already shown that conditions C2 and C3 are satisfied. For condition C4, we can run a blocked Gibbs sampler (Eq. 34) to generate samples from p_A. These sample points will not be independent, but the AIS estimator will still converge to the correct value, provided our Markov chain is ergodic [15]. However, assessing the accuracy of this estimator can be difficult, as it depends on both the variance of the importance weights and on autocorrelations in the Gibbs sampler.

4.1.3 Estimating Partition Functions of RBM's

In the previous section we showed that we can use AIS to obtain an estimate of Z_B/Z_A. Consider an RBM with parameter vector $\theta_A = \{0, \mathbf{b}^A, 0\}$, or an RBM with a zero weight matrix. From Eq. 33, we know:

$$Z_A = 2^{P_A}\prod_i(1 + e^{b_i}), \quad p_A(\mathbf{v}) = \prod_i p_A(v_i) = \prod_i 1/(1 + e^{-b_i}), \tag{43}$$

so we can draw exact independent samples from this "base-rate" RBM. AIS in this case allows us to obtain an *unbiased* estimate of the partition function Z_B. This approach closely resembles simulated annealing, since the intermediate distributions of Eq. 38 take form:

$$p_k(\mathbf{v}) = \frac{\exp((1-\beta_k)\mathbf{v}^\top \mathbf{b}^A)}{Z_k} \sum_{\mathbf{h}^B} \exp(-\beta_k E(\mathbf{v}, \mathbf{h}^B; \theta_B)). \tag{44}$$

We gradually change β_k (or inverse temperature) from 0 to 1, annealing from a simple "base-rate" model to the final complex model. The importance weights $w_{AIS}^{(i)}$ ensure that AIS produces correct estimates.
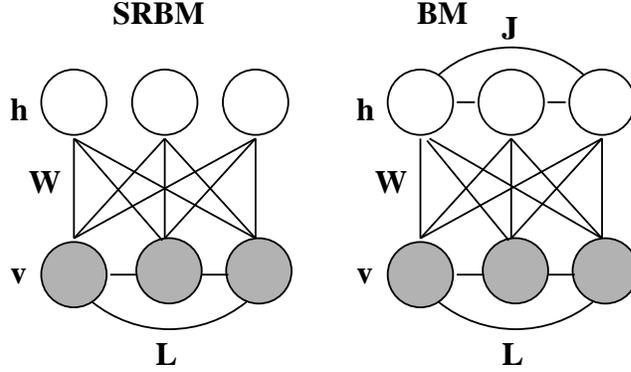
Figure 2: **Left:** Semi-restricted Boltzmann machine. The visible units form a fully or partially connected conditional MRF, with hidden units determining the biases of the visible units. **Right:** Boltzmann Machine with both visible-to-visible and hidden-to-hidden connections.

4.2 Semi-Restricted Boltzmann Machines (SRBM's)

Semi-restricted Boltzmann machines were introduced by [18]. In contrast to RBM's, the visible units of SRBM's form a fully or partially connected conditional MRF, with hidden states determining the biases of the visible units (see Fig. 2, left panel). The energy of the state $\{\mathbf{v}, \mathbf{h}\}$ takes form:

$$E(\mathbf{v}, \mathbf{h}; \theta) = -\sum_{i<k} L_{ik} v_i v_k - \sum_{i,j} W_{ij} v_i h_j - \sum_i b_i v_i - \sum_j a_j h_j, \qquad (45)$$

where L_{ij} represents the symmetric lateral interaction term between visible units i and j, with diagonal elements set to 0. The conditional distributions over hidden and visible units are given by:

$$p(h_j = 1|\mathbf{v}) = \sigma(\sum_i W_{ij} v_i + a_j), \qquad p(v_i = 1|\mathbf{h}, \mathbf{v}_{-i}) = \sigma(\sum_j W_{ij} h_j + \sum_{k \setminus i} L_{ik} v_j + b_i). \qquad (46)$$

The derivative of the log-likelihood with respect to the lateral interaction term L is given by:

$$\frac{\partial \log p(\mathbf{v}; \theta)}{\partial L} = \mathrm{E}_{P_{\mathrm{data}}}[\mathbf{v}\mathbf{v}^\top] - \mathrm{E}_{P_{\mathrm{model}}}[\mathbf{v}\mathbf{v}^\top]. \qquad (47)$$

