RBM-RELATED

A PREPRINT

Hao Ma

July 9, 2019

***K*eywords** Ising Model, Markov Chain Monte Carlo, Restricted Boltzmann Machine · Gibbs Sampling · Energy Model · Markov Random Field

1 Markov Chain Monte Carlo simulations for the Ising Model

1.1 Ising Model

The Ising model is a formalized stochastic model of ferromagnet (i.e., an ordinary magnet) and is a d-dimensional lattice that can be denoted as the set below

$$I = \Big\{(x_1, x_2, \dots, x_d) \in \mathbb{Z}^d : 1 \leq x_l \leq L, l = 1, 2, \dots, d\Big\}. \tag{1}$$

Each site can be considered as an iron atom if I is an iron magnet. Specifically, a 2-dimensional Ising model contains L^2 sites which are evenly distributed in a square lattice.

A *spin configuration* on I is a function

$$\sigma : I \to \Big\{\pm 1\Big\} \tag{2}$$

which assigns a spin up (+1) or a spin down (-1) to each site in I.

Hamiltonian $\mathcal{H}(\sigma)$ is a function which specifies the total energy of a spin configuration. In the absence of an external magnetic field, it is defined as

$$\mathcal{H}(\sigma) = -J \sum_{x,y} \sigma_x \sigma_y \tag{3}$$

where J is the *coupling constant*, and σ_x and σ_y represent the spin of site x and site y, respectively, and the summation is taken over all pairs of (x, y) where site x and y are neighbours. Two sites are neighbours if exactly one coordinate of two sites differ by 1.

The probability of a spin configuration follows the *Boltzmann distribution*:

$$p(\sigma) = \frac{1}{Z} \exp(-\frac{\mathcal{H}(\sigma)}{k_B T}) \tag{4}$$

where k_B is the Boltzmann constant and T is the absolute temperature. Z is a normalizing constant known as the *partition function* and is defined as

$$Z = \sum_{\sigma} \exp(-\frac{\mathcal{H}(\sigma)}{k_B T}) \tag{5}$$

1.2 Markov Chain Monte Carlo Simulations

The number of configuration in the Ising model is exponential of the number of sites, which makes the calculation of the normalizing constant Z intractable for a large L. Therefore, we often study the Ising model using Monte Carlo simulations. If we could sample N spin configurations from the Boltzmann distribution, then the expectation value of an observable $\mathcal{O}$ can be approximated by

$$\mathbb{E}(\mathcal{O}) = \frac{1}{N}\sum_{i}^{N}\mathcal{O}(\sigma^i) \tag{6}$$

where σ^i is the i_{th} spin configuration sample.

In spite of the computational intractability to compute the partition function Z, we can still sample the Boltzmann distribution using Markov Chain Monte Carlo (MCMC) techniques, specifically, *Metropolis Algorithm*, which is a special case of the *Metropolis-Hasting Algorithm*. For this specific problem, the algorithm goes like below:

1. Initialize σ^0. A common practice is to independently and randomly assign +1 or -1 to each site.
2. Randomly choose a site x.
3. Let σ' be the spin configuration obtained by the flipping the spin at x of the current spin configuration σ , i.e., $\sigma'_x = -\sigma_x$ and $\sigma'_y = -\sigma_y$ for all $y \neq x$.
4. Let U_n be a standard uniform random variable.
5. If $U_n \leq \frac{p(\sigma')}{p(\sigma)} = \exp\Big(-\frac{\mathcal{H}(\sigma')-\mathcal{H}(\sigma)}{k_B T}\Big)$, then accept the spin flip and set $\sigma^{n+1} = \sigma'$. Otherwise, reject the spin flip and set $\sigma^{n+1} = \sigma^n$.
6. $n = n + 1$ and return to Step 2.

After we obtain a sequence of spin configurations, we will sub-sample those configurations every τ steps to make sure that the samples are i.i.d samples. The result of the sub-sampling will be a sequence of spin configurations that contain

$$\sigma^T, \sigma^{T+\tau}, \sigma^{T+2\tau}, \dots \tag{7}$$

where T and τ are known as the *burn-in period* and the *sampling period*.

2 A Concise Tutorial for Restricted Boltzmann Machine

A Restricted Boltzmann Machine (RBM) is a bipartite graph. The structure of a RBM that contains 3 hidden nodes and 3 visible nodes is illustrated below.

In the following, we consider a RBM that contains V visible nodes and H hidden nodes, where every visible node is connected with every hidden node, with no connection among each layer. In this tutorial, we focus on Bernoulli RBM where the value of each node is either 0 or 1.

