CSE6243: Advanced Machine Learning **Fall 2024**

Lecture 12: VAE and Diffusion I

Lecturer: Bo Dai *Scribes: Yipu Chen, Mehrdad Moradi*

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

12.1 Recap

Energy Based Models Recall energy based models have the form:

$$P_\theta(x) = \frac{\exp(f_\theta(x))}{Z(\theta)}, Z_\theta = \int \exp(f_\theta(x))dx$$

For sampling from this distribution, we could use:

$$x_{t+1} \leftarrow x_t + \eta_t \nabla_{x_t} \log P_\theta(x_t) + \sqrt{2\eta_t}\epsilon$$

Note that for generation, just knowing the log probability is enough, this leads to the following two approaches:

1. Parameterize the log probability directly, and learn it. This leads to score matching.
2. Perturb the original distribution and estimate the gradient. This leads to diffusion models.

Note: although both methods throw away the energy and estimate the gradient instead, the original form of energy based models is still useful and practical. Later, the professor will show how to connect EBM with representation learning.

12.2 New Content

Today's topic is to talk about latent variational auto-encoders and how it connects with diffusion models.

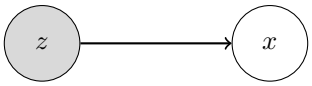
12.2.1 Latent Variable Model

An example latent variable model has the following form: $x_0 \sim P_0(x)$ is the distribution we are interested in. And say we are given the following transitions $x_t|x_{t-1} \sim \mathcal{N}(x_{t+1} + \eta_t S_\theta(x_{t-1}), \sigma^2 I) =: P(x_t|x_{t-1})$. There's a trajectory and we can multiply the Markov chain to obtain $P(\{x_i\}_{i=0}^T) = \prod_{i=1}^T P(x_i|x_{i-1})P(x_0)$. Note we can only observe x_T, and all the intermediate $t < T$ we cannot observe. Thus we can only match x_T. Note we have

$$P_\theta(x_T) = \int P(x_T, (x_0, \ldots, x_{T-1}) d\{x\}_0^{T-1}$$

And using some distance measure $D(\hat{P}(x_T)||P(x_T))$, matching this allows us to learn the $S_\theta(\cdot)$.

Next, let's look at some examples of latent variable models. Each of the following has a latent variable z and we can observe x which depends on z only. An graphical illustration is the following:



12.2.2 Example I: Gaussian Mixture Model

In Gaussian Mixture Model (GMM), the latent variable z comes from a categorical distribution: $z \sim \text{Cat}(\{p_i\}_i^k)$. If we use one hot representation $z \in 0, 1^k$. The conditional distribution $x|z \sim \mathcal{N}(c_z, \sigma^2 I)$. $C \in \mathbb{R}^{p \times k}$ is a matrix recording the centers of the Gaussian, and $c_z \in \mathbb{R}^{p \times 1}$ is a specific center. To fit a GMM, we usually use the expectation maximization method (EM).

12.2.3 Example II: Variational Autoencoder

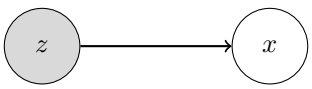
In variational autoencoder (VAE), specifically Gaussian VAE, the latent variable $z \sim \mathcal{N}(0, \sigma^2 I)$. The conditional distribution $p(x|z) \propto \exp(-\frac{||f_\theta(z)-x||^2}{2\sigma^2})$.

12.2.4 Example III: Vector Quantized Variational Autoencoder

In vector quantized variational autoencoder (VQ-VAE), the latent variable $z \sim \text{Multi}(\{p_i\}_{i=1}^k)$. The conditional distribution $p(x|z) \propto \exp(-\frac{||f_\theta(z)-x||^2}{2\sigma^2})$. VQ-VAEs are often used as image tokenizers.

