COMP 7070 Advanced Topics in AI and ML March 12, 2024

Lecture 9: Variational Autoencoders

Instructor: Yifan Chen *Scribes: Xuanzhe XIAO* *Proof reader: Zhanke Zhou*

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

9.1 Revisiting Maximum Likelihood Estimation

Consider the scenario where we are modeling data using $p_{x,z}(\cdot,\cdot;\boldsymbol{\theta})$, in which $\boldsymbol{\theta}$ represents the parameters of our model, and $x, \mathbf{z}$ are the observed and hidden variables, respectively. These hidden variables could take the form of labels or hidden embeddings, among others. Often, we employ maximum likelihood estimation (MLE) to estimate these parameters:

$$\hat{\theta} = \operatorname*{argmax}_{\theta} P(X;\theta).$$

Here, $P(X;\theta)$ is the marginal likelihood of X, also known as the *evidence*.

For models incorporating hidden variables, the marginal likelihood $P(X;\theta)$ can be calculated using the formula:

$$P(X;\theta) = \int P(X \mid z;\theta) \cdot P(z;\theta)\, dz.$$

In some complex models, this integral is challenging to compute, meaning that the marginal likelihood can be difficult to evaluate. This creates obstacles for maximizing $P(X;\theta)$. In the following example of a Bayesian Gaussian mixture model, we will see how difficult it can be to compute $P(X;\theta)$.

9.1.1 Bayesian Gaussian Mixture Model

Consider a Bayesian mixture of unit-variance univariate Gaussians. There are K mixture components, corresponding to K Gaussian distributions with means $\mu = \{\mu_1, \ldots, \mu_K\}$. The mean parameters are drawn independently from a common prior $p(\mu_k)$, which we assume to be a Gaussian $\mathcal{N}\left(0, \sigma^2\right)$; the prior variance σ^2 is a hyperparameter. To generate an observation x_i from the model, we first choose a cluster assignment c_i. It indicates which latent cluster x_i comes from and is drawn from a categorical distribution over $\{1, \ldots, K\}$. (We encode c_i as an indicator K-vector, all zeros except for a one in the position corresponding to x_i 's cluster.) We then draw x_i from the corresponding Gaussian $\mathcal{N}\left(c_i^\top \mu, 1\right)$. The full hierarchical model is

$$\begin{aligned} \mu_k &\sim \mathcal{N}\left(0, \sigma^2\right), && k = 1, \ldots, K, \\ c_i &\sim \text{Categorical}(1/K, \ldots, 1/K), && i = 1, \ldots, n, \\ x_i \mid c_i, \mu &\sim \mathcal{N}\left(c_i^\top \mu, 1\right), && i = 1, \ldots, n. \end{aligned}$$

For a sample of size n, the joint density of latent and observed variables is

$$p(\mu, c, x) = p(\mu) \prod_{i=1}^{n} p\left(c_i\right) p\left(x_i \mid c_i, \mu\right).$$

The latent variables are $\mathbf{z} = \{\mu, c\}$, the K class means and n class assignments. Here, the evidence is

$$p(x) = \int p(\mu) \prod_{i=1}^{n} \sum_{c_i} p\left(c_i\right) p\left(x_i \mid c_i, \mu\right) \mathrm{d}\mu. \tag{9.1}$$

The integrand in Equation 9.1 does not contain a separate factor for each μ_k. (Indeed, each μ_k appears in all n factors of the integrated.)

Thus, the integral in Equation 9.1 does not reduce to a product of one-dimensional integrals over the μ_k 's. The time complexity of numerically evaluating the K-dimensional integral is $\mathcal{O}\left(K^n\right)$ (Since the polynomial inside the integral is composed of n polynomials each with K terms, expanding this polynomial will result in K^n terms).

If we distribute the product over the sum in 9.1 and rearrange, we can write the evidence as a sum over all possible configurations c of cluster assignments,

$$p(x) = \sum_c p(c) \int p(\mu) \prod_{i=1}^{n} p\left(x_i \mid c_i, \mu\right) \mathrm{d}\mu.$$

Here each individual integral is computable, thanks to the conjugacy between the Gaussian prior on the components and the Gaussian likelihood. But there are K^n of them, one for each configuration of the cluster assignments. Computing the evidence remains exponential in K, hence intractable.

