Boltzmann Machines

Chunpai Wang

October 14, 2018

1 The Boltzmann Machine

A Boltzmann machine is a Markov random field having a particular structure.

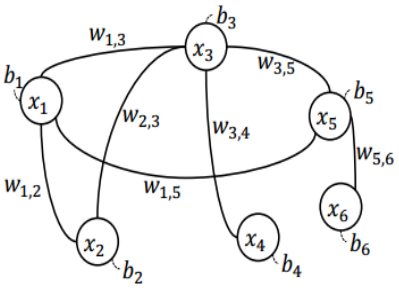

Figure 1: The Boltzmann Machine

A Boltzmann machine contains N units or particles, each of which has state in {0,1} and associated with a random variable X_i . We denotes the parameters in Boltzmann machine as $\theta = (b_1, \cdots, b_N, w_{1,2}, \cdots, w_{N-1,N})$, where b_i is bias for i^{th} unit and $w_{i,j}$is weight between unit i and unit j, and specifically $(i, j) \in [1, N-1]\times[i+1, N]$. The energy of the Boltzmann machine is defined as

$$E_\theta(\mathbf{x}) = -\sum_{i=1}^{N} b_i x_i - \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} w_{i,j} x_i x_j = -\mathbf{b}^\top \mathbf{x} - \mathbf{x}^\top \mathbf{W}\mathbf{x} \tag{1}$$

Of course, we can still use Boltzmann's Law to convert the energy into probability:

$$P_\theta(\mathbf{x}) = \frac{1}{Z(T)} \exp\{-E_\theta(\mathbf{x})/T\} = \frac{\exp(-E_\theta(\mathbf{x}))}{\sum_{\tilde{\mathbf{x}}\in S} \exp(-E_\theta(\tilde{\mathbf{x}})} \tag{2}$$

where we set $T = 1$, S denotes the space of all possible configurations of states, and the denominator is the well-known partition function Z. Since we are capable to convert the energy to probability, a Boltzmann machine can be used to model the probability distribution of a target pattern, denoted by $P_{data}(\tilde{\mathbf{x}})$.

2 Different Boltzmann Machines

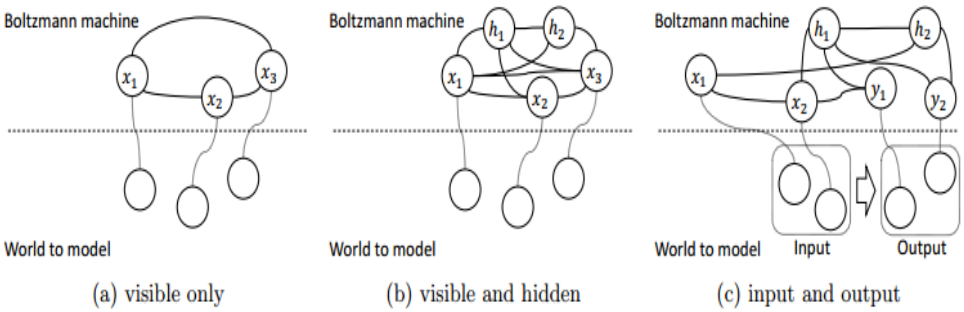

Figure 2: The Boltzmann Machines with Hidden Units, and Conditional Boltzmann Machine. (a) and (b) are generative models, and (3) is discriminative model.

We can build different Boltzmann machine to model various probability distributions, for example the distribution with hidden variable and the conditional distribution, which corresponds to the (b) and (c) in the figure above. The circle above the dash represent the $P_\theta(\cdot)$, and the ones under the dash line represent $P_{data}(\cdot)$. We will learn how to model those distribution with Boltzmann machine in following sections, and explain why the structure of Boltzmann machines is useful.

