CSE6243: Advanced Machine Learning **Fall 2023**

Lecture 18: Representation Learning from EBM view

Lecturer: Bo Dai *Scribes: Yuheng Li, Junha Lee*

Note: *LaTeX template courtesy of UC Berkeley EECS Department.*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

1.1 Recap on Energy-Based Model (EBM)

Energy-Based Models (EBMs) are a type of probabilistic model where the probability distribution of data is defined in terms of an energy function. They are often used in machine learning to model complex dependencies in structured data. The energy-based function can be defined as follows:

$$p(x) = \frac{\exp(f(x))}{Z(f)} \tag{1.1}$$

$$Z(f) = \int \exp(f(x))\, dx \tag{1.2}$$

- $p(x)$ represents the probability density of a data point x in the model.
- $f(x)$ is the energy function, often represented by a neural network or other parametric model. The function $f(x)$ assigns lower energy values to more probable (or desired) outcomes and higher energy values to less probable ones.
- $Z(f)$, known as the partition function, is a normalization constant. It ensures that $p(x)$ is a valid probability distribution by integrating over all possible states x.

In practice, calculating the partition function $Z(f)$ is often computationally intractable, as it involves integrating over all possible configurations of x. This is one of the main challenges in using EBMs. We have seen techniques like Contrastive Divergence and Score Matching to train EBMs.

Pros of using EBM: good model to capture structured data.
Cons of using EBM: It is difficult to sample.

1.2 New Content

1. Noise Contrastive Estimation (NCE)

Noise Contrastive Estimation (NCE) is a technique commonly used to train models where calculating the exact likelihood is challenging. Instead of maximizing the likelihood directly, NCE reformulates the problem as a classification task that distinguishes between observed data samples (from the true distribution) and samples from a noise distribution. By learning to discriminate between ”real” and ”noise” samples, the model can indirectly approximate the probability of the data without needing to compute complex normalization terms, such as the partition function in EBMs.

NCE assumes that we have two types of data: 1). True data samples: samples from the actual data distribution $p_{\text{data}}(x)$. and 2). Noise samples: samples generated from a known noise distribution $p_{\text{noise}}(x)$, which serves as a negative reference.

Binary NCE

In a binary NCE, we can use logistic regression as the classification objective. The model is trained to classify:
- Positive examples ($y = 1$) as samples from the data distribution $p_{\text{data}}(x)$.
- Negative examples ($y = 0$) as samples from the noise distribution $p_{\text{noise}}(x)$.

Formally, we define the objective function as follows:

- Let $x \sim p_{\text{data}}(x)$ represent samples drawn from the data distribution, and $y = 0$ denote a noise or negative label.
- The objective is to maximize the expected log-likelihood for our discriminator $D_\theta(x)$ for correctly classifying positive and negative samples. This can be expressed as:

$$\max_\theta \mathbb{E}_{p_{\text{data}}} [\log D_\theta(x)] + \mathbb{E}_{p_{\text{noise}}} [\log(1 - D_\theta(x))] \tag{1.3}$$

$$\frac{\partial \mathcal{L}(\theta)}{\partial D_\theta(x)} \to \mathbb{E}_{p_{\text{data}}} \left[\frac{1}{D_\theta(x)} \right] + \mathbb{E}_{p_{\text{noise}}} \left[\frac{-1}{1 - D_\theta(x)} \right] = 0 \tag{1.4}$$

Solving this equation provides insight into the optimal form of $D_\theta(x)$. Specifically, we find that the optimal discriminator $D^*_\theta(x)$ is the ratio of the probability densities for data and noise distributions:

$$\frac{p_{\text{data}}(x)}{p_{\text{noise}}(x)} = \frac{D^*_\theta(x)}{1 - D^*_\theta(x)} \Rightarrow D^*_\theta(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_{\text{noise}}(x)} \tag{1.5}$$

Assuming parameterization with $p(x) = p_{\text{data}} + p_{\text{noise}}$, we get:

$$D^*_\theta(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_{\text{noise}}(x)} \tag{1.6}$$

This expression represents the probability that a sample x belongs to the data distribution as opposed to the noise distribution, effectively separating the data from noise by comparing their relative probabilities. Using this optimal discriminator simplifies the problem by allowing us to focus on the data distribution without requiring a direct computation of the partition function.