The crucial aspect of SRBM is that inference of the hidden variables in this model is still exact, since there are no lateral connections between hidden units. Therefore the first term in Eq. 47 is still easy to compute. Learning in this model was originally carried out using Contrastive Divergence with a few damped mean-field updates on the visible units [29, 18]. Instead, we will apply SAP with the transition kernel T_θ defined by the blocked Gibbs update for the hidden units and sequential Gibbs updates for the visible units (Eqs. 46). Since we can explicitly sum out the hidden units, the AIS procedure for estimating the ratio of partition functions of two SRBM's will be almost identical to the AIS procedure we described for RBM's.

4.3 Boltzmann Machines (BM's)

A Boltzmann machine is a network of symmetrically coupled stochastic binary units with both visible-to-visible and hidden-to-hidden lateral connections as shown in Fig. 2 (right panel). The energy function is defined as:

$$E(\mathbf{v}, \mathbf{h}; \theta) = -\sum_{i<k} L_{ik} v_i v_k - \sum_{j<m} J_{jm} h_j h_m - \sum_{i,j} W_{ij} v_i h_j - \sum_i b_i v_i - \sum_j a_j h_j, \qquad (48)$$

13

The conditional distributions over hidden and visible units are given by:

$$p(h_j = 1|\mathbf{v}, \mathbf{h}_{-j}) = \sigma(\sum_i W_{ij} v_i + \sum_{m \setminus j} J_{jm} h_j + a_j), \tag{49}$$

$$p(v_i = 1|\mathbf{h}, \mathbf{v}_{-i}) = \sigma(\sum_j W_{ij} h_j + \sum_{k \setminus i} L_{ik} v_j + b_i). \tag{50}$$

As it was the case for the RBM's and SRBM's, the derivative of the log-likelihood with respect to the model parameters W and L are given by Eqs. 35, 47, and

$$\frac{\partial \log p(\mathbf{v}; \theta)}{\partial J} = \mathrm{E}_{P_{\text{data}}}[\mathbf{h}\mathbf{h}^\top] - \mathrm{E}_{P_{\text{model}}}[\mathbf{h}\mathbf{h}^\top]. \tag{51}$$

This simple learning algorithm was originally derived by [6]. In contrast to RBM's and SRBM's, exact inference in this model is intractable. Hinton and Sejnowski [6] proposed an approximate learning algorithm that uses Gibbs sampling to approximate both expectations. In the positive phase, the expectation $\mathrm{E}_{P_{\text{data}}}[\cdot]$ is approximated by running a separate Markov chain for every training data vector, clamped to the states of the visible units. In the negative phase, an additional chain is run to approximate $\mathrm{E}_{P_{\text{model}}}[\cdot]$. The main problem with this algorithm is that it is computationally very demanding and not particularly practical. For each iteration of learning, we must wait until each Markov chain reaches its stationary distribution to do learning.

It was further observed [29, 18] that for Contrastive Divergence to perform well, it is important to obtain exact samples from the conditional distribution $p(\mathbf{h}|\mathbf{v})$. Instead of using Contrastive Divergence learning, we will combine the ideas of variational learning, introduced in section 2, together with SAP. Consider the following variational lower bound:

$$\log p(\mathbf{v}; \theta) \geq - \sum_{\mathbf{h}} q(\mathbf{h}) E(\mathbf{v}, \mathbf{h}; \theta) + \mathcal{H}(q) - \log Z(\theta), \tag{52}$$

with equality attained if and only if $q(\mathbf{h}) = p(\mathbf{h}|\mathbf{v})$. Using a mean-field approach of section 2.3, we choose a fully factorized distribution in order to approximate the true posterior: $q(\mathbf{h}) = \prod_{j=1}^P q(h_i)$, with $q(h_i = 1) = \mu_i$ and P is the number of hidden units. In this case, the lower bound on the log-probability of the data takes form:

$$\begin{aligned}
\log p(\mathbf{v}; \theta) &\geq \sum_{i<k} L_{ik} v_i v_k + \sum_{j<m} J_{jm} \mu_j \mu_m + \sum_{i,j} W_{ij} v_i \mu_j + \sum_i b_i v_i + \sum_j a_j \mu_j \\
&- \sum_j \left[\mu_j \log(\mu_j) + (1 - \mu_j) \log(1 - \mu_j) \right] - \log Z(\theta).
\end{aligned} \tag{53}$$

