CPSC 538R Project Report

Lifted Parallel Tempering for Training Restricted Boltzmann Machines

Amit Kadan, Saifuddin Syed, Esten Nicolai Wøien

Submitted to
Dr. Siamak Ravanbakhsh
April 20, 2018

Contents

1 Introduction

Restricted Boltzmann Machines (RBMs) are Markov random fields with applications including density estimation, dimensionality reduction and collaborative filtering.

2 Restricted Boltzmann Machines

An RBM is a Markov random field where the underlying graph is a complete bipartite graph separating visible units $\mathbf{v} = (v_1, \ldots, v_n) \in \{0,1\}^n$ from the latent hidden units $\mathbf{h} = (h_1, \ldots, h_m) \in \{0,1\}^m$ as shown in Figure 2.

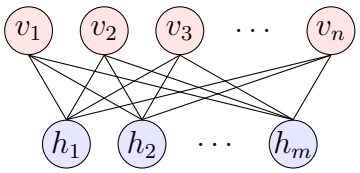

Figure 1: Graphical representation of an RBM.

Furthermore an RBM is assumed to have the Gibbs distribution given by

$$P(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} \exp(-E(\mathbf{v}, \mathbf{h})), \tag{1}$$

where the energy $E(\mathbf{v}, \mathbf{h})$ has the form,

$$\begin{aligned} E(\mathbf{v}, \mathbf{h}) &= -\sum_{i=1}^{n}\sum_{j=1}^{m} v_i w_{ij} h_j - \sum_{i=1}^{n} b_i v_i - \sum_{j=1}^{m} c_j h_j \\ &\equiv -\mathbf{v}^T \mathbf{W} \mathbf{h} - \mathbf{b}^T \mathbf{v} - \mathbf{c}^T \mathbf{h}, \end{aligned}$$

for some $\mathbf{W} \in \mathbb{R}^{n \times m}$, $\mathbf{b} \in \mathbb{R}^n$, and $\mathbf{c} \in \mathbb{R}^m$. We have $w_{i,j}$ encodes the interaction potential between v_i, and h_j and the b_i, c_j encodes the local potential for v_i and h_j respectively.

2.1 RBM Likelihood

Suppose we have data $\mathcal{D} = \{\mathbf{v}^1, \ldots, \mathbf{v}^\ell\}$, we want to find $\mathbf{W}, \mathbf{b}, \mathbf{c}$, that maximize the log-likelihood function given by

$$\mathcal{L}(\theta|\mathcal{D}) = \log P(\mathcal{D}|\theta)$$

As shown in §4.2 in [FI14]), the gradient of $\mathcal{L}$ can be written as,

$$\frac{\partial \mathcal{L}}{\partial \theta} = \mathbb{E}_{P_{data}}\left(\frac{\partial E}{\partial \theta}\right) - \mathbb{E}_{P}\left(\frac{\partial E}{\partial \theta}\right) \tag{2}$$

where P_{data} is the empirical distribution of the data. So by taking the patial derivatives with respect to w_{ij}, b_i, c_j, we get the gradient updates given by,

$$\frac{\partial \mathcal{L}}{\partial w_{ij}} = \mathbb{E}_{P_{data}}(v_i h_j) - \mathbb{E}_P(v_i h_j)$$
$$\frac{\partial \mathcal{L}}{\partial b_i} = \mathbb{E}_{P_{data}}(v_i) - \mathbb{E}_P(v_i)$$
$$\frac{\partial \mathcal{L}}{\partial c_j} = \mathbb{E}_{P_{data}}(h_j) - \mathbb{E}_P(h_j)$$

Hinton [Hin02] showed that this update also emerges by minimizing the KL divergence between the data distribution and the equilibrium distribution over the visible variables.

Therefore in order to optimize for our model parameters, we need to compute the above expectations, which in general are intractable. We thus resort to approximating them using MCMC. The natural MCMC method for us to use is Gibbs sampling as the bipartite structure of our graphical model tells us that $v_i \perp\!\!\!\perp v_k \mid \mathbf{h}$ and $h_j \perp\!\!\!\perp h_l \mid \mathbf{v}$. Thus we get the following conditional distributions

$$P(v_i = 1 \mid \mathbf{h}) = \sigma\Big(b_i + \textstyle\sum_j w_{ij} h_j\Big), \tag{3}$$
$$P(h_j = 1 \mid \mathbf{v}) = \sigma\Big(c_j + \textstyle\sum_i w_{ij} v_i\Big), \tag{4}$$

where $\sigma(x) = (1 + e^{-x}))^{-1}$ is the sigmoid function.

