Parallelizable Sampling of Markov Random Fields

James Martens
University of Toronto

Ilya Sutskever
University of Toronto

Abstract

Markov Random Fields (MRFs) are an important class of probabilistic models which are used for density estimation, classification, denoising, and for constructing Deep Belief Networks. Every application of an MRF requires addressing its inference problem, which can be done using deterministic inference methods or using stochastic Markov Chain Monte Carlo methods. In this paper we introduce a new Markov Chain transition operator that updates all the variables of a pairwise MRF in parallel by using auxiliary Gaussian variables. The proposed MCMC operator is extremely simple to implement and to parallelize. This is achieved by a formal equivalence result between arbitrary pairwise MRFs and a particular type of Restricted Boltzmann Machine. This result also implies that the later can be learned in place of the former without any loss of modeling power, a possibility we explore in experiments.

1 INTRODUCTION

Pairwise Markov Random Fields are probabilistic models useful for denoising (Malfait and Roose, 1997; Portilla et al., 2003), density estimation (Roth and Black, 2005; Wainwright and Simoncelli, 2000; Cross and Jain, 1981), classification (Larochelle and Bengio, 2008), and for learning Deep Belief Networks (Hinton et al., 2006; Hinton and Salakhutdinov, 2006).

Every application of MRFs requires dealing with inference, which is the problem of computing statistics with respect to the MRF's distribution. While the inference problem can be solved approximately for general MRFs and exactly for some special cases via deterministic methods (Wainwright, 2008), in this paper we focus on stochastic Markov Chain Monte Carlo methods (Neal, 1993).

Markov chain Monte Carlo (MCMC) is attractive as a general inference method because it is applicable to almost every probabilistic model and it is guaranteed to be unbiased and converge in the limit. In contrast, deterministic methods typically are not unbiased and may not even converge, except on models that exhibit special structure such as acyclic dependencies.

In this work, we introduce a mapping between fully-connected pairwise MRFs and a particular type of MRF known as Restricted Boltzmann Machines (RBMs) (Smolensky, 1986), with real-valued auxiliary hidden variables, such that the original MRF's distribution is recovered by integrating out these variables. By running parallel block Gibbs sampling on the RBM and discarding the hidden variable samples, we can obtain samples from the original MRF. Our method is very easy to implement, and its parallelizable nature makes it much more cost effective than SGS on parallel computing architectures such as GPUs.

In addition, the equivalence we show between MRFs and the marginal distributions of these special RBMs suggests that it may be beneficial to work directly with the latter, since they are equally expressive but easier to sample from efficiently.

2 RELATED WORK

Sampling from pairwise MRFs is an important and well-studied problem. Two well known MCMC methods are the Sequential Gibbs Sampling algorithm (SGS) and the Swendsen-Wang algorithm (SW). Arguably the simplest and most commonly used, SGS updates one variable at a time in sequence by conditioning on the rest. SW introduces auxiliary binary-valued variables into an MRF, one for each potential, and samples from a joint distribution whose marginal is equal to the original distribution. The samples are produced using a standard Gibbs chain, which alternates between sampling the auxiliary variables in-

Appearing in Proceedings of the 13^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2010, Chia Laguna Resort, Sardinia, Italy. Volume 9 of JMLR: W&CP 9. Copyright 2010 by the authors.

dependently and sampling the original ones using a reasonably efficient "cluster-and-flip" algorithm. This type of sampling is, in some contexts, preferable to SGS because it can make "global moves" that involve flipping large groups of highly correlated nodes all at once (a situation where SGS struggles). However, there is no general guarantee that the chain will mix fast (Gore and Jerrum, 1999) and in particular we do not even know if it will mix faster than SGS.

For certain pairwise MRFs with special properties there are exact algorithms for computing the marginal statistics in polynomial time that do not require sampling. For example, planarity of the associated graph is required in the case of the recent MRF inference algorithm due to Schraudolph and Kamenetsky (Schraudolph and Kamenetsky, 2009). In general these exact deterministic methods cannot be applied to arbitrary pairwise MRFs.

3 EVERY PAIRWISE MRF HAS AN EQUIVALENT RBM

In this section we introduce our main technical result, which states that every nonzero discrete-valued pairwise MRF (DPMRF) over the variables x can be represented by the marginal distribution of some RBM. Specifically, we introduce a special RBM $p(x, y)$ over the MRF variables x and a continuous-valued auxiliary variables y, and show that the marginal distribution $p(x)$ can be made equal to the distribution of any pairwise nonzero MRF. This result turns out to be useful because RBMs are much easier to sample from.