The learning proceeds with the **positive phase**: maximizing this lower bound with respect to the variational parameters μ for fixed θ, which results in the mean-field fixed-point equations:

$$\mu_j \leftarrow \sigma(\sum_i W_{ij} v_i + \sum_{m \setminus j} J_{mj} h_m + a_j), \tag{54}$$

followed by the **negative phase**: applying SAP to update the model parameters θ. In more detail, consider the training set of N data vectors $\{\mathbf{v}\}_{n=1}^N$, the Boltzmann machine learning proceeds as follows:

Boltzmann Machine Learning Procedure:

1. Randomly initialize θ_0 and the negative samples $\{\tilde{\mathbf{v}}^{0,1}, \tilde{\mathbf{h}}^{0,1}\}, ..., \{\tilde{\mathbf{v}}^{0,M}, \tilde{\mathbf{h}}^{0,M}\}$,

2. For t=0 to T (# of iterations)

   Positive Phase:

   (a) For each training sample $\mathbf{v}^n$, n=1 to N

   - Randomly initialize μ and run mean-field updates until convergence:

     $$\mu_j \leftarrow \sigma\Big(\sum_i W_{ij} v_i^n + \sum_{m \backslash j} J_{mj}\mu_m + a_j\Big).$$

   - Set $\mu^n = \mu$.

   Negative Phase:

   (b) For each negative sample m=1 to M

   - Obtain a new binary state $(\tilde{\mathbf{v}}^{t+1,m}, \tilde{\mathbf{h}}^{t+1,m})$ by running a k-step Gibbs sampler using Eqs. 49, 50, initialized at $(\tilde{\mathbf{v}}^{t,m}, \tilde{\mathbf{h}}^{t,m})$.

   Parameter Update:

   (c) Update

   $$W^{t+1} = W^t + \alpha_t \bigg(\frac{1}{N}\sum_{n=1}^N \mathbf{v}^n(\mu^n)^\top - \frac{1}{M}\sum_{m=1}^M \tilde{\mathbf{v}}^{t+1,m}(\tilde{\mathbf{h}}^{t+1,m})^\top\bigg),$$

   $$J^{t+1} = J^t + \alpha_t \bigg(\frac{1}{N}\sum_{n=1}^N \mu^n(\mu^n)^\top - \frac{1}{M}\sum_{m=1}^M \tilde{\mathbf{h}}^{t+1,m}(\tilde{\mathbf{h}}^{t+1,m})^\top\bigg),$$

   $$L^{t+1} = L^t + \alpha_t \bigg(\frac{1}{N}\sum_{n=1}^N \mathbf{v}^n(\mathbf{v}^n)^\top - \frac{1}{M}\sum_{m=1}^M \tilde{\mathbf{v}}^{t+1,m}(\tilde{\mathbf{v}}^{t+1,m})^\top\bigg).$$

   (d) Decrease α_t.

The choice of resorting to the naive mean-field approach in the positive phase was deliberate. First, the convergence is usually very fast, which greatly facilitates learning. More importantly, conditioned on the data, we do not want the posterior distribution over the hidden units to be multimodal, since we would not want to have multiple alternative explanations about the data. The mean-field inference exactly solves this problem. During learning, if the posterior given a training data is multimodal, the mean-field will lock onto exactly one mode, and learning will make it more probable. Hence, our learning procedure will attempt to find regions in the parameter space in which the true posterior is unimodal. It is also interesting to observe that if we set hidden-to-hidden connections to 0, we exactly recover the learning procedure for SRBM's.

In the next section we will show that together with the stochastic approximation procedure in the negative phase, we are able to efficiently learn a good generative model of MNIST digits. It is important to point out that this procedure readily extends to learning with real-valued, count, or tabular data, provided the distributions are in the exponential family.