3 Contrastive Divergence

Contrastive Divergence (CD) is the standard way for training RBMs [Hin02]. CD approximates $\mathbb{E}_P\left(\frac{\partial E}{\partial \theta}\right)$ with $\langle \frac{\partial E}{\partial \theta} \rangle^k$, the Monte-Carlo average after sampling the hidden and visible units using k Gibbs Sampling updates.

CD begins by initializing the visible units to a given data vector. Then, in what is known as a "negative phase", the hidden units are updated via Gibbs sampling, with the visible units clamped. This is followed by a "positive" phase, where the visible units are updated with Gibbs sampling, while the hidden units are clamped using

$$P(h_j = 1 \mid \mathbf{v}) = \sigma\Big(c_j + \textstyle\sum_i w_{ij} v_i\Big).$$

The resulting state of the visible and hidden units gives us the sample $\langle \frac{\partial E}{\partial \theta} \rangle^0$. One can repeat this process, updating the visible and hidden units one layer at a time to obtain the estimate $\langle \frac{\partial E}{\partial \theta} \rangle^k$ for any arbitrary k [Hin12]. When $k > 1$, we will call this algorithm CD-k.

To obtain an unbiased estimator of $\mathbb{E}_P\left(\frac{\partial E}{\partial \theta}\right)$, this process must run to equilibrium, requiring many iterations. However, this is computationally expensive. Hinton [Hin02] notes that after 1 iteration, it can be seen in which direction the model is wandering. Hinton proposes the simplified learning rule

$$\Delta\theta = \left\langle \frac{\partial E}{\partial \theta} \right\rangle^0 - \left\langle \frac{\partial E}{\partial \theta} \right\rangle^1$$

CD guarantees that weights are updated in the correct direction, albeit with wrong magnitudes [Tie08].

By using a small k, one gets biased estimates of the probability distribution underlying the model. Thus, one does not necessarily reach a minimum of the negative log likelihood during training. More problematically, Fischer and Igel [**?**] showed that the log-likelihood can even decrease as more iterations of gradient ascent are taken.

3.1 PCD

Although CD works for simple models, it can lead to bad models if the mixing rate of the Markov Chain is low. Tieleman [Tie08] notes that each time a Markov Chain is reinitialized with a data vector to approximate $\mathbb{E}_P\left(\frac{\partial E}{\partial \theta}\right)$, valuable information about the model is lost. Tieleman proposes the use of persistent chains. Rather than being initialized to a data vector, chains "persists", beginning with the final state of a previous Markov Chain. This variant of contrastive divergence is called Persistent contrastive divergence (PCD).

However, Tieleman, and Hinton [TH09] note that PCD still relies on approximating the gradient of the KL divergence. PCD minimizes a difference of KL divergences

$$KL(P_{data} \mid\mid P) - KL(P^t \mid\mid P),$$

where P_t is the distribution of persistent Markov chain after time t. The second term is an error incurred for not running the chain to equilibrium.

Desjardins [DCB+10] notes that although the error term is small relative to the desired term in the beginning of training, as t increases, the error term dominates due to the desired term vanishing, and the mixing of the chain decreasing. PCD encourages the model to settle to recently visited modes in successive iterations, leading to few deep minima in the energy.

4 Tempering Methods

As we have discussed previously, the problem of learning the model for an RBM can be boiled down to effectively we can approximate $\mathbb{E}_P\left(\frac{\partial E}{\partial \theta}\right)$. In order to do this, we had to resort to Gibbs Sampling. Gibbs sampling performs local updates to generate a Markov Chain with stationary distribution $P(\mathbf{v}, \mathbf{h})$ which is complex and multi-modal. Despite the theoretical guaranteed convergence of Gibb's sampling, the local nature of the updates means that chain can have a very difficult time traversing the modes of P and has a tendency to get trapped exploring the local areas of our sample space of high probability. This means that certain modes are over-represented, while other are under-represented and leading to a poor approximation of our expectation.