3 Learning A Generative Model

Now we would like to use our Boltzmann machine to learn a $P_\theta(\cdot)$ that best approximates a given $P_{data}(\cdot)$. The classical method is to minimize the KL-divergence from $P_\theta(\cdot)$ to $P_{data}(\cdot)$

$$\begin{aligned} KL(P_{data}\|P_\theta) &= \sum_{\tilde{\mathbf{x}}\in S} P_{data}(\tilde{\mathbf{x}}) \log \frac{P_{data}(\tilde{\mathbf{x}})}{P_\theta(\tilde{\mathbf{x}})} \\ &= \sum_{\tilde{\mathbf{x}}\in S} P_{data}(\tilde{\mathbf{x}}) \log P_{data}(\tilde{\mathbf{x}}) - \underbrace{\sum_{\tilde{\mathbf{x}}\in S} P_{data}(\tilde{\mathbf{x}}) \log P_\theta(\tilde{\mathbf{x}})}_{f(\theta)} \end{aligned}$$

Note that, minimizing the $KL(P_{data}\|P_\theta)$ is same as maximizing the log likelihood of sampled data. In other words, we sample m data from real distribution $P_{data}(\cdot)$, and we would like to find the parameter θ such that maximize the $P_\theta(\cdot)$. In order to find the optimal θ , we only need to maximize the second term $f(\theta)$ by taking the gradient of $f(\theta)$ with respect to θ:

$$\nabla f(\theta) = \sum_{\tilde{\mathbf{x}}} P_{data}(\tilde{\mathbf{x}}) \nabla \log P_\theta(\tilde{\mathbf{x}}) \tag{3}$$

Is the function $f(\theta)$ convex ?

3.1 Gradient-Based Method

Here, we only consider the case that all of the units are visible. Once we find the gradient, we can use the **gradient-based method** to find the maximal point:

$$\theta = \theta + \eta \nabla f(\theta) \tag{4}$$

We can write down the closed form of $\nabla \log P_\theta(\tilde{\mathbf{x}})$,

$$\begin{aligned} \nabla \log P_\theta(\mathbf{x}) &= \nabla \log \frac{\exp(-E_\theta(\mathbf{x}))}{\sum_{\tilde{\mathbf{x}}} \exp(-E_\theta(\hat{\mathbf{x}}))} && (5)\\ &= -\nabla E_\theta(\mathbf{x}) - \nabla \log \sum_{\hat{\mathbf{x}}} \exp(-E_\theta(\hat{\mathbf{x}})) && (6)\\ &= -\nabla E_\theta(\mathbf{x}) - \frac{\sum_{\hat{\mathbf{x}}} \exp(-E_\theta(\hat{\mathbf{x}})) \nabla E_\theta(\hat{\mathbf{x}})}{\sum_{\tilde{\mathbf{x}}} \exp(-E_\theta(\tilde{\mathbf{x}}))} && (7)\\ &= -\nabla E_\theta(\mathbf{x}) + \sum_{\hat{\mathbf{x}}} \frac{\exp(-E_\theta(\hat{\mathbf{x}}))}{\sum_{\tilde{\mathbf{x}}} \exp(-E_\theta(\tilde{\mathbf{x}}))} \nabla E_\theta(\hat{\mathbf{x}}) && (8)\\ &= -\nabla E_\theta(\mathbf{x}) + \sum_{\hat{\mathbf{x}}} P_\theta(\hat{\mathbf{x}}) \nabla E_\theta(\hat{\mathbf{x}}) && (9) \end{aligned}$$