4.3.1 Estimating Partition Functions of BM's

AIS can be used to estimate the partition function of a Boltzmann machine. In contrast to RBM's and SRBM's, we cannot explicitly sum out hidden units. Nevertheless, we can run the AIS procedure of

section 3.2 by setting $\mathbf{x} = \{\mathbf{v}, \mathbf{h}\}$. The intermediate distributions (see Eq. 22) are given by:

$$p_k(\mathbf{v}, \mathbf{h}) = \frac{\exp\left((1 - \beta_k)\mathbf{v}^\top \mathbf{b}^A\right)}{Z_k} \exp\left(-\beta_k E(\mathbf{v}, \mathbf{h}^B; \theta)\right),$$

where $\mathbf{b}^A$ are the biases of the base-rate model, and the energy term is defined in Eq. 48. The transition operator T_k is straightforward to derive from Eqs. 55,48. Furthermore, using the variational lower bound of Eq. 52, the estimate of the partition function, together with the mean-field updates of Eq. 54, will allow us to easily estimate the lower bound on the log-probability of test data.

5 Empirical Comparisons

In our experiments we used the MNIST digit dataset, which contains 60,000 training and 10,000 test images of ten handwritten digits (0 to 9), with 28×28 pixels. The dataset was binarized: each pixel value was stochastically set to 1 in proportion to its pixel intensity. We also created a toy dataset containing 60,000 training patches with 4×4 pixels, which were extracted from images of digits simply by placing a square at a random position in each of the 28×28 image.

Annealed importance sampling requires setting β_k's that define a sequence of intermediate distributions. In all of our experiments this sequence was chosen by quickly running a few preliminary experiments and picking the spacing of β_k so as to minimize the log variance of the final importance weights. The biases $\mathbf{b}^A$ of a base-rate model (see Eq. 43) were set by maximum likelihood, then smoothed to ensure that $p(\mathbf{v}) > 0, \forall\, \mathbf{v} \in \mathcal{V}$.

To speed-up learning, we subdivided datasets into 600 mini-batches, each containing 100 cases, and updated the weights after each mini-batch. The number of negative samples was also set to 100. For SAP, we always used 5 Gibbs updates. Each model was trained using 500 passes (epochs) through the entire training dataset. The initial learning rate was set 0.005 and was gradually decreased to 0. In all of our experiments we use natural logarithms, providing values in nats.

5.1 Toy Models

The RBM model had 25 hidden and 784 visible units, and was trained on the full binarized MNIST dataset. The SRBM and BM models had 500 and 5 hidden units respectively, and were trained on the toy 4×4 patch dataset. For all three models we could calculate the exact value of the partition function. For all models we also used 1,000 β_k spaced uniformly from 0 to 0.5, 4,000 β_k spaced uniformly from 0.5 to 0.9, and 5,000 β_k spaced uniformly from 0.9 to 1.0, with a total of 10,000 intermediate distributions. We also run loopy BP and tree-reweighted sum-product (TRW) algorithms to obtain the deterministic Bethe approximation and an upper bound on the log-partition function. For the TRW, the distribution ρ over the spanning trees was also optimized using conditional gradient together with the maximum weight spanning tree algorithm. For both loopy BP and TRW, the messages passing updates were damped.