Parallel tempering (PT) is a class of Monte Carlo methods first introduced by Geyer in [Gey91] that improve the miximng our our MCMC chain, by simultaneuously running N MCMC chains in parallel but at different temperatures. The higher temperature chains focus on exploration of the state space, while the room temperature chain focus on the accuracy of samples. In order for there to be communication between the hot and room temperature chains, we periodically propose swaps between the chains in such a what that does not propose

4.1 Parallel Tempering

Let P be a Gibbs distribution over some sample space Ω of the form,

$$P(x) = \frac{1}{Z}\exp(-E(x)).$$

Note that E can be that of an RBM but does not have to be. We define our inverse-temperature space as $\mathcal{B} = [\beta_{\min}, 1]$, for $0 \leq \beta_{\min} < 1$. Given $\beta \in \mathcal{B}$, let $P^{(\beta)}$ the probability distribution identified with the un-normalized density $P(x)^\beta$, which we can write

$$P^{(\beta)}(x) = \frac{1}{Z(\beta)}\exp(-\beta E(x)),$$

where $Z(\beta)$ is the partition function.

Given a sequence of inverse temperatures $0 \leq \beta_{\min} = \beta_N < \cdots < \beta_1 < \beta_0 = 1$, we define the measure $\tilde{P}$ on Ω^{N+1} by,

$$\tilde{P} = P^{(\beta_0)} \times \cdots \times P^{(\beta_N)}.$$

In parallel tempering we construct a Markov chain $\tilde{X}_n = (X_n^0, \cdots, X_n^N)$ on Ω^N with stationary distribution $\tilde{\pi}$. We update X_n by alternate between two different type of dynamics: exploration and communication.

During the exploration phase, we allow for each X_n^i to independently explore Ω according to $P^{(\beta_i)}$ via our favourite MCMC algorithm such as Gibbs Sampling. Since MCMC updates of each component i leaves $P^{(\beta_i)}$ invariant, we have the exploration phase leaves $\tilde{P}$ invariant.

During the communication phase, we propose a swap between the β_i and β_j components of $\tilde{X}_n$, in other words we swap X_n^i and X_n^j in $\tilde{X}_n$. To simplify notation, given $x = (x^0, \cdots, x^i, \cdots, x^j, \cdots, x^N)$, we will define $x_{(i,j)} = (x^0, \cdots, x^j, \cdots, x^i, \cdots, x^N)$. So the proposed state during the communication phase is $(\tilde{X}_n)_{(i,j)}$. In order to keep $\tilde{X}_n$ stationary with respect to $\tilde{P}$, we can make this update reversible by accepting this proposed swap according to the Metropolis acceptance,

$$\begin{aligned}
\alpha &= 1 \wedge \frac{\tilde{P}((\tilde{X}_n)_{(i,j)})}{\tilde{P}(\tilde{X}_n)} \\
&= 1 \wedge \frac{P^{(\beta_i)}(X_n^j)P^{(\beta_j)}(X_n^i)}{P^{(\beta_i)}(X_n^i)P^{(\beta_j)}(X_n^j)} \\
&= 1 \wedge \exp\left[(\beta_j - \beta_i)(E(X_n^j) - E(X_n^i))\right] \qquad (5)
\end{aligned}$$

Since the likelihood of acceptance is proportional to the change in temperature, we usually restrict to our swaps to nearest neighbours during the communication phase.

We refer to the sequence $(X_n^i)_{n\geq 0}$ as the i-th chain, the sequence of states at a particular temperature β_i. During the communication phase, we can view a proposed state as a swap between two states at while fixing the temperature. Alternatively we can view the communication phase as increase or decreasing the temperature of the state. We will call this sequence of states with varying temperatures as a replica. In particular, the i-th replica is the sequence $(X_n^{\eta_n^i})$, where $\eta_0^i = i$ and η_n^i represents the temperature of state X_n at time n. The i-th replica follows the trajectory of the X_0^i state through the state space as it's temperature changes. A change in η_n^i indicates a successful swap in the communication phase at time n.