RBM is a theoretical model inspired by statistical mechanics. In this model, the value of $\mathbf{v} = [v_1, v_2, \dots, v_V]$ and $\mathbf{h} = [h_1, h_2, \dots, h_H]$ satisfy Boltzmann (Gibbs) distribution, which claims that the probability of a system staying in a certain micro-state is the function of the system energy and the system temporature. For the RBM model, the probability distribution of $\mathbf{v}$ and $\mathbf{h}$ can be expressed as:

$$p_\theta(\mathbf{v}, \mathbf{h}) = \frac{1}{Z}\exp(-E(\mathbf{v}, \mathbf{h})) \tag{8}$$

where $\theta = \{\mathbf{W}, \mathbf{b}, \mathbf{c}\}$, and the normalization constant $Z = \sum_{\mathbf{v},\mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}))$ is referred to as the *partition function*, and $E(\mathbf{v}, \mathbf{h})$ is the energy function calculated by

$$E(\mathbf{v}, \mathbf{h}) = -\mathbf{h}^T\mathbf{W}\mathbf{v} - \mathbf{b}^T\mathbf{v} - \mathbf{c}^T\mathbf{h} = -\sum_j\sum_k h_j W_{j,k} v_k - \sum_k b_k v_k - \sum_j c_j h_j \tag{9}$$

where $\mathbf{c}$ and $\mathbf{b}$ are the visible and hidden bias terms, respectively. Note here that Z is intractable to compute for a large RBM model.

2.1 Training

The first question that we need to ask is what the objective function is. We can look at this from two perspectives:

- Kullback-Leibler (K-L) Divergence
- Maximum Likelihood

K-L divergence is a measurement of distance between two probability distributions. In RBM, we would like to minimize the K-L divergence between the unknown distribution of the training data $p_{\mathbf{train}}(\mathbf{v})$ and the distribution represented by the RBM $p_\theta(\mathbf{v})$:

$$\mathrm{KL}(p_{\mathbf{train}} \| p_\theta) = \sum_{\mathbf{v}} \left(p_{\mathbf{train}}(\mathbf{v}) \log \frac{p_{\mathbf{train}}(\mathbf{v})}{p_\theta(\mathbf{v})} \right) = \sum_{\mathbf{v}} p_{\mathbf{train}}(\mathbf{v}) \log p_{\mathbf{train}}(\mathbf{v}) - \sum_{\mathbf{v}} p_{\mathbf{train}}(\mathbf{v}) \log p_\theta(\mathbf{v})$$

So

$$\begin{aligned} \min_\theta \mathrm{KL} &\iff \min_\theta \sum_{\mathbf{v}} p_{\mathbf{train}}(\mathbf{v}) \log p_\theta(\mathbf{v}) \\ &\iff \max_\theta \frac{1}{N} \sum_{\hat{\mathbf{v}}_i} \log p_\theta(\hat{\mathbf{v}}_\mathbf{i}) \text{ , } \hat{\mathbf{v}}_i \text{ is a sample from } p_{\mathbf{train}}(\hat{\mathbf{v}}) \\ &\iff \max_\theta \prod_{\hat{\mathbf{v}}_i} p_\theta(\hat{\mathbf{v}}_i) \end{aligned} \tag{10}$$

So the minimization of the KL divergence is equivalent to the Maximum Likelihood.

$$\begin{aligned} p_\theta(\mathbf{v}) &= \sum_{\mathbf{h}} p_\theta(\mathbf{v}, \mathbf{h}) \qquad (11) \\ &= \sum_{\mathbf{h}} \frac{1}{Z} \exp(-E(\mathbf{v}, \mathbf{h})) \\ &= \frac{1}{Z} \sum_{\mathbf{h}} \exp(\mathbf{h}^T \mathbf{W} \mathbf{v} + \mathbf{b}^T \mathbf{v} + \mathbf{c}^T \mathbf{h}) \\ &= \frac{1}{Z} \exp(\mathbf{b}^T \mathbf{v}) \sum_{\mathbf{h}} \exp(\mathbf{h}^T \mathbf{W} \mathbf{v} + \mathbf{c}^T \mathbf{h}) \\ &= \frac{1}{Z} \exp(\mathbf{b}^T \mathbf{v}) \sum_{\mathbf{h}} \exp(\sum_{j=1}^{H} (h_j W_{j,:} \mathbf{v} + c_j h_j)) \\ &= \frac{1}{Z} \exp(\mathbf{b}^T \mathbf{v}) \sum_{h_1 \in \{0,1\}} \exp(h_1 W_{1,:} \mathbf{v} + c_1 h_1) \ \ldots \sum_{h_H \in \{0,1\}} \exp(h_H W_{H,:} \mathbf{v} + c_H h_H) \\ &= \frac{1}{Z} \exp(\mathbf{b}^T \mathbf{v}) \prod_j (1 + \exp(W_{j,:} \mathbf{v} + b_j)) \\ &= \frac{1}{Z} \exp(-F(\mathbf{v})) \end{aligned}$$