3.1 MARGINALIZING THE CONTINUOUS VARIABLES

Consider the energy function $E(x, y)$ defined over $x \in X \subseteq \mathbb{R}^n$ and $y \in \mathbb{R}^m$ which is given by

$$E(x, y) = \frac{1}{2}y^\top y - y^\top Wx - f(x)$$

where $W \in \mathbb{R}^{m\times n}$ and $f : X \to \mathbb{R}$.

If $Z = \int \exp(-E(x, y))\, dx\, dy < \infty$ then E defines a Boltzmann distribution over x and y by the equation

$$p(x, y) = \frac{1}{Z}\exp(-E(x, y))$$

We can rewrite the energy function E as

$$E(x, y) = \frac{1}{2}(y - Wx)^\top(y - Wx) - \frac{1}{2}x^\top W^\top Wx - f(x), \tag{1}$$

from which it follows that $p(y|x)$ is a multivariate normal distribution

$$p(y|x) = \frac{p(x, y)}{p(x)} \propto \exp\left((y - Wx)^\top(y - Wx)\right).$$

A useful property of the multivariate normal distribution is that its partition function does not depend on the mean-parameter. In particular, we have

$$\int \exp\left(-\frac{1}{2}(y - Wx)^\top(y - Wx)\right) dy = (2\pi)^{n/2}$$

which, critically, does not depend on x. Using this fact and exploiting the re-written expression for E (eq. 1) we can obtain a simple expression for the marginal on x:

$$\begin{aligned} p(x) &= \frac{1}{Z}\int \exp\left(-\frac{1}{2}(y - Wx)^\top(y - Wx)\right) dy \\ &\qquad \cdot \exp\left(\frac{1}{2}x^\top W^\top Wx + f(x)\right) \\ &= \frac{(2\pi)^{n/2}}{Z}\exp\left(\frac{1}{2}x^\top W^\top Wx + f(x)\right) \end{aligned}$$

But this is just the Boltzmann distribution for the energy E' where $E'(x) = -x^\top W^\top Wx/2 - f(x)$.

3.2 SAMPLING FROM p(x)

We saw in the last section that the conditional $p(y|x)$ is a multivariate normal with mean Wx and covariance matrix I and so each unit can be sampled independently and in parallel. And depending on the form of $f(x)$, the conditional $p(x|y)$ may also be easy to sample from using a parallelizable algorithm. If this is the case then there is an efficient block-Gibbs method for sampling from the joint distribution $p(x, y)$, where we alternate between sampling all of the units of x conditioned on y and all of the units of y conditioned on x. The existence of this kind of block-Gibbs sampling algorithm is one of the main advantages of an RBM over a fully connected Boltzmann machine (Smolensky, 1986) and if we take $X = \{0, 1\}^n$ and $f(x) = d^\top x/2$ then $p(x, y)$ is in fact an RBM where y are "Gaussian units". Sampling from $p(x)$ then reduces to sampling from $p(x, y)$ (which is easy) and simply discarding the y-components.

3.3 BINARY-VALUED MRF

In this section we will show that for any binary-valued pairwise MRF there is a choice of f and X such that $p(x, y)$ is an RBM with Gaussian hidden units whose marginal $p(x)$ corresponds exactly to the MRF. Later, we will generalize this result beyond the binary case.

A general binary-valued pairwise MRF is defined by a Boltzmann distribution $p(x) \propto \exp(-E(x))$ over $X = \{0, 1\}^n$ with energy function $E(x) = -x^\top Ax/2$, where A is a symmetric matrix (there are other equivalent ways of parameterizing it, but this way is the most useful for our purposes). If we set

$$f(x) \equiv \frac{1}{2} d^\top x = \frac{1}{2} x^\top \mathrm{diag}(d) x$$

where $d \in \mathbb{R}^n$ then we have that $p(x|y) = \prod_{i=1}^n p(x_i|y)$ where

$$p(x_i = 1|y) = \sigma\left(W_{(:,i)}^\top y + \frac{1}{2} d_i\right)$$

and where σ is the logistic sigmoid function and $W_{(:,i)}$ denotes the i^{th} column of W.

This choice of f satisfies our first requirement, namely that each x_i can be sampled parallel from $p(x|y)$. Moreover, it can be easily seen that $p(x, y)$ is in fact an RBM with Gaussian units y and binary units x. It remains to choose a value of W and d so the marginal $p(x)$ corresponds to the Boltzmann distribution with the desired energy (i.e., our DPMRF). In particular we require that for all $x \in X$,

$$-\frac{1}{2} x^\top A x = -\frac{1}{2} x^\top W^\top W x - \frac{1}{2} x^\top \mathrm{diag}(d) x$$

This will be satisfied iff $A = W^\top W + \mathrm{diag}(d)$.