Table 1 shows that for all three models, using only 10 AIS runs, we were able to obtain good estimates of partition functions. Furthermore, Fig. 3 (top row) reveals that as the number of annealing runs is increased, AIS can almost exactly recover the true value of the partition function across all three models. Bethe provided quite reasonable approximations for the SRBM and BM models, but was off by about 4 nats for the "semi-simple" RBM model. TRW, on the other hand, provided very loose upper bounds on the log-partition functions. In particular, for the RBM model with $\log Z = 354.88$, the tree-reweighted upper bound was 1207.83.
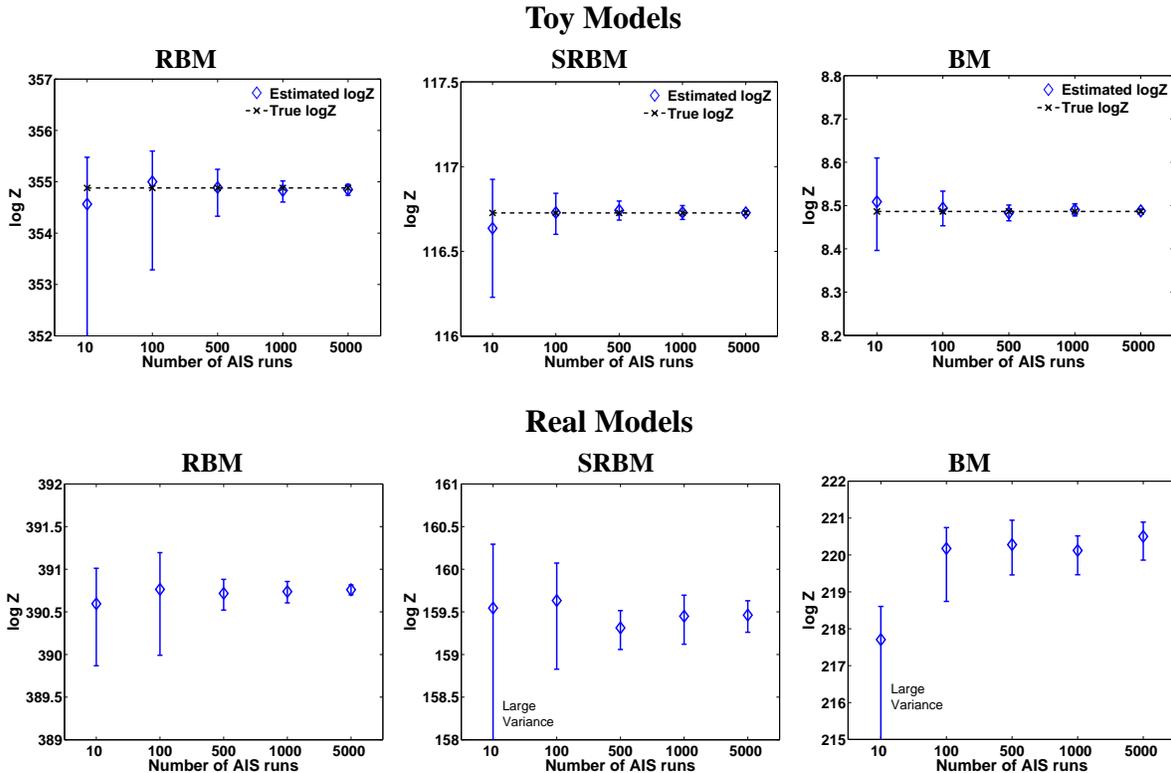
Figure 3: Estimates of the log-partition functions $\log \hat{Z}$ in nats as we increase the number of annealing runs. The error bars show $\log(\hat{Z} \pm 3\hat{\sigma})$. For all models we used 10,000 intermediate distributions.

Table 1: Results of estimating partition functions (nats) of toy RBM, SRBM, and BM models. For all models we used 10,000 intermediate distributions.

| AIS Runs | | True $\log Z$ | Estimates | | | Bethe $\log Z$ | TRW $\log Z$ |
			$\log \hat{Z}$	$\log(\hat{Z} \pm \hat{\sigma})$	$\log(\hat{Z} \pm 3\hat{\sigma})$		
100	RBM	354.88	354.99	$354.68, 355.24$	$353.28, 355.59$	350.75	1205.26
	SRBM	116.72	116.73	$116.68, 116.76$	$116.60, 116.84$	115.90	146.30
	BM	8.49	8.49	$8.48, 8.51$	$8.45, 8.53$	7.67	20.28

5.2   Real Models

In our second experiment we trained an RBM, an SRBM, and a fully connected BM on the binarized MNIST images. All models had 500 hidden units. We used exactly the same spacing of β_k as before and exactly the same base-rate model. Results are shown in table 2. For each model we were also able to get what appears to be a rather accurate estimate of Z. Of course, we are relying on an empirical estimate of AIS's accuracy, which could potentially be misleading. Nonetheless, Fig. 3 (bottom row) shows that as we increase the number of annealing runs, the value of the estimator does not fluctuate drastically. The difference between Bethe approximation and AIS estimate is quite large for all three models, and TRW did not provide any meaningful upper bounds.