12.2.5 Training VAEs

Next we will derive a method to train VAEs. Recall that in a VAE we have observed variable x that depends on the latent variable z.



$$p(x) = \int p(x|z)p(z)dx$$

We only have observed data x: $D = \{x_i\}_{i=1}^n$. Let's perform MLE:

$$KL(\hat{p}(x)||p(x)) \propto \frac{1}{n}\sum_{i=1}^{n} \log \int p_\theta(x_i|z)p(z)dz =: L(\theta)$$

In the first try we can directly calculate the gradient with respect to θ. The following derivation focuses on just one data-sample. The same idea applies when we have a dataset.

$$\begin{aligned}
\nabla_\theta L(\theta) &= \frac{\int \nabla_\theta p_\theta(x_i|z)p(z)dz}{\int p_\theta(x_i|z)p(z)dz} && \text{(integration is linear so we can pull in the gradient operator)} \\
&= \frac{\int p_\theta(x_i|z)\nabla_\theta \log p_\theta(x_i|z)p(z)dz}{\int p_\theta(x_i|z)p(z)dz} && \text{(log trick)} \\
&= \frac{E_{p(z)}[p(x|z)\nabla_\theta \log p_\theta(x|z)]}{E_{p(z)}[p_\theta(x|z)]}
\end{aligned}$$

This estimation is not very good because of the high variance induced by taking the division between MC estimations (the variance multiplies). Even though the two estimates are unbiased estimators, the division result is not.

Another Approach

The maximum likelihood estimation (MLE) can be expressed as:

$$\text{MLE:} \quad \log \int p_\theta(x|z)p(z)dz = \log \int \frac{q_\lambda(z|x)p_\theta(x|z)p(z)}{q_\lambda(z|x)}dz$$

Using Jensen's Inequality:

$$= \log E_{q_\lambda(z|x)}\left[\frac{p_\theta(x|z)p(z)}{q_\lambda(z|x)}\right] \geq E_{q_\lambda(z|x)}\left[\log \frac{p_\theta(x|z)p(z)}{q_\lambda(z|x)}\right]$$

To make the values match as closely as possible, we can try to choose $q_\lambda(z|x)$ that maximizes the above expression:

$$\max_{q_\lambda(z|x)} \int q_\lambda(z|x)\left[\log p_\theta(x|z) + \log \frac{p(z)}{q_\lambda(z|x)}\right]$$

Thus the original MLE objective after the introduction of the $q_\lambda(z|x)$ becomes:

$$\max_\theta \log \int p_\theta(x|z)p(z)dz = \max_\theta \max_{q_\lambda(z|x)} E_{q_\lambda(z|x)}\left[\log p_\theta(x|z) + \log \frac{p(z)}{q_\lambda(z|x)}\right]$$

After fixing a $q_\lambda(z|x)$ we can then maximize with respect to θ. With $q(z|x)$ determined the gradient w.r.t. θ becomes:

$$\nabla_\theta \Omega(\theta) = E_{q_\lambda(z|x)}\left[\nabla_\theta \log p_\theta(x|z)\right]$$

EM Algorithm

1. Update λ to update $q_\lambda(z|x)$ (E-step). 2. Update θ to update θ (M-step).

How to Get $\nabla_\lambda \Omega(\lambda)$

If $q_\lambda(z|x)$ is Gaussian, we have a straightforward solution. In general, we need gradient updates to obtain q_λ. The equation becomes:

$$\Omega(\lambda) = E_{q_\lambda(z|x)}\left[\log p_\theta(x|z)\right] + E_{q_\lambda(z|x)}\left[\frac{p(z)}{q_\lambda(z|x)}\right]$$

The gradient ∇_λ is given by:

$$\nabla_\lambda \Omega(\lambda) = \nabla_\lambda E_{q_\lambda(z|x)} \left[\log p_\theta(x|z)\right] + \nabla_\lambda E_{q_\lambda(z|x)} \left[\log \frac{p(z)}{q_\lambda(z|x)}\right]$$

First, we focus on the first component using the log trick:

$$\nabla_\lambda E_{p_\theta(z|x)} \left[\log p_\theta(x|z)\right] = \nabla_\lambda \int \log p_\theta(x|z) q_\lambda(z|x) =$$

$$\int \log p_\theta(x|z) \nabla_\lambda \log q_\lambda(z|x) q_\lambda(z|x) = E_{q_\lambda(z|x)} \left[\nabla_\lambda \log q_\lambda(z|x) \log p_\theta(x|z)\right]$$