where $F(\mathbf{v})$ is the *free energy* and can be calculated by

$$F(\mathbf{v}) = -\mathbf{b}^T \mathbf{v} - \sum_{j=1}^{H} \log(1 + \exp(W_{j,:} \mathbf{v} + c_j))) \tag{12}$$

Define $\theta = \{\mathbf{W}, \mathbf{b}, \mathbf{c}\}$, we will have

$$\begin{aligned}
-\frac{\partial \log p_\theta(\mathbf{v})}{\partial \theta} &= -\frac{\partial}{\partial \theta}\Big(\log \frac{\exp(-F(\mathbf{v}))}{Z}\Big) \\
&= \frac{\partial}{\partial \theta}(F(\mathbf{v}) + \log Z) \\
&= \frac{\partial F(\mathbf{v})}{\partial \theta} + \frac{1}{Z}\frac{\partial Z}{\partial \theta} \\
&= \frac{\partial F(\mathbf{v})}{\partial \theta} + \sum_{\mathbf{v}'} \frac{1}{Z}\exp(-F(\mathbf{v}'))\frac{\partial(-F(\mathbf{v}'))}{\partial \theta} \\
&= \frac{\partial F(\mathbf{v})}{\partial \theta} - \sum_{\mathbf{v}'} p_\theta(\mathbf{v}')\frac{\partial F(\mathbf{v}')}{\partial \theta} \\
&= \frac{\partial F(\mathbf{v})}{\partial \theta} - \mathbb{E}_{\mathbf{v}'}\Big[\frac{\partial F(\mathbf{v}')}{\partial \theta}\Big] \qquad (13)
\end{aligned}$$

with $\mathbf{v}' \sim p_\theta(\mathbf{v}')$. while $\frac{\partial F(\mathbf{v})}{\partial \theta}$ can be expressed as

$$\begin{aligned}
\frac{\partial F(\mathbf{v})}{\partial w_{j,k}} &= -p_\theta(h_j = 1|\mathbf{v})v_k \\
\frac{\partial F(\mathbf{v})}{\partial \mathbf{b}} &= -\mathbf{v} \\
\frac{\partial F(\mathbf{v})}{\partial \mathbf{c}} &= -p_\theta(h_j = 1|\mathbf{v})
\end{aligned}$$

The second term in Eq. (13) requires an exponential computational complexity, so we will use samples drawn from the $p_\theta(\mathbf{v}')$ to approximate the expectation based on MCMC techniques. However, we could apply block Gibbs Sampling, which is actually a special case of *Metropolis-Hastings algorithm*, to obtain samples if $p(\mathbf{v}|\mathbf{h})$ and $p(\mathbf{h}|\mathbf{v})$ are available:

1. Initialization: $\mathbf{v}^1$
2. Find: $p(\mathbf{h}^1|\mathbf{v}^1)$
3. Sample $\mathbf{h}^1 \sim p(\mathbf{h}^1|\mathbf{v}^1)$
4. Find: $p(\mathbf{v}^2|\mathbf{h}^2)$
5. Sample: $\mathbf{v}^2 \sim p(\mathbf{v}^2|\mathbf{h}^2)$

   $\dots$
6. Until equilibrium.

After sampling N samples of $\mathbf{v}'$, i.e., $\{\mathbf{v}'_k\}|_{k=1,2,\dots,N}$, we can use the sample mean to approximate the expectation:

$$\mathbb{E}_{\mathbf{v}'}\Big[\frac{\partial F(\mathbf{v}')}{\partial \theta}\Big] \approx \frac{1}{N}\sum_k \frac{\partial F(\mathbf{v}'_k)}{\partial \theta} \qquad (14)$$

However, this process, which requires us to wait until the Markov chain reaches equilibrium, is computationally inefficient. To resolve this, we use Contrastive Divergence (CD) to approximate the true value with a point estimate. CD-n indicates that the point estimate $\frac{\partial F(\mathbf{v}^n)}{\partial \theta}$ can be used to approximate $\mathbb{E}_{\mathbf{v}'}\Big[\frac{\partial F(\mathbf{v}')}{\partial \theta}\Big]$, given the initialization is the training data point.

