6.7900 Fall 2024: Lecture Notes 21

In this lecture, our goal is to generalize the latent structure from finite discrete class to infinite continuous class, such that we can study more complex entities such as images and molecules. Our model will consist of 2 parts: $P(z|x)$ as the inference part and $Q(x|z)$ as the generative part (see Figure 1). The two parts are connected via a simple yet powerful formula: $Q(x)Q(z|x) = P(z)P(x|z)$. We will discuss how to carry out the two parts approximately in complicated models.

Figure 1: Our goal

1 GMM: Motivation of EM

Recall in the Gaussian Mixture Model

$$P(x|\theta) = \sum_{z=1}^{k} P(z|\theta)P(x|z,\theta) = \sum_{z=1}^{k} \pi_z \mathcal{N}(x|\mu_z, \Sigma_z).$$

For simplicity, let's assume $\Sigma_z = \sigma_z^2 I$. We can estimate the parameters by MLE

$$\ell(D;\theta) = \sum_{i=1}^{N} \log P(x^{(i)}|\theta) = \sum_{i=1}^{N} \log \left(\sum_{z=1}^{k} \pi_z \mathcal{N}(x^{(i)}|\mu_z, \sigma_z^2 I) \right).$$

Here, we assume the data generating process follows an i.i.d. manner. It can be established that

$$\nabla_\theta \log P(x|\theta) = \sum_{z=1}^{k} P(z|x,\theta) \nabla_\theta \log P(x,z|\theta),$$

where $P(x,z|\theta) = P(z|\theta)P(x|z,\theta)$ is the complete likelihood of (x,z).

A pseudo EM algorithm works as follows.

E-step: Calculate $Q(z|x) = P(z|x, \theta)$.

M'-step: Update parameters using gradient ascent

$$\theta \leftarrow \theta + \eta \sum_{z=1}^{k} Q(z|x) \nabla_\theta \log P(x, z|\theta)$$

A natural question arises: in the second step, can we make several updates or even just do maximization with fixed $Q(z|x)$? The answer is yes:

M-step: Update parameters solving the maximization problem:

$$\theta \leftarrow \arg\max \sum_{i=1}^{N} \sum_{z=1}^{k} Q(z|x^{(i)}) \nabla_\theta \log P(x^{(i)}, z|\theta).$$

The maximization problem is easy to do in closed form under Gaussian Mixture Model.

Exercise: How should we handle the case that contains partially missing data? Specifically, for certain indices i we have complete pairs $(x^{(i)}, z^{(i)})$, but for others, z can be missing.

2 ELBO: Justification of EM

For the analysis below, we take a single x. ELBO$(Q; \theta)$ is defined as

$$\begin{aligned}
\mathbb{E}_{z\sim Q(\cdot|x)} \left[\frac{P(x, z|\theta)}{Q(z|x)} \right] &= \sum_{z=1}^{k} Q(z|x) \log[P(z|\theta)P(x|z, \theta)] + H(Q_{z|x}) \\
&= \sum_{z=1}^{k} Q(z|x) \log[P(z|\theta)P(x|z, \theta)] + \sum_{z} Q(z|x) \log \frac{1}{Q(z|x)} \\
&= \log P(x|\theta) - \mathrm{KL}(Q_{z|x} \| P_{z|x,\theta}) \\
&\leq \log P(x|\theta)
\end{aligned}$$

The last inequality can also be proved by Jensen's Inequality:

$$\begin{aligned}
&\sum_{z=1}^{k} Q(z|x) \log[P(z|\theta)P(x|z, \theta)] + H(Q_{z|x}) \\
&= \sum_{z=1}^{k} Q(z|x) \log \left[\frac{P(z|\theta)P(x|z, \theta)}{Q(z|x)} \right] \\
&\leq \log \left[\sum_{z=1}^{k} Q(z|x) \cdot \frac{P(z|\theta)P(x|z, \theta)}{Q(z|x)} \right] \\
&= \log P(x|\theta).
\end{aligned}$$

The inequality is tight if and only if $Q(z|x) = P(z|x, \theta)$. This means ELBO is a valid lower bound of the log-likelihood, and it coincides with the log-likelihood if and only if $Q(z|x) = P(z|x, \theta)$. Thus, we can view EM algorithm as follows:

E-step: maximize ELBO$(Q; \theta)$ with respect to Q.