If A is positive definite we can simply take $d = \vec{0}$, $m = n$ and $W = \mathrm{chol}(A)$ where $\mathrm{chol}(\cdot)$ denotes the Cholesky decomposition. Otherwise we must choose d more carefully. In particular, if we choose d so that $A - \mathrm{diag}(d)$ is positive definite then we can take $W = \mathrm{chol}(A - \mathrm{diag}(d))$.

There can be many different choices for d, but one choice that will always work is $d = \alpha\vec{1}$ where $\vec{1}$ is the vector of ones, $\alpha = (1 + \epsilon)\min(\lambda_1, 0)$, ϵ is some small constant greater than 0, and λ_1 is the most negative eigenvalue of A. It turns out that the choice of d is important and will influence the mixing properties of the block-Gibbs chain on $p(x, y)$. We will examine this issue in a later section.

One important point to note is that if A is sparse (e.g. if the connections follow a lattice structure) then $W = \mathrm{chol}(A)$ will generally be sparse too, which is a useful property to have if n is very large and efficiency depends on sparsity. Even when A is dense, computing W is as hard as computing a Cholesky decomposition for which there are reasonably efficient algorithms.

3.4 MULTI-VALUED MRFS

In developing the procedure for determining d and W we made use of the fact that $X = \{0, 1\}^n$ so that $\frac{1}{2} d^\top x = \frac{1}{2} x^\top \mathrm{diag}(d) x$. However, the construction would also apply if instead we required only that $X \subseteq \{0, 1\}^n$ and in this section we show that for any DPMRF there is such a choice of X so that its energy may be written in the required form $-x^\top Ax/2$ and that $p(x, y)$ remains an easy-to-sample RBM, albeit one with "softmax units" instead of the standard binary units.

Suppose that the DPMRF is defined over the variables $s = (s_1, \ldots, s_m)$, each of which takes values in $\{0, \ldots, k - 1\}$. We will use a 1-of-k encoding for the variables s by representing it as an mk-dimensional vector x

$$x_{k \cdot j + \ell} = \delta(s_j, \ell)$$

where $0 \leq j \leq m - 1$ indexes the unit, and ℓ indexes the value of unit j. Thus every state s can be uniquely mapped to a state $x(s)$ using the 1-of-k encoding. The set of such valid x's is given by

$$X = \left\{ x \in \{0, 1\}^{m \cdot k} \;\middle|\; \sum_{\ell=0}^{k-1} x_{k \cdot j + \ell} = 1 \text{ for all } j \right\}$$

Having defined X we next show how to choose A. Suppose that the DPMRF distribution is defined by

$$q(s) \propto \prod_{i \neq j} \Phi_{ij}(s_i, s_j) \prod_{i=1}^n \Phi_i(s_i)$$

Then by setting $A_{ki+u,kj+v}$ to $\log \Phi_{ij}(u, v)/2$ if $i \neq j$, setting $A_{ki+u,ki+u}$ to $\log \Phi_i(u)$, and to zero if the potential Φ_{ij} is not provided, the resulting distribution $p(x) \propto \exp(x^\top Ax/2)$ is equivalent to the distribution $q(s)$ in the sense that $q(s) = p(x(s))$.

We can now apply the transformation described in the previous section to obtain a Boltzmann distribution $p(x, y)$ whose marginal distribution is equal to $p(x)$. Our choice of X makes sampling from the conditional distribution $p(x|y)$ easy, because it factorizes as the product

$$p(x|y) = \prod_{i=1}^m p\left(x_{(i-1)\cdot k}, \ldots, x_{i \cdot k-1} | y\right)$$

where each term is a simple multinomial distribution.

3.5 SEMI-RESTRICTED BOLTZMANN MACHINES

An interesting special case of the binary MRF, and one which we will devote most of our experiments to in this paper, is the semi-restricted Boltzmann Machine (SRBM) (Osindero and Hinton, 2008). The SRBM, like the standard RBM, introduces binary hidden units $h \in \{0, 1\}^p$ which are connected to each visible unit x but not to each other. But unlike the RBM, the visible units x of the SRBM are connected to each other, which prevents the use of block-Gibbs sampling in such a model.

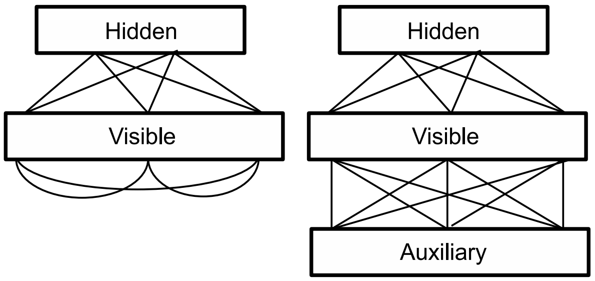

Figure 1: An SRBM (left) and an equivalent RBM (right)

The energy term of the SRBM can be written as:

$$E(x,h) = -\frac{1}{2}x^\top Ax - x^\top Mh - b^\top h$$

where the connection matrix between the visible units A is in $\mathbb{R}^{n\times n}$, the hidden-visible connection matrix M is in $\mathbb{R}^{n\times p}$, and the biases b are in $\mathbb{R}^p$.