Now we can proceed to define the energy-based probability density function p(x) in terms of an energy function:

$$p(x) = \frac{\exp(f_\theta(x))}{Z(\theta)} = \exp\left(f_\theta(x) - \log(Z(\theta))\right) = \exp(g_\theta(x)) \tag{1.7}$$

where:
- $Z(\theta)$ is the partition function, given by $Z(\theta) = \int \exp(f_\theta(x))\, dx$,
- $g_\theta(x)$ represents the energy function, $f_\theta(x) - \log(Z(\theta))$.

Then, we define the discriminator $D_\theta(x)$ in NCE as:

$$D_\theta(x) = \frac{\exp(g_\theta(x))}{\exp(g_\theta(x)) + p_{\text{noise}}(x)} \tag{1.8}$$

To use NCE, we need to sample from $p_{\text{noise}}(x)$ and ensure it has some density.

Ranking-Based NCE: Extending NCE to Multi-Class

To extend NCE to a multi-class scenario, we consider $x_k \sim p_{\text{data}}(x)$, where x_k belongs to one of k classes, and $\{x_j\}_{j=1}^{k-1} \sim p_{\text{noise}}(x)$.

The objective becomes:

$$q(k|(\{x_i\}_{i=1}^k)) = \frac{p_{\text{data}}(x_i)\prod_{i=1}^{k-1} p_{\text{noise}}(x_j)}{\sum_{j=1}^k p_{\text{data}}(x_j)\prod_{i\neq j}^k p_{\text{noise}}(x_i)} \tag{1.9}$$

divide by $\prod_{i=1}^k p_{\text{noise}}(x_i)$:

$$q = \frac{p_{\text{data}}(x_k)}{p_{\text{noise}}(x_k)} \Big/ \sum_{j=1}^k \frac{p_{\text{data}}(x_j)}{p_{\text{noise}}(x_j)} \tag{1.10}$$

Let $g_\theta(x) = \frac{p_{\text{data}}(x)}{p_{\text{noise}}(x)}$. Then we can parameterize it as:

$$p_\theta\left(k|\{x_i\}_{i=1}^k\right) = \frac{\exp(g_\theta(x_i))}{\sum_{j=1}^k \exp(g_\theta(x_j))} \tag{1.11}$$

To minimize the KL divergence, we aim to solve:

$$\min \text{KL}\left(q\left(k \mid \{x_i\}_{i=1}^k\right) \parallel p_\theta\left(k \mid \{x_i\}_{i=1}^k\right)\right) \tag{1.12}$$

Expanding this, we get:

$$= -\sum_{i=1}^n \left(g_\theta(x_i) - \log\sum_{j=1}^k \exp(g_\theta(x_j))\right) \tag{1.13}$$

where $x_i \sim p_{\text{data}}$ and $x_j \sim p_{\text{noise}}$ for $j \neq i$.

We also have the likelihood ratio for each x' relative to the anchor x:

$$p(x' \mid x) = p_n(x')\exp(\phi(x')^T\phi(x)) \tag{1.14}$$

which implies:

$$\frac{p(x' \mid x)}{p_n(x')} = \exp(\phi(x')^T \phi(x)) \tag{1.15}$$

Finally, we minimize ϕ as follows:

$$\min_{\phi} - \sum_{i=1}^{n} \left[\phi(x'_k)^T \phi(x_i) - \log \sum_{j=1}^{k-1} \exp(\phi(x_j)^T \phi(x_i)) \right] \tag{1.16}$$

This completes the derivation of SimCLR.

2. Extending to CLIP

We can view CLIP as a multimodal extension of SimCLR.

Let x represent text and y represent an image. We have the following conditional probabilities:

$$p(y|x) = p_n(y) \exp(\phi(x)^T \mu(y)) \tag{1.17}$$
$$p(x|y) = p_n(x) \exp(\phi(x)^T \mu(y)) \tag{1.18}$$

To extend the NCE loss to image-text pairs, we minimize the following objective:

$$\min_{\phi,\mu} - \sum_{i=1}^{n} \left(\phi(x_i)^T \mu(y_i) - \log \sum_{y_j \sim P_n(y)} \exp(\phi(x_i)^T \mu(y_j)) \right) \tag{1.19}$$

In practice, CLIP uses a symmetric loss over both text-to-image and image-to-text directions, resulting in the final objective:

$$\mathcal{L}_{\text{CLIP}} = -\frac{1}{2} \sum_{i=1}^{n} \left(\log \frac{\exp(\phi(x_i)^T \mu(y_i))}{\sum_{y_j} \exp(\phi(x_i)^T \mu(y_j))} + \log \frac{\exp(\phi(x_i)^T \mu(y_i))}{\sum_{x_j} \exp(\phi(x_j)^T \mu(y_i))} \right) \tag{1.20}$$