$$\mathbb{E}_{\mathbf{v}'}\Big[\frac{\partial F(\mathbf{v}')}{\partial \theta}\Big] \approx \frac{\partial F(\mathbf{v}^n)}{\partial \theta} \qquad (15)$$

So Eq. (13) can be approximated as

$$\begin{aligned}
-\frac{\partial \log p_\theta(\mathbf{v})}{\partial \theta} &= \frac{\partial F(\mathbf{v})}{\partial \theta} - \mathbb{E}_{\mathbf{v}'}\Big[\frac{\partial F(\mathbf{v}')}{\partial \theta}\Big] \\
&\approx \frac{\partial F(\mathbf{v})}{\partial \theta} - \frac{\partial F(\mathbf{v}^n)}{\partial \theta} \qquad (16)
\end{aligned}$$

So

$$-\frac{\partial \log p_\theta(\mathbf{v})}{\partial w_{j,k}} = -p_\theta(h_j = 1|\mathbf{v})v_k + p_\theta(h_j = 1|\mathbf{v}^n)v_k^n \tag{17}$$

$$-\frac{\partial \log p_\theta(\mathbf{v})}{\partial \mathbf{b}} = -\mathbf{v} + \mathbf{v}^n \tag{18}$$

$$-\frac{\partial \log p_\theta(\mathbf{v})}{\partial \mathbf{c}} = -p_\theta(h_j = 1|\mathbf{v}) + p_\theta(h_j = 1|\mathbf{v}^n) \tag{19}$$

Now let's talk about how to calculate $p_\theta(\mathbf{h}|\mathbf{v})$ and $p(\mathbf{v}|\mathbf{h})$.

$$\begin{aligned}
p_\theta(\mathbf{h}|\mathbf{v}) &= \frac{p_\theta(\mathbf{v},\mathbf{h})}{p_\theta(\mathbf{v})} \\
&= \frac{\exp(-E(\mathbf{v},\mathbf{h}))}{\sum_{\mathbf{h}} \exp(-E(\mathbf{v},\mathbf{h}))} \\
&= \frac{\exp(\mathbf{h}^T\mathbf{W}\mathbf{v} + \mathbf{c}^T\mathbf{h})}{\sum_{\mathbf{h}} \exp(\mathbf{h}^T\mathbf{W}\mathbf{v} + \mathbf{c}^T\mathbf{h})} \\
&= \frac{\exp(\sum_j (h_j W_{j,:}\mathbf{v} + c_j h_j))}{\sum_{h_1} \dots \sum_{h_H} \exp(\sum_j (h_j W_{j,:}\mathbf{v} + c_j h_j))} \\
&= \frac{\prod_j \exp(h_j W_{j,:}\mathbf{v} + c_j h_j)}{(\sum_{h_1} \exp(h_1 W_{1,:}\mathbf{v} + c_1 h_1))(\sum_{h_2} \exp(h_2 W_{2,:}\mathbf{v} + c_2 h_2)) \dots (\sum_{h_H} \exp(h_H W_{H,:}\mathbf{v} + c_H h_H)} \\
&= \prod_j \frac{\exp(h_j W_{j,:}\mathbf{v} + c_j h_j)}{1 + \exp(W_{j,:}\mathbf{v} + c_j)} \\
&= \prod_j p_\theta(h_j|\mathbf{v})
\end{aligned} \tag{20}$$

So h_1, h_2, ..., h_H are conditionally independent given $\mathbf{v}$, with

$$p_\theta(h_j|\mathbf{v}) = \frac{\exp(h_j W_{j,:}\mathbf{v} + c_j h_j)}{1 + \exp(W_{j,:}\mathbf{v} + c_j)} \tag{21}$$

Specifically,

$$\begin{aligned}
p_\theta(h_j = 1|\mathbf{v}) &= \frac{\exp(W_{j,:}\mathbf{v} + c_j)}{1 + \exp(W_{j,:}\mathbf{v} + c_j)} \\
p_\theta(h_j = 0|\mathbf{v}) &= \frac{1}{1 + \exp(W_{j,:}\mathbf{v} + c_j)}
\end{aligned}$$

Similarly, we can derive that

$$p_\theta(v_k|\mathbf{h}) = \frac{\exp(\mathbf{h}^T W_{:,k} v_k + b_k v_k)}{1 + \exp(\mathbf{h}^T W_{:,k} + b_k)} \tag{22}$$