and therefore, we can derive the $\nabla f(\theta)$

$$\begin{aligned} \nabla f(\theta) &= \sum_{\tilde{\mathbf{x}}} P_{data}(\tilde{\mathbf{x}}) \nabla \log P_\theta(\tilde{\mathbf{x}}) && (10)\\ &= \sum_{\tilde{\mathbf{x}}} P_{data}(\tilde{\mathbf{x}}) \left\{ -\nabla E_\theta(\mathbf{x}) + \sum_{\hat{\mathbf{x}}} P_\theta(\hat{\mathbf{x}}) \nabla E_\theta(\hat{\mathbf{x}}) \right\} && (11)\\ &= -\sum_{\tilde{\mathbf{x}}} P_{data}(\tilde{\mathbf{x}}) \nabla E_\theta(\tilde{\mathbf{x}}) + \sum_{\tilde{\mathbf{x}}} P_{data}(\tilde{\mathbf{x}}) \sum_{\hat{\mathbf{x}}} P_\theta(\hat{\mathbf{x}}) \nabla E_\theta(\hat{\mathbf{x}}) && (12)\\ &= -\sum_{\tilde{\mathbf{x}}} P_{data}(\tilde{\mathbf{x}}) \nabla E_\theta(\tilde{\mathbf{x}}) + 1.0 * \sum_{\hat{\mathbf{x}}} P_\theta(\hat{\mathbf{x}}) \nabla E_\theta(\hat{\mathbf{x}}) && (13)\\ &= -\sum_{\tilde{\mathbf{x}}} (P_{data}(\tilde{\mathbf{x}}) - P_\theta(\tilde{\mathbf{x}})) \nabla E_\theta(\tilde{\mathbf{x}}) && (14)\\ &= \underbrace{\mathbb{E}_\theta[\nabla E_\theta(\mathbf{x})]}_{\substack{\text{expected gradient of energy} \\ \text{w.r.t to data distribution}}} - \underbrace{\mathbb{E}_{data}[\nabla E_\theta(\mathbf{x})]}_{\substack{\text{expected gradient of energy} \\ \text{w.r.t model distribution} \\ \text{with the current value of } \theta}} && (15) \end{aligned}$$

3.2 Interpretation

Now, let's interpret the formula (13) and (14) in term of the gradient-ascent update. We would like to maximize the $f(\theta)$, which is same as maximizing the log-likelihood. For formula (14), for each configuration $\tilde{\mathbf{x}}$, we compare $P_\theta(\tilde{\mathbf{x}})$ and $P_{data}(\tilde{\mathbf{x}})$:

- If $P_\theta(\tilde{\mathbf{x}}) > P_{data}(\tilde{\mathbf{x}})$, that means the configuration $\tilde{\mathbf{x}}$ is likely generated by our model P_θ rather than sampled from data distribution, therefore we update θ along the gradient of energy w.r.t θ that it increases the energy $E_\theta(\tilde{\mathbf{x}})$ (note that for $Tilde\mathbf{x}$, and large energy means incompatibility of states) so that the $\tilde{\mathbf{x}}$ becomes less likely to be generated with P_θ.
- If $P_\theta(\tilde{\mathbf{x}})$ is smaller than $P_{data}(\tilde{\mathbf{x}})$, we update θ in a way that $E_\theta(\tilde{\mathbf{x}})$ decreases.

For formula (13), the learning rule can be considered as decreasing the energy of all "positive" or "real" samples that are generated according to a target data distribution P_{data} and increasing the energy of "negative" or "fake" samples that are generated according to the current model. With this learning rule, this model can be trained as a good discriminator that classify if the example is generated by real data distribution or by our model, since our model will give low energy to configuration $\tilde{\mathbf{x}}$ from data distribution, and high energy to configuration $\tilde{\mathbf{x}}$ generated by our model.
Eventually, the P_θ will become similar to P_{data}(close but not same, since it only gets the local optima).

3.3 Stochastic Gradient-Based Method

We can rewrite the $\nabla \log P_\theta(\mathbf{x})$ as

$$\nabla \log P_\theta(\mathbf{x}) = -\nabla E_\theta(\mathbf{x}) + \mathbb{E}_\theta[\nabla E_\theta(\mathbf{x})] \tag{16}$$

4 Hidden Variables

If we introduce hidden variables in our model, likewise the hidden variable in EM algorithm, we introduce the **free energy** to replace the energy function for Boltzmann machine.

4.1 Gradient-Based

4.2 Stochastic Gradient-Based

5 Learning a Discriminative Model

6 Evaluating Expectation w.r.t a Model Distribution

References

[1] Nowozin, Sebastian, and Christoph H. Lampert. "Structured learning and prediction in computer vision." Foundations and Trends® in Computer Graphics and Vision 6.3–4 (2011): 185-365.

[2] LeCun, Yann, et al. "A tutorial on energy-based learning." Predicting structured data 1.0 (2006).

[3] Osogami, Takayuki. "Boltzmann machines and energy-based models." arXiv preprint arXiv:1708.06008 (2017).