9.2 Evidence Lower Bound

Given the aforementioned challenges of directly maximizing $P(X;\theta)$ in the presence of latent variables, here we introduce an indirect approach that involves maximizing something called the *Evidence Lower Bound* (ELBO), which is a core technique in variational inference. Let's see how this is done. We can introduce a new distribution $q(z)$ and transform the logarithm of the evidence as follows:

$$\begin{aligned}
\log P(X;\theta) &= \log \int P(X,z;\theta)\, dz \\
&= \log \int P(X,z;\theta) \cdot \frac{q(z)}{q(z)}\, dz \\
&= \log \int \frac{P(X,z;\theta)}{q(z)} q(z)\, dz \\
&= \log \mathbb{E}_q \left[\frac{P(X,z;\theta)}{q(z)}\right] \\
&\geqslant \mathbb{E}_q \log \frac{P(X,z;\theta)}{q(z)} \\
&= \underbrace{\mathbb{E}_q \log P(X,z;\theta) - \mathbb{E}_q \log q(z)}_{\text{ELBO}}.
\end{aligned}$$

The inequality comes from the convexity of the logarithm function. Let's subtract the logarithm of the evidence from the ELBO and see what the difference between the two is:

$$\begin{aligned}
\log P(X;\theta) - \text{ELBO} &= \log P(X;\theta) - (\mathbb{E}_q[\log P(X,z;\theta)] - \mathbb{E}_q[\log q(z)]) \\
&= \mathbb{E}_q[\log P(X;\theta)] - \mathbb{E}_q[\log P(X,z;\theta)] + \mathbb{E}_q[\log q(z)] \\
&= \mathbb{E}_q\left[\log \frac{P(X;\theta)}{P(X,z;\theta)}\right] + \mathbb{E}_q[\log q(z)] \\
&= \mathbb{E}_q\left[\log \frac{q(z)}{P(z|X;\theta)}\right] \\
&= \mathrm{KL}(q(z) \| P(z|X;\theta)).
\end{aligned}$$

Hence, maximizing the ELBO is essentially about minimizing the Kullback-Leibler divergence between $q(z)$ and $P(z|X;\theta)$. Therefore, the central concept of variational inference lies in identifying the optimal q from a given distribution family.

Furthermore, based on the derivation above, it is also easy to see that the ELBO can be decomposed into the log evidence and the difference between the Kullback-Leibler divergence of $q(z)$ from $P(z|X;\theta)$, that is:

$$\text{ELBO} = \log P(X;\theta) - \mathrm{KL}(q(z) \parallel P(z|X;\theta)).$$

9.3 The Expectation-Maximization Algorithm

The Expectation Maximization (EM) algorithm is essentially a coordinate ascent that maximizes $\text{ELBO}(\theta, q_z)$.

(E-step) First, we optimize $\text{ELBO}\,(\theta, q_z)$ over q_z with fixed θ. Based on the previous decomposition of ELBO, we know:

$$\text{ELBO}(\theta, q_z) = \underbrace{\log P_X(X;\theta)}_{\text{unrelated to } q_z} - \text{KL}\left(q_Z(\cdot) \| P_{Z|X}(\cdot \mid X;\theta)\right).$$

For given θ, we denote the best q_z maximizing ELBO as q_z^*, and we have

$$q_z^*(\theta) \triangleq \underset{q_z}{\text{argmax}}\, \text{ELBO}\,(\theta, q_z) = p_{z|x}(\cdot \mid x;\theta).$$

(M-step) Next, we will optimize over θ with fixed q_z. Suppose the current (old) parameter is θ' and plug $q_z = q_z^*\,(\theta') = p_{z|x}\,(\cdot \mid x;\theta')$ into $\text{ELBO}\,(\theta, q_z)$, and we aim to find

$$\theta^* \triangleq \underset{\theta}{\text{argmax}}\, \text{ELBO}\left(\theta, p_{z|x}\left(\cdot \mid x;\theta'\right)\right).$$

Since $\text{ELBO}\,(\theta, q_z) = \mathbb{E}_{z\sim q_z}\left[\log p_{x,z}(x,z;\theta)\right] \underbrace{-\mathbb{E}_{z\sim q_x}\left[\log q_z(z)\right]}_{\text{The Entropy of } q_z}$,

$$\text{ELBO}\left(\theta, p_{z|x}\left(\cdot \mid x;\theta'\right)\right) = \mathbb{E}_{z\sim p_{z|x}(\cdot|x;\theta')}\left[\log p_{x,z}(x,z;\theta)\right] + \underbrace{H\left(p_{z|x}\left(\cdot \mid x;\theta'\right)\right)}_{\text{unrelated to } \theta},$$

where $H(p)$ represents the entropy of a distribution p. If we define

$$U\left(\theta;\theta'\right) \triangleq \mathbb{E}_{z\sim p_{zx}(\cdot|x;\theta')}\left[\log p_{x,z}(x,z;\theta)\right],$$

the optimal (new) becomes $\theta^* = \text{argmax}_\theta\, U\,(\theta;\theta')$. The two steps are repeated until convergence.