The SRBM has a number of particularly appealing features for modeling densities of image-like data (Osindero and Hinton, 2008). Intuitively, the connections between the visible units model the “obvious” correlations between adjacent pixels, allowing the hidden units to use their capacity to model the more complex, higher order features of the model that are impossible to model with pairwise connections alone. As a result, the SRBM’s hidden-visible connections tend to look qualitatively different from corresponding connections in a standard RBM, because the RBM’s hidden-visible connections are forced to model the higher order statistics as well as the simpler pairwise correlations.

To achieve efficient sampling in SRBMs we may treat x and h as a homogeneous collection of variables and apply the previously developed conversion method to find an equivalent RBM, but it is more economical to apply the method only to x and its associated connection matrix A, leaving the h to x connections (i.e. M) unchanged. This transforms the SRBM into what is essentially a 3-layer RBM where the visible units x are in the *middle* layer (see figure (1)). That is, by taking $d = (1+\epsilon)\min(\lambda_1, 0)\vec{1}$ and $W = \text{chol}(A - \text{diag}(d))$ we obtain a Boltzmann distribution with energy

$$E(x,h,y) = -x^\top Mh - b^\top h - x^\top Wy + \frac{1}{2}x^\top \text{diag}(d)x$$

The marginal $p(x,h)$ of this distribution is equal to the joint distribution of the SRBM and the associated energy E has the exact form of the energy function of a 3-layer RBM with Gaussian units y. Fortunately, just as with a 2-layer RBM, there is an efficient block-Gibbs sampling algorithm for a 3-layer RBM. To see this note that the auxiliary units y are not connected to the hidden units h and so we can sample both independently given x. This allows us to alternate between sampling from $p(x|y,h)$ and sampling from both $p(h|x)$ and $p(y|x)$. The conditional $p(y|x)$ is computed as before and the remaining conditionals can be computed as $p(h|x) = \prod_i p(h_i|x)$ and $p(x|h,y) = \prod_i p(x_i|h,y)$, where

$$\begin{aligned} p(h_i = 1|x) &= \sigma\left(M_{(:,i)}^\top x + b_i\right) \\ p(x_i = 1|h,y) &= \sigma\left(M_{(i,:)}h + W_{(:,i)}^\top y + d_i/2\right) \end{aligned}$$

4 MODEL CONVERSION VERSUS DIRECT LEARNING

The equivalence results we have developed can be used in two basic ways. First, they can be used to achieve a much more paralellizable sampling algorithm by way of conversion to the equivalent RBM. For a DPMRF the conversion involves computing a Cholesky decomposition and thus can become computationally burdensome if we have to do it repeatedly, say in the case of learning where the parameters of the DPMRF constantly change.

A second and perhaps more interesting use of our results is to motivate the use of RBM-type models in place of traditional DPMRFs. The existence of the conversion from the original MRF parameterizations proves that the equivalent RBMs have the same expressive power. So, for example, instead of learning an SRBM and using our method to transform it to an equivalent RBM for the purposes of sampling, we could directly learn the parameters of the RBM as long as it has the general structure needed for the conversion result to hold.

5 EXPERIMENTAL RESULTS

5.1 DATASETS AND TRAINING ALGORITHMS

For the majority of our experiments we used three datasets: the USPS digit dataset, which consists of 10,000 16×16-images of handwritten digits, the MNIST digit dataset, which consists of 60,000 28×28 images of handwritten digits, and the MNORB dataset (used in Tieleman and Hinton, 2009), which is a binary 32×32 version of the NORB image dataset, consisting of 12,000 training images (LeCun et al., 2004).

The training algorithm we use is persistent-CD (PCD) (Tieleman, 2008; Younes, 1988), in which a set of negative particles is used as a persistent approximate sample from the model’s distribution, which is used for approximating the gradient of the model’s log probability. The resulting weight update is described in (Tieleman, 2008). The persistent particles are a sensible approximation to the model's distribution, because the model constantly updates them with a Markov chain that keeps the current model invariant. Although the current model changes as learning proceeds, the hope is that the persistent particles will be updated sufficiently quickly and remain a good approximation to the model's distribution throughout learning. It is known that if the learning rate is reduced at a certain rate, this algorithm locally converges to the maximum-likelihood setting of the parameters (Younes, 1988).