Note again the variance of this term is large because it is the product of two terms. Next we focus on the second term:

$$\begin{aligned}
&\nabla_\lambda \int q_\lambda(z|x) \log \frac{p(z)}{q_\lambda(z|x)} dz \\
=&\int \nabla_\lambda (q_\lambda(z|x)) \log \frac{p(z)}{q_\lambda(z|x)} dz + \int q_\lambda(z|x) \nabla_\lambda (\frac{p(z)}{q_\lambda(z|x)}) dz && \text{(product rule)} \\
=&\int q_\lambda(z|x) \nabla_\lambda (\log q_\lambda(z|x)) \log \frac{p(z)}{q_\lambda(z|x)} dz + \int q_\lambda(z|x) \cdot -\nabla_\lambda (q_\lambda(z|x)) dz \\
& && \text{(log trick on the first term, property of log in the second term)} \\
=&E_{q_\lambda(z|x)} \left[\nabla_\lambda (\log q_\lambda(z|x)) \log \frac{p(z)}{q_\lambda(z|x)}\right] - \int \nabla_\lambda q_\lambda(z|x) dz && \text{(log trick on the second term)} \\
=&E_{q_\lambda(z|x)} \left[\nabla_\lambda (\log q_\lambda(z|x)) \log \frac{p(z)}{q_\lambda(z|x)}\right] - \nabla_\lambda \int q_\lambda(z|x) dz && \text{(move gradient op outside of integration)} \\
=&E_{q_\lambda(z|x)} \left[\nabla_\lambda (\log q_\lambda(z|x)) \log \frac{p(z)}{q_\lambda(z|x)}\right] - 0 \\
& && \text{(gradient of constant–marginal distribution integrates to 1–is 0)}
\end{aligned}$$

Combining the two parts together we obtain:

$$\begin{aligned}
\nabla_\lambda \Omega(\lambda) &= E_{q_\lambda(z|x)} \left[\nabla_\lambda \log q_\lambda(z|x) \log p_\theta(x|z)\right] + E_{q_\lambda(z|x)} \left[\nabla_\lambda (\log q_\lambda(z|x)) \log \frac{p(z)}{q_\lambda(z|x)}\right] \\
&= E_{q_\lambda(z|x)} \left[\nabla_\lambda (\log q_\lambda(z|x)) \log \frac{p_\theta(x|z)p(z)}{q_\lambda(z|x)} dz\right]
\end{aligned}$$

This generic derivation works on any q, using this gradient we can update λ and maximize the objective. However the log of the ratio is unstable and prevents good estimate of the gradient.

Another Approach

If $q_\lambda(z|x)$ is Gaussian:

$$q_\lambda(z|x) = \mathcal{N}(h_\lambda(x), \sigma^2 I), \quad z = h_\lambda(x) + \epsilon \cdot \sigma, \quad \epsilon \sim \mathcal{N}(0, I)$$

We can apply the reparameterization trick:

$$\Omega(\lambda) = E_{q_\lambda(z|x)} \left[\log p_\theta(x|z) + \log p(z)\right] - E_{q_\lambda(z|x)} \left[\log q_\lambda(z|x)\right]$$

This simplifies to:

$$E_z\left[\log p_\theta(x|h_\lambda(x)+\sigma\epsilon)+\log p(h_\lambda(x)+\sigma\epsilon)\right]-E_z\left[\log q_\lambda(h_\lambda(x)+\sigma\epsilon|x)\right]$$

The gradient w.r.t. λ is:

$$\nabla_\lambda\Omega(\lambda)=E_z\left[\nabla_\lambda\log p_\theta(x|h_\lambda(x)+\sigma\epsilon)+\nabla_\lambda\log p(h_\lambda(x)+\sigma\epsilon)\right]-E_z\left[\nabla_\lambda\log q_\lambda(h_\lambda(x)+\sigma\epsilon|x)\right]$$

Note in this method, the ratio is gone and we can obtain are fairly good estimate. However, this method has some drawbacks because it must be in the Gaussian form. It may not be applicable in discrete cases.

In the next lecture, we are going to see how to apply VAE to derive diffusion models.