Table 2 further shows an estimate of the average train/test log-probability of the RBM and SRBM models, and an estimate of the lower bound on the train/test log-probability of the BM model. For the RBM and SRBM models, the estimate of the test log-probability was -86.90 and -86.06 respectively. For the BM model, the estimate of the lower bound on the test log-probability was -85.68. Fig. 4 shows samples generated from all models by randomly initializing binary states of the visible and hidden units

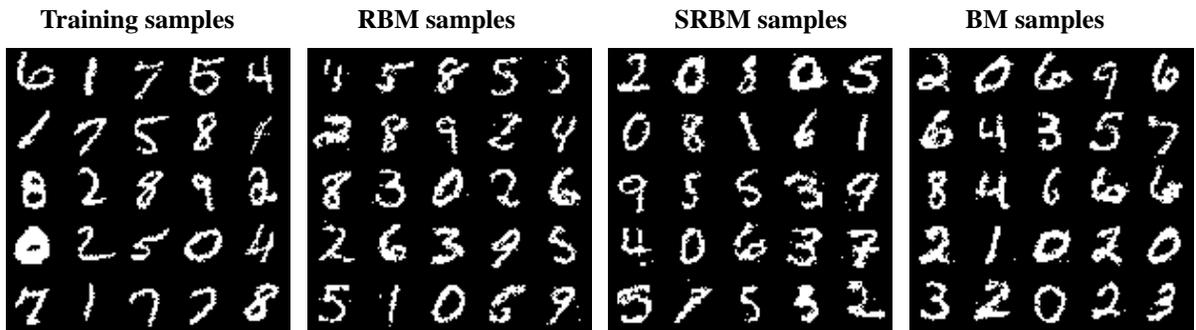| Training samples | RBM samples | SRBM samples | BM samples |

Figure 4: Random samples from the training set along with samples generated from RBM, SRBM, and BM models.

Table 2: Results of estimating partition functions (nats) of real RBM, SRBM, and BM models, along with the estimates of the average training and test log-probabilities. For the BM model, we report the lower bound on the log-probability. For all models we used 10,000 intermediate distributions.

AIS Runs		True $\log Z$	Estimates			Avg. log-prob.		Bethe $\log Z$
			$\log \hat{Z}$	$\log (\hat{Z} \pm \hat{\sigma})$	$\log (\hat{Z} \pm 3\hat{\sigma})$	Test	Train	
100	RBM	—	390.76	390.56, 390.92	389.99, 391.19	−86.90	−84.67	378.98
	SRBM	—	159.63	159.42, 159.80	158.82, 160.07	−86.06	−83.39	148.11
	BM	—	220.17	219.88, 220.40	218.74, 220.74	−85.59	−82.96	197.26

and running Gibbs sampler for 100,000 steps. Certainly, all samples look like the real handwritten digits.

We should point out that there are some difficulties with using AIS. There is a need to specify the β_k that define a sequence of intermediate distributions. The number and the spacing of β_k will be problem dependent and will affect the variance of the estimator. We also have to rely on the empirical estimate of AIS accuracy, which could potentially be very misleading [15, 16]. Even though AIS provides an unbiased estimator of Z, it occasionally gives large overestimates and usually gives small underestimates, so in practice, it is more likely to underestimate the true value of the partition function, which will result in an overestimate of the log-probability. But these drawbacks should not result in disfavoring the use of AIS for RBM's, SRBM's and BM's: it is much better to have a slightly unreliable estimate than no estimate at all, or an extremely indirect estimate, such as discriminative performance [8].

6 Conclusions

In this paper we provided a brief overview of the variational framework for estimating log-partition functions and some of the Monte Carlo based methods for estimating partition functions of arbitrary MRF's. We then developed an annealed importance sampling procedure for estimating partition functions of RBM, SRBM and BM models, and showed that they provide much better estimates compared to some of the popular variational methods.

We further developed a new learning algorithm for training Boltzmann machines. This learning procedure is computationally more demanding compared to learning RBM's or SRBM's, since it requires mean-field settling. Nevertheless, we were able to successfully learn a good generative model of MNIST digits. Furthermore, by appropriately setting some of the visible-to-hidden and hidden-to-hidden connections to zero, we can create a deep multi-layer Boltzmann machine with many layers of hidden variables. We can efficiently train these deep hierarchical undirected models, and together with AIS, we can obtain good estimates of the lower bound on the log-probability of the *test* data. This

will allow us to obtain some quantitative evaluation of the generalization performance of these deep hierarchical models. Furthermore, this learning procedure and AIS can be easily applied to undirected graphical models that generalize BM's to exponential family distributions. This will allow future application to models of real-valued data, such as image patches [18], or count data, such as word-count vectors of documents [3].

References

[1] C. H. Bennett. Efficient estimation of free energy dierences from Monte Carlo data. *Journal of Computational Physics*, 22:245–268, 1976.