5.2 MIXING SPEED EXPERIMENTS

While no Markov chain for arbitrary binary MRFs can mix fast in a certain strong sense unless NP = RP (see Gore and Jerrum, 1999, for details), we can still discuss and compare mixing speed in a more practical sense. While sampling from $p(x, y)$ in the equivalent RBM can be done with easily parallelized block-Gibbs steps, we do not necessarily know how fast the chain will mix in practice. And in particular we do not know if it will mix nearly as fast (in terms of the number of passes over the complete set of variables) as a sequential Gibbs chain applied directly to the original MRF. In this section we will investigate the mixing speed of the block-Gibbs chain run on the equivalent RBM and show that it mixes only slightly slower (per step) than the SGS chain run on the original SRBM [1]. Thus block-Gibbs sampling in the equivalent RBM will be very cost effective on parallel computing architectures, which we demonstrate in the next section.

In section 3.3 we gave a conversion formula for computing the parameters of an the equivalent RBM for an arbitrary MRF. The formula depends on a constant α that is only required to be larger than $\alpha_{min} = \min(\lambda_1, 0)$. Our first experiment investigates the relationship between the mixing speed of the Gibbs chain on the RBM, and the choice of α used to construct it. In particular we use an SRBM model that was trained with PCD on the USPS dataset as the target MRF (the details of this training are given in section 5.4).

To quantify mixing speed we first ran 1024 parallel chains of sequential Gibbs for 10000 steps after 5000 steps of burn-in, in order to estimate a "ground truth" for the statistic $\mathbb{E}[hx^\top] - \mathbb{E}[h]\mathbb{E}[x]^\top$ (hereafter referred to as the hx-statistic). We chose this 2nd-order statistic because first-order statistics such as $\mathbb{E}[x]$ are too easily estimated even by poorly mixing chains, and because this statistic is arguably the most important one for learning. Then, for several different values of α, we computed the equivalent RBM and ran 1024 parallel Gibbs chains on it for 3000 steps (after 2000 steps of burn-in) and measured the relative difference between the current estimate of the hx-statistic and the ground truth.

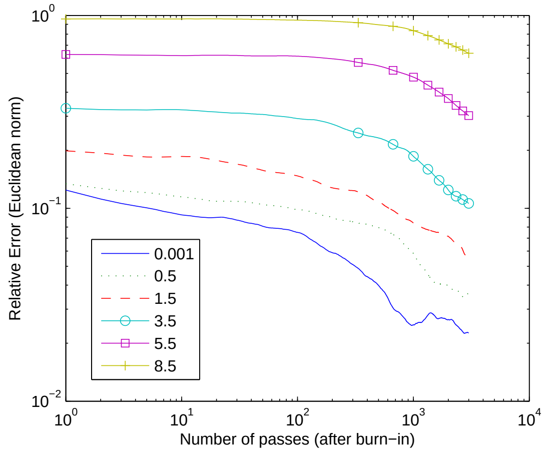

Figure 2: Error versus number of runs for the block-Gibbs chain running on the RBM constructed using $\alpha = (1 + \epsilon)\alpha_{min}$. The legend indicates the value of ϵ used.

The results of this experiment are given as Figure 2. Not too surprisingly, they clearly suggest that as we raise α the mixing performance of the RBM chain steadily degrades. Fortunately α_{min} tends to be reasonably small in practice. This well behaved nature of α may be due to some qualitative property of the A matrix when it models real data and it is something we plan to investigate in future work.

The second set of experiments we performed were designed to compare the mixing speed of SGS on an SRBM against block-Gibbs on the equivalent RBM. We trained models with both parameterizations and on two different datasets (for a total of 4). As before we computed the relative error of the hx-statistic as estimated by 1024 parallel chains. Since we could estimate ground truth using either sampling method (with a much longer chain) for each model we did both and found, reassuringly, that they were always within less than 3% relative error of each other. The results of these experiments are given as Figures 3 and 4.

In addition to measuring the speed of convergence of the hx-statistic estimates we also ran Markov chain analysis software (Cowles et al., 2006) to compute the "effective number of independent samples" produced by 15000 consecutive samples from each chain. Note that this method analyzes each dimension of the chain independently, producing individual estimates, over which we take the median to obtain a single number. These are reported in table 1.

[1] Here a "step" of SGS means a sequential sampling pass over all of the dimensions.