M-step: maximize ELBO$(Q; \theta)$ with respect to θ.

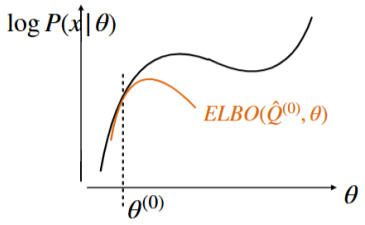

Figure 2: ELBO vs. log-likelihood

This view also generates a simple and elegant proof of the convergence of EM algorithm: it always increase the likelihood in each iteration.

$$\log P(x|\theta^{(0)}) = \text{ELBO}(Q^{(0)}, \theta^{(0)}) \leq \text{ELBO}(Q^{(0)}, \theta^{(1)}) \leq \text{ELBO}(Q^{(1)}, \theta^{(1)}) = \log P(x|\theta^{(1)}).$$

Lastly, we would like to point out that in practical EM implementation, initialization matters a lot — different initial points can lead to different locally optimal points (see Slides pp.12-21). Another issue is how to set k. There is no uniform criterion in practice, and generally people use a validation dataset combined with prior knowledge to decide an appropriate k.

3 Deep Generative Models and VAEs

We now consider more general latent mechanics: what if there are many classes or even continuous classes? An example is as follows: $z \sim N(0, I)$, $x \sim \mathcal{N}(g(z; \theta), \sigma^2 I)$ where $g(z; \theta) = r\frac{z}{\|z\|} + m$.

The idea of ELBO

$$\text{ELBO}(Q; \theta) = \int Q(z|x) \log[P(z)P(x|z, \theta)]dz + H(Q_{z|x})$$

can be used to optimize any mixture (discrete or continuous). A simple idea is as follows:

E'-step: infer likely z from x.

M'-step: update θ based on the complete data.

The problem of the approach above is that the inference step can be very difficult to do precisely without approximation. To solve it, we further parameterize ELBO as follows:

$$\begin{aligned}\text{ELBO}(Q_\phi;\theta) &= \int Q(z|x,\phi)\log[P(z)P(x|z,\theta)]dz + H(Q_{z|x,\phi}) \\ &= \int Q(z|x,\phi)\log[P(x|z,\theta)]dz - \text{KL}(Q_{z|x,\phi}\|P_z)\end{aligned}$$

where $Q(z|x,\phi)$ and $P(x|z,\theta)$ are two (deep neural) networks, corresponding to "encoder" and "decoder". The encoder and decoder are both learned via stochastic gradient ascent steps on ELBO. This leads to the VAE structure.

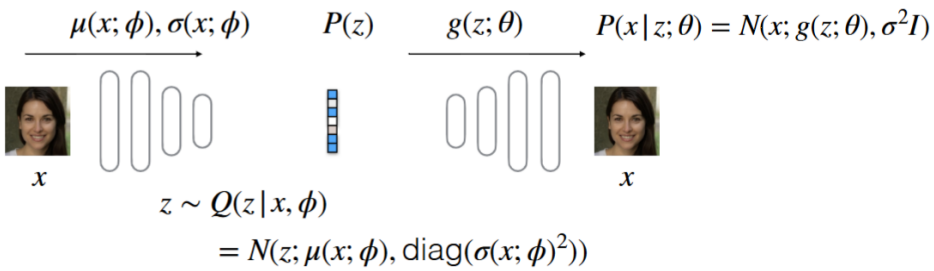

Figure 3: VAE structure. $P(z)$ is generally taken as a simple distribution, e.g., $\mathcal{N}(0, I)$.

When optimizing ELBO, the KL term is easy to calculate (divergence between 2 Gaussian distributions), but the first term is hard to optimize. A straightforward idea is to approximate the integral by sampling. However, a direct sampling would not retain any dependence on ϕ, making it hard to optimize the "encoder". We need to introduce a trick called **Reparametrization**:

$$z_\phi(x,\epsilon) = \mu_\phi(x) + \sigma_\phi(x)\odot\epsilon$$

where $\epsilon \sim \mathcal{N}(0, I)$. Then

$$\int Q(z|x,\phi)\log[P(x|z,\theta)]dz = \mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)}\left[\log P(x|z_\phi(x,\epsilon),\theta)\right] \approx \log P(x|z_\phi(x,\hat{\epsilon}),\theta)$$

See Slides pp.46 for the full optimization procedure of VAE.