9.4 Variational Auto-Encoder

Suppose we want to infer the parameters θ in such a model:

1. $z \sim \mathcal{N}(0, I)$.
2. $X \sim \mathcal{N}(\text{D}(z;\theta), I)$, where $\text{D}(\cdot)$ is a Neural Network called decoder.

Unless $\text{D}(\cdot)$ is linear, the posterior probability $P(z|X)$ does not have a closed-form expression. This necessitates the adoption of the variational approach mentioned earlier, introducing $q(z)$. In variational autoencoders, $q(z)$ is constrained to the family of isotropic Gaussian distributions and can be expressed as:

$$q\,(z_i \mid X_i) = N(\mu(X;\theta), \Sigma(X;\theta)),$$

where $\mu(X;\theta)$ and $\Sigma(X;\theta)$ are obtained through an encoder, which is also a neural network.

Therefore, the process of maximizing the evidence can also be achieved by maximizing the evidence lower bound. Let's review the ELBO, which can be decomposed as:

$$\mathbb{E}_{z\sim q(z|X)} \log P(X \mid z;\theta) - KL(q(z \mid X) \| p(z)).$$

This consists of two parts. Let's analyze them one by one:

1. **Reconstruction Loss:** For this part $\mathbb{E}_{z\sim q(z|X)} \log P(X \mid z;\theta)$, we can sample a z from $q(z \mid X)$. Note that $P(X \mid z;\theta)$ is Gaussian, and the logarithm of this term is proportional to $\frac{1}{2}\|x - \text{D}(z)\|^2$. Therefore, the reconstruction loss can be approximated using mean squared error loss.
2. **Variational Regularization Term:** Another part involves minimizing $KL(q(z \mid X) \| p(z))$. However, both q and p are normal distributions, which means the KL divergence between them can be expressed in closed form.

Thus, we can train the variational autoencoder by maximizing the ELBO.

9.5 Generative Adversarial Network

Generative Adversarial Networks involve a basic adversarial game between a generator G_θ and a discriminator D_φ. The generator aims to generate fake data as realistically as possible, while the discriminator aims to distinguish between real data and fake data generated by the generator. In terms of probability models, the generative model minimizes the maximum likelihood of the discriminator, as shown below:

$$\min_{G_\theta} \max_{D_\varphi} \mathbb{E}_{X,Y} \left[D_\varphi(X)\right]^Y \left[1 - D_\varphi(X)\right]^{1-Y}.$$

Here, X represents the observed data, which could be real or fake. Y denotes the label indicating whether the data is real or fake, and in this context, it is a latent variable. The discriminator $D_\varphi(X)$ is used to estimate the probability that the data is real, i.e., $P(Y = 1 \mid X)$. The generator, on the other hand, generates data distribution when the data is fake, defined as $G_\theta(X) \triangleq P(X \mid Y = 0)$.

Upon converting the likelihood to a logarithmic form, we can continue with the subsequent derivation:

$$\begin{aligned}
&\min_{G_\theta} \max_{D_\varphi} \mathbb{E}_{X,Y} \log \left[D_\varphi(X)\right]^Y \left[1 - D_\varphi(X)\right]^{1-Y} \\
=&\min_{G_\theta} \max_{D_\varphi} \left\{\mathbb{E}_{X|Y=1} \log \left[D_\varphi(X)\right] + \mathbb{E}_{X|Y=0} \log \left[1 - D_\varphi(Y)\right]\right\} \\
=&\min_{G_\theta} \max_{D_\varphi} \left\{\mathbb{E}_{P_{\text{data}}} \log \left[D_\varphi(X)\right] + \mathbb{E}_{P_{G_\theta}} \log \left[1 - D_\varphi(Y)\right]\right\}.
\end{aligned}$$