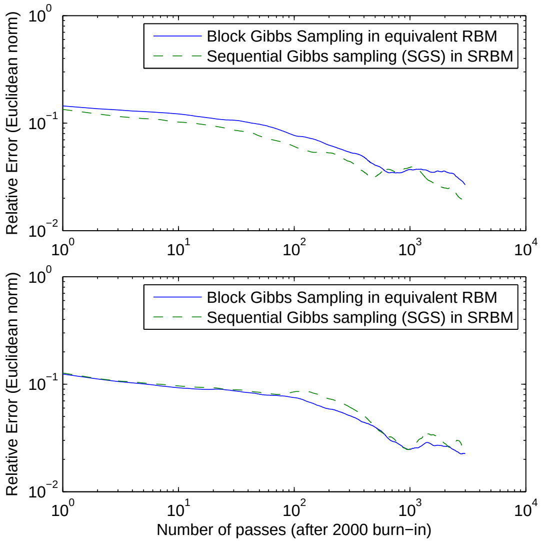

Figure 3: Relative estimation error (of the hx-statistic) versus step number for models trained on USPS with the SRBM parameterization (bottom) and with the equivalent RBM parameterization (top).

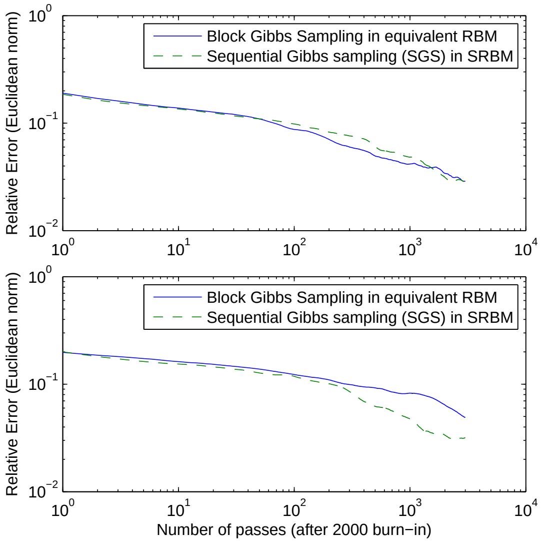

Figure 4: Relative estimation error (of the hx-statistic) versus step number for models trained on MNIST with the SRBM parameterization (bottom) and with the equivalent RBM parameterization (top).

Together, these experiments seem to indicate that the per-step mixing speed of the two methods is similar, with SGS being only moderately faster in some cases. However, since a single step of the RBM sampling method can be parallelized, the method will likely be much more cost-effective than SGS on parallel computing architectures.

Table 1: Median Effective Sample Size Estimated by CODA

Model, Dataset	**SGS**	**Block-Gibbs**
SRBM, USPS	264	153
Equiv. RBM, USPS	188	142
SRBM, MNIST	377	218
Equiv. RBM, MNIST	401	416

5.3 EVALUATING SWENDSEN-WANG

Our next experiment examines the possibility of using the Swendsen-Wang algorithm instead of SGS to sample from an SRBM. We used the same MNIST-trained SRBM from the previous experiment as our target model. Let a be the vector of auxiliary binary units introduced by SW. To sample from the SRBM using the SW algorithm we run a Gibbs chain, alternating between sampling h and a given x (as a block-step) and sampling x given h and a using the SW "cluster-and-flip" procedure with clusters defined by a and flip probabilities given by the input biases $M^T h$. Since the clustering procedure is computationally very expensive compared to SGS or block-Gibbs (even with a hybrid Matlab-C implementation we used) we only ran 5 parallel chains and assessed the speed of mixing by examining the quality of the samples. It was immediately apparent by simple inspection that the SW chains were mixing very slowly (per step) compared to the other methods and that the samples did not resemble anything close to the USPS digits until around the 1000th step. See Figure 5 for some typical samples produced by SW contrasted with ones produced by sequential Gibbs sampling of the SRBM and block-Gibbs sampling of the equivalent RBM.

The likely explanation for the failure of SW on learned RBM-type models is that they are highly frustrated MRFs with a very non-uniform structure and are qualitiatively very different from the sort of physics models for which SW was originally designed.

5.4 SRBM LEARNING EXPERIMENTS

In this section, through a series of experiments, we compare learning under the SRBM parameterization and the equivalent RBM parameterization.

In our experiments, we trained, via PCD, the parameters of an SRBM and those of an equivalent RBM on the same datasets and with the same learning-rate parameters. The training and test set log probabilities were estimated as training proceeded. For comparison, we also trained a standard RBM without visible-visible connections, to support the claim that such connections provide a real advantage in terms of modeling power on these datasets.

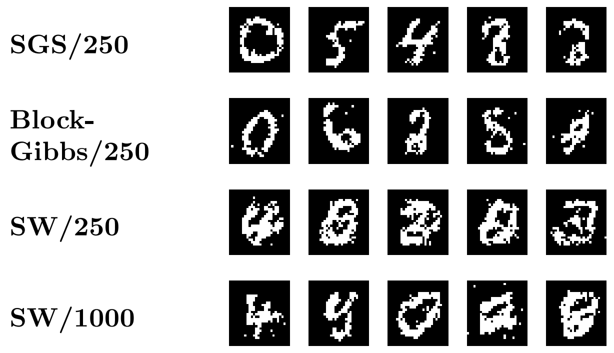

Figure 5: Typical samples produced by the 3 chains after the number of indicated steps of burnin.