4.1.1 Temperature spacing

The mixing of our MCMC algorithm during the exploration phase improves as β decreases as $P^{(\beta)}(x)$ "flattens" our and makes it modes less pronounces. The temperature spacing is somewhat important as we see the acceptance probability of a proposed swap depends on the change in inverse-temperature as seen in (5). So if the change in inverse-temperature during a swap is small, we are far more likely to accept, but we will not be exploring a significantly better mixing chain. Alternatively, if the change in inverse-temperature is large, we will potentially swap with a better mixing chain, but the acceptance ration in (5) will be potentially very small. We want to chose the temperature spacing in such a way that maximizes the speed we can communicate information from the high temperature chains, to the room temperature chains.

In the high-dimensional setting, the literature seems to suggest that we want approximately to be roughly 23.4% [ARR11]. However to achieve this optimate temperature spacing, is in general quite difficult to attain and is more of a theoretical guideline. In practice, it can be shown that a geometric spacing can be a good temperature spacing for many applications of interest [ARR11], [Kof02] $\beta_n = \gamma^n$ for some $\gamma \in (0, 1)$.

4.1.2 Problems of Reversibility

In the classical parallel tempering algorithm, because of detailed balance condition for the swaps, the replicas must be proposed a swap that can potentially raise or lower the temperature. This forces the temperature of the replica to traverse the inverse-temperature space $\mathcal{B}$ in a random walk fashion and leads to diffusive behaviour. It takes roughly $O(N^2)$ number of swap attempts to communicate information between N chains [DHN00]. This is not favourable as the goal is to have the information from the high temperature states reach room temperature as quickly as possible, without getting distracted along the way.

In order to eliminate this diffusive behaviour and speed up the rate of communication between states, we need to construct swap proposals that are non-reversible. We will discuss two such methods called lifted parallel tempering (LPT) outlined in [Wu17] and the closely related deterministic even/odd algorithm which we will refer to as LTPD as discussed in [LDMT09].

4.2 Lifted Parallel Tempering

Before we discuss LPT, we will need to introduce the Generalized Metropolis Hastings framework as outlined in [SR$^+$10].

4.2.1 Generalized Metropolis-Hastings Framework

Suppose we have a probability measure P over measurable space $(\Omega, \mathcal{F})$ and a function $S : (\Omega, \mathcal{F}) to (\Omega, \mathcal{F})$ is a measure preserving involution for P. If $Q(y|x)$ is a Metropolis-proposal kernel, then we will construct a Markov transition kernel as follows. Suppose we are at state x, we propose a state y via the kernel $\tilde{Q}$ given by $\tilde{Q}(y|x) = Q(S(dy)|x)$ and

accept y as a new state with probability

$$\alpha(x,y) = \frac{\pi(y)\tilde{Q}(y|x)}{\pi(x)\tilde{Q}(x|y)}.$$

This induces the reversible Metropolis transition kernel,

$$\tilde{K}(y|x) = \alpha(x,y)\tilde{Q}(y|x) + \left(1 - \int \alpha(x,y)\tilde{Q}(y|x)dy\right)\delta_x(y).$$

which has stationary distribution P. Since $\tilde{K}$ and S preserve P, then so does $K \equiv S \circ \tilde{K}$, which is given by,

$$K(y|x) = \alpha(x,y)Q(y|x) + \left(1 - \int \alpha(x,y)Q(y|x)dy\right)\delta_{S(x)}(y).$$

Effectively, we make a proposal according to Q, which is accepted with probability α and apply S if the proposal is rejected. This in general produces an non-reversible scheme as the composition of two reversible kernels is not necessarily reversible.

4.2.2 Lifted Parallel Tempering Algorithm

Suppose we have the same set-up as PT outlined in §4.1. The only modification we will make for LPT will be for the communication phase.

We have N additional chains running simultaneously with joint distribution $\tilde{P} = \pi^{(\beta_0)} \times \cdots \times \pi^{(\beta_N)}$. We denote the state at time n by $X_n = (X_n^0, \ldots, X_n^N)$, where X^i is at temperature β_i. We will choose to make one of the replicas "lifted", say replica η_0. In addition we introduce the lifted parameters $(\eta_n, \varepsilon_n) \in \{0, 1, \ldots, N\} \times \{\pm 1\}$. Here η_n represents temperature index of the η_0-th replica (which we will call the "lifted" replica) and ε_n represents the direction the replica will attempt to swap, i.e, the next proposed swap will be between $X_n^{\eta_n}$ and $X_n^{\eta_n+\varepsilon_n}$.