Specifically,

$$\begin{aligned}
p_\theta(v_k = 1|\mathbf{h}) &= \frac{\exp(\mathbf{h}^T W_{:,k} + b_k)}{1 + \exp(\mathbf{h}^T W_{:,k} + b_k)} \\
p_\theta(v_k = 0|\mathbf{h}) &= \frac{1}{1 + \exp(\mathbf{h}^T W_{:,k} + b_k)}
\end{aligned}$$

Alternatively, the derivation in some tutorials or papers does not involve the free energy function $F(\mathbf{v})$. In that case, $p_\theta(\mathbf{v})$ can be expressed as

$$\begin{aligned}
p_\theta(\mathbf{v}) &= \sum_{\mathbf{h}} p_\theta(\mathbf{v},\mathbf{h}) \\
&= \sum_{\mathbf{h}} \frac{1}{Z} \exp(-E(\mathbf{v},\mathbf{h}))
\end{aligned} \tag{23}$$

So

$$\begin{aligned}
-\frac{\partial \log p_\theta(\mathbf{v})}{\partial\theta} &= -\frac{\partial}{\partial\theta}\Big(\log\sum_{\mathbf{h}}\exp(-E(\mathbf{v},\mathbf{h}))\Big) + \frac{\partial}{\partial\theta}\Big(\log\sum_{\mathbf{v},\mathbf{h}}\exp(-E(\mathbf{v},\mathbf{h}))\Big) \\
&= \frac{1}{\sum_{\mathbf{h}}\exp(-E(\mathbf{v},\mathbf{h}))}\sum_{\mathbf{h}}\exp(-E(\mathbf{v},\mathbf{h}))\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial\theta} - \frac{1}{\sum_{\mathbf{v},\mathbf{h}}\exp(-E(\mathbf{v},\mathbf{h}))}\sum_{\mathbf{v},\mathbf{h}}\exp(-E(\mathbf{v},\mathbf{h}))\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial\theta} \\
&= \sum_{\mathbf{h}} p_\theta(\mathbf{h}|\mathbf{v})\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial\theta} - \sum_{\mathbf{v},\mathbf{h}} p_\theta(\mathbf{v},\mathbf{h})\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial\theta} \qquad (24)
\end{aligned}$$

Eq. (13) is actually the same with Eq. (24) since

$$F(\mathbf{v}) = -\log\Big(\sum_{\mathbf{h}}\exp\big(-E(\mathbf{v},\mathbf{h})\big)\Big) \qquad (25)$$

2.2 RBM Training Algorithm

```
Initialization for the $\theta = \{\mathbf{W}, \mathbf{b}, \mathbf{c}\}$;
Fix the maximum number of epochs $max\_epoch$, and mini-batch size $M$;
for $i = 1, \ldots, max\_epoch$ do
    for mini-batch $\mathbf{v}_{mb}$ in Dataset $\mathbf{D}$ do
        $\overline{\Delta\mathbf{W}}$, $\overline{\Delta\mathbf{b}}$, $\overline{\Delta\mathbf{c}} \leftarrow \mathbf{0}$;
        for each data sample $\mathbf{v}$ in $\mathbf{v}_{mb}$ do
            $\mathbf{v}^{(0)} \leftarrow \mathbf{v}$;
            for $t = 0, \ldots, n-1$ do
                Sample $\mathbf{h}^{(t)} \sim p_\theta(\mathbf{h}|\mathbf{v}^{(t)})$ ;
                Sample $\mathbf{v}^{(t+1)} \sim p_\theta(\mathbf{v}|\mathbf{h}^{(t)})$ ;
            end
            Calculate the gradient $\Delta\mathbf{W}$, $\Delta\mathbf{b}$, and $\Delta\mathbf{c}$ according to Eq. (17) (18) (19);
            $\overline{\Delta\mathbf{W}}$, $\overline{\Delta\mathbf{b}}$, $\overline{\Delta\mathbf{c}} \leftarrow \overline{\Delta\mathbf{W}} + \Delta\mathbf{W}$, $\overline{\Delta\mathbf{b}} + \Delta\mathbf{b}$, $\overline{\Delta\mathbf{c}} + \Delta\mathbf{c}$;
        end
        $\mathbf{W}, \mathbf{b}, \mathbf{c} \leftarrow \mathbf{W} - \frac{1}{M}\overline{\Delta\mathbf{W}}$, $\mathbf{b} - \frac{1}{M}\overline{\Delta\mathbf{b}}$, $\mathbf{c} - \frac{1}{M}\overline{\Delta\mathbf{c}}$;
    end
end
```

Algorithm 1: RBM Training Algorithm

2.3 Implementation in Python

TO BE COMPLETED