We now describe details of our PCD training. For simplicity, we used the same learning parameters for all the models. The models were trained for 100,000 weight updates on batches consisting of 256 training examples and on 256 negative particles for PCD. The learning rate was kept at 0.01. The learning rate for the visible-visible connections in the SRBM or the visible-Gaussian connections in the equivalent RBM was 10 times smaller. We used 3 steps of the Gibbs sampling chain for updating the negative particles for the equivalent RBM.

The small USPS model had 256 visible units and 40 hidden units, while the large USPS model had 256 hidden units. The MNIST and the MNORB models had 784 and 1024 visible units, respectively, and both had 500 hidden units. We observed that overfitting was a significant problem on the MNORB dataset for all three models, which was likely the result of the training set and the test set being qualitatively different.

The training and the test log probability were estimated with Annealed Importance Sampling (Neal, 2001), where the chain was initialized with a "base-rate" model fitted to the training data (Salakhutdinov and Murray, 2008). We used 1024 AIS runs for 32,000 intermediate distributions, where 2000 equally spaced distributions were placed in the interval $[0, 0.5)$, 10000 were equally spaced in the temperature interval $[0.5, 0.9)$, and 20000 transitions were used in the interval $[0.9, 1.0]$.

Our GPU-based implementations used the Python library Cudamat (Mnih, 2009).

The results (fig. 6) demonstrate that models learned with the RBM parameterization can be as good as those learned with the standard SRBM parameterization in terms of log probability, with training that proceeds an order of magnitude faster on parallel computing architectures like GPUs due to the parallelizable nature of block-Gibbs.

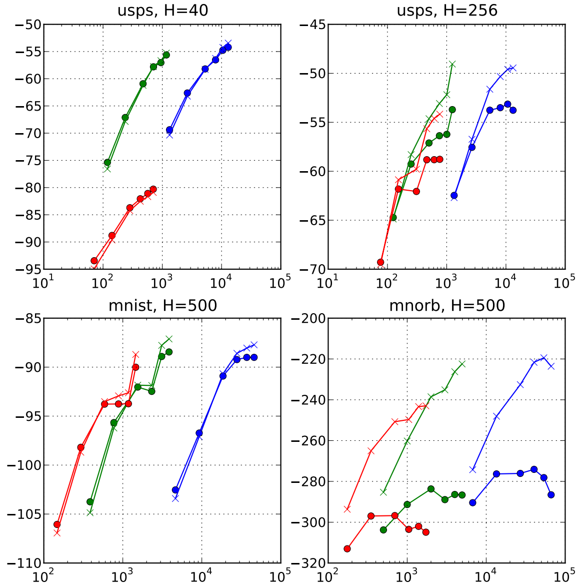

Figure 6: These plots show the training and test log probabilities of the different models as learning progresses. The blue curves gives the log probabilities for the SRBM, the green curves the equivalent RBM, and the red curves the standard RBM (without any lateral connections or special Gaussian units). Circles correspond to the test log probability and the x's correspond to the training log probability. The x-axis gives the learning time in seconds (log-scale), while the y-axis shows the log probability.

6 A FAILURE MODE

Our experience with the SRBM suggests that the SGS and the auxiliary variable method mix at roughly the same speed. However, in this section we present a simple pathological example where the SRBM can mix much faster than the auxiliary variable method. Let $t > 0$ be a large scalar, and let

$$P(x_1, x_2) \propto \exp(x_1 t^2/2 + x_2 t^2 - x_1 x_2 t^2)$$

be our MRF defined over $x \in \{0, 1\}^2$. Below is the equivalent model with the Gaussian hidden units,

$$P(x_1, x_2, y_1, y_2) \propto \exp(x_1 y_1 t - x_2 y_1 t + x_2 y_2 t - y_1^2/2 - y_2^2/2)$$

which can be seen to be equal to $P(x_1, x_2)$ when the hidden units y are marginalized.

By inspecting $P(x_1, x_2)$ and enumerating the four possible states, we can see that an overwhelming fraction of the probability mass lies at $(0, 1)$ (i.e., $x_1 = 0, x_2 = 1$) when t is large. Now, if we apply the sequential Gibbs chain to $P(x_1, x_2)$, we can see that all states quickly bring us to $(0, 1)$. If we perform sequential a Gibbs sweep by first updating x_1 and then x_2, then the states $(0, 0)$ and $(1, 0)$ are both taken to $(1, 0)$ or $(1, 1)$ with probability $\sim 1/2$, while the state $(1, 1)$ is taken to $(0,1)$ with overwhelming probability. Thus the SGS will quickly fixate on $(0,1)$.

