CSE6243: Advanced Machine Learning **Fall 2023**

Lecture 11: auto-regressive model

Lecturer: Bo Dai *Scribes: Weihan Li, Shiqin Zeng*

11.1 Recap

The learning methods for *energy-based model*:

- Contrastive Divergence.
- Score Matching.
- Noise Contrastive Estimation.

11.2 Noise Contrastive Estimation

Fact 1. Assume $p(y=1)=p(y=0)=\frac{1}{2}$, $p(x|y=1)=p_\theta(x)$, and $p(x|y=0)=p_n(x)$. Where $p_\theta(x)$ is the target distribution, and $p_n(x)$ presents a defined noise distribution. Using Bayes' rule, we have:

$$\begin{aligned} p(y=1|x) &= \frac{p(x,y=1)}{p(x)} \\ &= \frac{p(y=1)\times p(x|y=1)}{p(x|y=1)\times p(y=1)+p(x|y=0)\times p(y=0)} \\ &= \frac{\frac{1}{2}p_\theta(x)}{\frac{1}{2}p_\theta(x)+\frac{1}{2}p_n(x)} \\ &= \frac{p_\theta(x)}{p_\theta(x)+p_n(x)}, \end{aligned} \tag{11.1}$$

where we can rewrite $p_\theta(x)$ to be:

$$p_\theta(x) = \frac{p(y=1|x)p_n(x)}{1-p(y=1|x)}. \tag{11.2}$$

This means we don't need to learn the target distribution $p_\theta(x)$ directly, but we can obtain it by $p_n(x)$ and $p(y=1|x)$ instead, where $p_n(x)$ is a known distribution.

Fact 2. $p(y=1|x)$ can be computed by following equation:

$$p(y=1|x) = \arg\min_D \mathbf{E}_{p_+}[\log f(x)] + \mathbf{E}_{p_n}[\log(1-f(x))], \tag{11.3}$$

where $p_+(x)$ is a distribution sampled from the dataset D. Then, we take derivatives of (11.3):

$$\begin{aligned} p_+(x)\frac{1}{f(x)} - p_n(x)\frac{1}{1-f(x)} &= 0 \\ (1-f(x))p_+(x) - f(x)p_n(x) &= 0 \\ \frac{p_+(x)}{p_+(x)+p_n(x)} &= f(x), \end{aligned} \tag{11.4}$$

which means if we set $p_+(x)$ to be the distribution obtained from data D, then $f(x)$ is the solution to (11.3). In summary, the steps for Noise Contrastive Estimation ares:

- Pre-define $p_n(x)$ (usually Gaussian distribution) and get $p_+(x)$ from data $D : \{x_i, y_i\}_{i=1}^n$.
- Compute solution $p(y = 1|x) = f(x)$ for (11.3).
- Compute $p_\theta(x)$ using Eq 11.2.

Compared with Contrastive Divergence and Score Matching, the advantages of Noise Contrastive Estimation is that it doesn't use samples to approximate the intractable terms (e.g., integrals). This idea is also similar to amortize learning, which uses neural network to directly learn something that could bring errors by traditional methods.

11.3 Auto-Regressive Model

The definition of *auto-regressive model* is:

$$p(x) = \prod_{i=1}^{n} p(x_i|x_{<i}), \tag{11.5}$$

where $x = [x^1, \ldots, x^d]$ and $x_{<i} = [x^1, \ldots, x^{i-1}]$.

We first Consider a simple case where using Bernoulli distribution to model (11.5):

$$p(x_i|x_{<i}) = \text{Bernoulli}(f(x_{<i})), \tag{11.6}$$

where $x^i \in \{0, 1\}$. Then we have:

$$x^i = \text{sigmoid}(f(x_<))^{x^i}(1 - \text{sigmoid}(f(x_{<i})))^{1-x^i}. \tag{11.7}$$

The model parameters θ can be estimated by maximum likelihood estimation:

$$\theta = \max \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{d} \log(p(x_n^i|x_n^{<i})), \tag{11.8}$$

where N is the number of data samples and d is the number of dimension for each sample.

The next question is how can we connect auto-regressive model to energy-based model? Let's consider restricted Boltzmann machine model (rRBM). If we denote hidden variables to be h and observed variables to be x, then we have:

$$p(x, h) \propto \exp(g(x, h)), \tag{11.9}$$

where $g(x,h) = x^T wh + b^T x + c^T h$, and w, b, c are the parameters of rRBM. We can derive the posterior $p(h|x)$ as:

$$\begin{aligned} p(h|x) &\propto \exp(h^T(c + w^T x)) \\ &= \prod_{j=1}^{D} p(h_j|x), \end{aligned} \tag{11.10}$$

and $p(h_j = 1|x)$ is:

$$\begin{aligned} p(h_j = 1|x) &= \frac{\exp(h_j(c^T + w^T x)_j)}{\sum_{h\in\{0,1\}} \exp(h_j(c^T + w^T x)_j)} \\ &= \frac{\exp((c^T + w^T x)_j)}{1 + \exp((c^T + w^T x)_j)} \\ &= \text{sigmoid}((c^T + w^T x)_j). \end{aligned} \tag{11.11}$$

Similarly, we can derive the likelihood $p(x|h)$ as:

$$p(x|h) = \prod_{i=1}^{d} p(x_i|h), \tag{11.12}$$

where each $p(x_i|h)$ is

$$p(x_i|h) = \text{sigmoid}((wh + b)_i). \tag{11.13}$$

In summary, we can get the mean of hidden variables h by

$$\hat{h} = \text{sigmoid}(c^T + w^T x), \tag{11.14}$$

and the likelihood $p(x|h)$ by

$$p(x|h) = \text{sigmoid}(wh + b). \tag{11.15}$$

These two equations can be used to generate data from RBM if given $x^0 = [0, \ldots, 0]$:

- Compute $\hat{h_1}$ using (11.14).
- Compute x^1 using $\hat{h_1}$ and (11.15).
- Repeat the above two steps.

Not surprisingly, from the perspective of generative process, rRBM can be considered as an auto-regressive model by applying the Gibbs sampling to sample hidden and observed variables.

Besides, we can apply the Gibbs sampling to a neural network, for example, recurrent neural network (RNN), which also equals an auto-regressive model:

$$\begin{aligned} z_i &= z_{i-1} + w^T x_i \\ h_i &= \text{sigmoid}(z_i) \\ p(x_{i+1}|h_i) &= \text{sigmoid}(wh_i + b_i), \end{aligned} \tag{11.16}$$