To deal with the boundary cases, note that the replica is at temperature β_0 or β_N then it must be proposed a swap with the state at temperature β_1 and β_{N-1} respectively. So we have

$$(\eta_n, \varepsilon_n) \in \{0, 1, \ldots, N\} \times \{\pm 1\} \backslash \{(0,-1), (N,1)\} \equiv \mathcal{X}.$$

Let $Z_n = (X_n, \eta_n, \epsilon_n)$ be a Markov chain in enlarged space will be $\Omega^{N+1} \times \mathcal{X}$. Suppose we are working with the deterministic proposal

$$Q(y|z) = \delta_{\phi(z)}(y)$$

given by,

$$\phi(X, \eta, \varepsilon) = \begin{cases} (X_{(\eta,\eta+\varepsilon)}, \eta + \varepsilon, \varepsilon) & 0 < \eta + \varepsilon < N, \\ (X_{(\eta,\eta+\varepsilon)}, \eta + \varepsilon, -\varepsilon) & \text{Otherwise.} \end{cases}$$

We also define the involution $S : \Omega^{N+1} \times \mathcal{X}$ given by,

$$S(X, \eta, \varepsilon) = \begin{cases} (X, \eta, -\varepsilon) & 0 < \eta + \varepsilon < N, \\ (X, \eta, \varepsilon) & \text{Otherwise,} \end{cases}$$

which preserves the measure $\tilde{\pi}^P \times \text{Unif}(\mathcal{X})$. Together Q and S give us the recipe for the lifted parallel tempering algorithm.

If a swap is accepted and increases (decreases) the temperature of the replica, then the next swap will also be propose an increase (decrease) in temperature. If the swap is rejected, then the next swap will propose a decrease (increase) in temperature. If the replica achieves a temperature of either reaches β_0 or β_N, then we reverse the direction of the swaps. To summarize in LPT, the lifted replica is proposes swaps with a "momentum" (i.e. the communication resists change of direction). It was shown in [DHN00] that this persisted swapping required $O(N)$ number of swap attempts on average to swap information between N chains compared to the $O(N^2)$ number of swaps required for PT.

4.3 Deterministic Even/Odd Algorithm

Finally we will introduce one more variant of PT that has become quite popular in practice recently call the deterministic even/odd algorithm as outlined in (LPTD).

4.4 PT for RBM

5 Experiments

6 References

[ARR11] Yves F. Atchad, Gareth O. Roberts, and Jeffrey S. Rosenthal. Towards optimal scaling of metropolis-coupled Markov chain Monte Carlo. *Statistics and Computing*, 21(4):555–568, October 2011.

[DCB+10] Guillaume Desjardins, Aaron Courville, Yoshua Bengio, Pascal Vincent, and Olivier Delalleau. Tempered markov chain monte carlo for training of restricted boltzmann machines. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 145–152, 2010.

[DHN00] Persi Diaconis, Susan Holmes, and Radford M Neal. Analysis of a nonreversible markov chain sampler. *Annals of Applied Probability*, pages 726–752, 2000.

[FI14] Asja Fischer and Christian Igel. Training restricted boltzmann machines: An introduction. *Pattern Recognition*, 47(1):25–39, 2014.

[Gey91] Charles J Geyer. Markov chain monte carlo maximum likelihood. 1991.

[Hin02] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[Hin12] Geoffrey E Hinton. A practical guide to training restricted boltzmann machines. In *Neural networks: Tricks of the trade*, pages 599–619. Springer, 2012.

[Kof02] David A Kofke. On the acceptance probability of replica-exchange monte carlo trials. *The Journal of chemical physics*, 117(15):6911–6914, 2002.

[LDMT09] Martin Lingenheil, Robert Denschlag, Gerald Mathias, and Paul Tavan. Efficiency of exchange schemes in replica exchange. *Chemical Physics Letters*, 478(1-3):80–84, 2009.

[SR+10] Gabriel Stoltz, Mathias Rousset, et al. *Free energy computations: A mathematical perspective*. World Scientific, 2010.

[TH09] Tijmen Tieleman and Geoffrey Hinton. Using fast weights to improve persistent contrastive divergence. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1033–1040. ACM, 2009.

[Tie08] Tijmen Tieleman. Training restricted boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th international conference on Machine learning*, pages 1064–1071. ACM, 2008.

[Wu17] Wu. Irreversible Parallel Tempering, 2017.