However, we can show that block-Gibbs sampling on $p(x_1, x_2, y_1, y_2)$ keeps the low probability state $x = (1,0)$ unchanged with overwhelming probability, causing the chain to mix very slowly when it is initialized to $x = (1,0)$ and t is large. In particular, the probability of transitioning away from this state after Gibbs sampling $y \sim N((t,0), I)$ is $1 - \sigma(ty_1)\sigma(ty_1 - ty_2) = 1 - \sigma(t^2 + tn_1)\sigma(t^2 + tn_1 - tn_2)$ where $n_1, n_2 \sim N(0,1)$. For large t there is a high probability this will be closely approximated by $1 - \sigma(t^2)\sigma(t^2)$ which will be very close to 0.

Finally, while this example demonstrates the existence of parameter settings where the auxiliary-variable sampling method performs significantly worse than SGS, we have found that in practice, when applied to models that have been learned on real data such as in the previous section, such performance gaps do not seem to arise. Moreover, the natural annealing which takes place during PCD learning may help to ensure that the chain doens't easily get stuck in the "bad" states.

7 CONCLUSIONS

In this paper, we showed that the distribution of any pairwise discrete MRF can be represented as the marginal distribution of a Restricted Boltzmann Machine with Gaussian hidden units. Our results show that the resulting block-Gibbs chain is comparable to sequential Gibbs in terms of mixing speed, while being readily parallelizable and thus more efficient. Our results also demonstrate the usefulness of an alternative learning strategy which is motived by the equivalence result: learning the parameters of special RBMs, augmented with Gaussian hiddens units, in place of harder-to-sample-from MRFs such as SRBMs. The simplicity of the conversion proceedure between RBMs and MRFs, combined with the parallelizability of sampling with RBMs, makes these techniques for sampling and learning of MRFs very suitable for practical applications.

ACKNOWLEDGMENTS

We are grateful to Iain Murray for his helpful advice. This work was funded by NSERC and the University of Toronto.

References

M.K. Cowles, N. Best, K. Vines, and M. Plummer. R-CODA 0.10-5, 2006. Available from http://www-fis.iarc.fr/coda/.

G.R. Cross and A.K. Jain. Markov random field texture models. In *Conference on Pattern Recognition and Image Processing, Dallas, TX*, pages 597–602, 1981.

V.K. Gore and M.R. Jerrum. The Swendsen–Wang process does not always mix rapidly. *Journal of Statistical Physics*, 97(1):67–86, 1999.

G.E. Hinton and R.R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313 (5786):504–507, 2006.

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

H. Larochelle and Y. Bengio. Classification using discriminative restricted boltzmann machines. In *ICML*, 2008.

Y. LeCun, F. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004.

M. Malfait and D. Roose. Wavelet-based image denoising using a Markov random field a priorimodel. *IEEE Transactions on Image Processing*, 6(4):549–565, 1997.

V. Mnih. Cudamat: a CUDA-based matrix class for python. Technical Report UTML TR 2009-004, Department of Computer Science, University of Toronto, November 2009.

R.M. Neal. *Probabilistic inference using Markov chain Monte Carlo methods.* 1993.

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

S. Osindero and G. Hinton. Modeling image patches with a directed hierarchy of Markov random fields. In *NIPS*, 2008.

J. Portilla, V. Strela, M.J. Wainwright, and E.P. Simoncelli. Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Transactions on Image Processing*, 12(11):1338–1351, 2003.

S. Roth and MJ Black. Fields of experts: A framework for learning image priors. In *CVPR 2005*, volume 2, 2005.

R. Salakhutdinov and I. Murray. On the quantitative analysis of deep belief networks. In *ICML*, 2008.

N.N. Schraudolph and D. Kamenetsky. Efficient exact inference in planar Ising models. *NIPS*, 2009.

P. Smolensky. Information processing in dynamical systems: Foundations of harmony theory. *Parallel distributed processing: Explorations in the microstructure of cognition*, 1:194–281, 1986.

T. Tieleman. Training restricted Boltzmann machines using approximations to the likelihood gradient. In *ICML*, 2008.

T. Tieleman and G.E. Hinton. Using Fast Weights to Improve Persistent Contrastive Divergence. In *ICML*, 2009.

M.J. Wainwright. *Graphical Models, Exponential Families, and Variational Inference.* Now Publishers, 2008.

M.J. Wainwright and E.P. Simoncelli. Scale mixtures of Gaussians and the statistics of natural images. In *NIPS*, 2000.

L. Younes. Estimation and annealing for Gibbsian fields. *Ann. Inst. H. Poincare Probab. Statist*, 24:269–294, 1988.