[2] G. E. Crooks. Path-ensemble averages in systems driven far from equilibrium. *Physical Review*, 61:2361–2366, 2000.

[3] P. Gehler, A. Holub, and M. Welling. The Rate Adapting Poisson (RAP) model for information retrieval and object recognition. In *Proceedings of the 23rd International Conference on Machine Learning*, 2006.

[4] A. Gelman and X. L. Meng. Path sampling for computing normalizing constants: identities and theory. Technical report, Department of Statistics, University of Chicago, 1994.

[5] A. Globerson and T. Jaakkola. Approximate inference using conditional entropy decompositions. In *11th International Workshop on AI and Statistics (AISTATS'2007)*, 2007.

[6] G. Hinton and T. Sejnowski. Learning and relearning in Boltzmann machines. In Rumelhart and McClelland, editors, *Parallel Distributed Processing*, pages 283–335, 1986.

[7] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1711–1800, 2002.

[8] G. E. Hinton, S. Osindero, and Y. W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

[9] C. Jarzynski. A nonequilibrium equality for free energy differences. *Physical Review Letters*, 78:2690–2693, 1997.

[10] David MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, September 2003.

[11] X. L. Meng and W. H. Wong. Simulating ratios of normalizing constants via a simple identity: A theoretical exploration. *Statistica Sinica*, 6:831–860, Jun 1996.

[12] P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *J.R. Statist. Soc. B*, 68(3):411–436, 2006.

[13] R. M. Neal. Connectionist learning of belief networks. *Artif. Intell*, 56(1):71–113, 1992.

[14] R. M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto, September 1993.

[15] R. M. Neal. Annealed importance sampling. *Statistics and Computing*, 11:125–139, 2001.

[16] R. M. Neal. Estimating ratios of normalizing constants using linked importance sampling. Technical Report 0511, Department of Statistics, University of Toronto, 2005.

[17] R. M. Neal and G. E. Hinton. A view of the EM algorithm that justifies incremental, sparse and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 355–368. Kluwer Academic Press, 1998.

[18] S. Osindero and G. Hinton. Modeling image patches with a directed hierarchy of Markov random fields. In *NIPS 20*, Cambridge, MA, 2008. MIT Press.

[19] H. Robbins and S. Monro. A stochastic approximation method. *Ann. Math. Stat.*, 22:400–407, 1951.

[20] R. Salakhutdinov and I. Murray. On the quantitative analysis of deep belief networks. In *Proceedings of the International Conference on Machine Learning*, volume 25, 2008.

[21] J. Skilling. Nested sampling. *Bayesian inference and maximum entropy methods in science and engineering, AIP Conference Proceeedings*, 735:395–405, 2004.

[22] D. Sontag. Cutting plane algorithms for variational inference in graphical models. Technical report, MIT, 2007.

[23] D. Sontag and T. Jaakkola. New outer bounds on the marginal polytope. In *NIPS 20*, Cambridge, MA, 2008. MIT Press.

[24] E. Sudderth, M. Wainwright, and A. Willsky. Loop series and bethe variational bounds in attractive graphical models. In *NIPS 20*, Cambridge, MA, 2008. MIT Press.

[25] T. Tieleman. Training restricted Boltzmann machines using approximations to the likelihood gradient. In *Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004)*. ACM, 2008.

[26] M. J. Wainwright, T. Jaakkola, and A. S. Willsky. A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 51(7):2313–2335, 2005.

[27] M. J. Wainwright and M. Jordan. Graphical models, exponential families, and variational inference. Technical report, Department of Statistics, University of California, Berkeley, 2003.

[28] M. J. Wainwright and M. Jordan. Log-determinant relaxation for approximate inference in discrete markov random fields. *IEEE Transactions on Signal Processing*, 54(6), 2006.

[29] M. Welling and G. E. Hinton. A new learning algorithm for mean field Boltzmann machines. *Lecture Notes in Computer Science*, 2415, 2002.

[30] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. pages 239–236, January 2003.

[31] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 51(7):2282–2312, 2005.

[32] L. Younes. On the convergence of Markovian stochastic algorithms with rapidly decreasing ergodicity rates, March 17 2000.

[33] A. L. Yuille. The convergence of contrastive divergences. In *NIPS*, 2004.