COMP 7070 Advanced Topics in AI and ML March 19, 2025

Lecture 9: Variational Autoencoders

Instructor: Yifan Chen *Scribes: Sitian Chen, Xuetao Chen* *Proof reader: Xiong Peng*

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

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.*

9.1 Revisiting Maximum Likelihood Estimation

Consider the scenario where we are modeling data using $p(x, z; \theta)$, in which θ represents the fixed parameters of our model, and x, z are the observed and hidden variables, respectively. The hidden variables z could take the form of labels or hidden embeddings, among others. We typically employ maximum likelihood estimation (MLE) to estimate these parameters:

$$\hat{\theta} = \arg\max_{\theta} P(X; \theta)$$

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:

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

In this scenario, z is assumed to follow a Gaussian prior distribution:

$$z \sim \mathcal{N}(\mu, \sigma^2)$$

where μ is the mean, and σ^2 is the variance. Here, μ is typically a fixed parameter (often part of θ), while σ^2 can either be fixed or learned.

In many cases, this integral becomes challenging to compute, particularly for Gaussian priors, which complicates the process of maximizing $P(X; \theta)$. As we will see in the case of a Bayesian Gaussian mixture model, evaluating $P(X; \theta)$ is not always tractable, presenting obstacles in the estimation process.

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 $N(0, \sigma^2)$; 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. This assignment 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 vector of length K, with all zeros except for a one in the position corresponding to the cluster of x_i.) We then draw x_i from the corresponding Gaussian $N(c_i^\top \mu, 1)$.

The full hierarchical model is:

$$\mu_k \sim N(0, \sigma^2), \quad k = 1, \ldots, K \tag{9.1}$$

$$c_i \sim \text{Categorical}\left(\frac{1}{K}, \ldots, \frac{1}{K}\right), \quad i = 1, \ldots, n \tag{9.2}$$

$$x_i | c_i, \mu \sim N(c_i^\top \mu, 1), \quad i = 1, \ldots, n \tag{9.3}$$

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

$$p(\mu, c, x) = p(\mu) \prod_{i=1}^{n} p(c_i) p(x_i | c_i, \mu) \tag{9.4}$$

The latent variables are $z = \{\mu, c\}$, consisting of the K class means and the n class assignments. The evidence, i.e., the marginal likelihood of x, is:

$$p(x) = \int p(\mu) \prod_{i=1}^{n} \sum_{c_i} p(c_i) p(x_i | c_i, \mu) d\mu \tag{9.5}$$

The integrand in Equation (9.4) does not contain a separate factor for each μ_k. In fact, each μ_k appears in all n factors of the integrand. Therefore, the integral in Equation (9.5) does not reduce to a product of one-dimensional integrals over the μ_k's. The time complexity of numerically evaluating this K-dimensional integral is $O(Kn)$.

By distributing the product over the sum in Equation (9.5) and rearranging, we can write the evidence as a sum over all possible configurations c of cluster assignments:

$$p(x) = \sum_{c} p(c) \int p(\mu) \prod_{i=1}^{n} p(x_i | c_i, \mu) d\mu \tag{9.6}$$

Here, each individual integral is computable, thanks to the conjugacy between the Gaussian prior on the components and the Gaussian likelihood. However, there are K^n such integrals, one for each possible configuration of the cluster assignments. Thus, computing the evidence remains exponential in K, making it intractable.

9.2 Evidence Lower Bound

Given the aforementioned challenges of directly maximizing $P(X; \theta)$ in the presence of latent variables, we introduce an indirect approach called the Evidence Lower Bound (ELBO), which is central to variational inference. The main idea is to optimize an objective function that serves as a lower bound on the log marginal likelihood (the evidence). Let's break this down in detail.

We start by expressing the log evidence as follows:

$$\log P(X; \theta) = \log \int P(X, z; \theta) \, dz \tag{9.7}$$

where $P(X, z; \theta)$ represents the joint likelihood of the observed data X and the latent variables z. Directly maximizing this log evidence is difficult due to the integral over the latent variables.

Introducing a Variational Distribution $q(z)$. To make this optimization tractable, we introduce a new distribution $q(z)$, which approximates the true posterior $P(z|X; \theta)$. This allows us to transform the logarithm of the evidence into an expectation with respect to $q(z)$. We multiply and divide by $q(z)$ under the integral:

$$\log P(X; \theta) = \log \int P(X, z; \theta) \, \frac{q(z)}{q(z)} \, dz \tag{9.8}$$

Next, we can express this as an expectation with respect to $q(z)$:

$$= \log \mathbb{E}_q \left[\frac{P(X, z; \theta)}{q(z)} \right] \tag{9.9}$$

Applying Jensen's Inequality. Since the logarithm is a concave function, we can apply Jensen's inequality to obtain a lower bound:

$$\log P(X; \theta) \geq \mathbb{E}_q \left[\log P(X, z; \theta) - \log q(z) \right] \tag{9.10}$$

This inequality ensures that we have a tractable lower bound to maximize, known as the Evidence Lower Bound (ELBO). The term on the right-hand side is easier to handle compared to the original log marginal likelihood.

Decomposing the Evidence and ELBO. To better understand the relationship between the log evidence and the ELBO, let's subtract the ELBO from the log evidence:

$$\log P(X;\theta) - \text{ELBO} = \log P(X;\theta) - \mathbb{E}_q\left[\log P(X,z;\theta) - \log q(z)\right] \tag{9.11}$$

This simplifies to:

$$= \mathbb{E}_q\left[\log q(z) - \log P(z|X;\theta)\right] \tag{9.12}$$

Notice that the right-hand side is simply the Kullback-Leibler (KL) divergence between the variational distribution $q(z)$ and the true posterior distribution $P(z|X;\theta)$:

$$= \text{KL}(q(z)\,||\,P(z|X;\theta)) \tag{9.13}$$

Thus, we conclude that maximizing the ELBO is equivalent to minimizing the KL divergence between $q(z)$ and $P(z|X;\theta)$. This tells us that the central idea of variational inference is to find an approximation $q(z)$ that is as close as possible to the true posterior $P(z|X;\theta)$.

From the above, we can see that the ELBO is decomposed into two parts: the log evidence and the KL divergence. We can write this as:

$$\text{ELBO} = \log P(X;\theta) - \text{KL}(q(z)\,||\,P(z|X;\theta)) \tag{9.14}$$

Therefore, by maximizing the ELBO, we are effectively maximizing the log evidence while minimizing the difference between the approximate posterior $q(z)$ and the true posterior $P(z|X;\theta)$. This forms the foundation of variational inference, where the goal is to optimize the variational distribution $q(z)$ to make it as close as possible to the true posterior.

9.3 The Expectation-Maximization Algorithm

The Expectation-Maximization (EM) algorithm is a coordinate ascent method to estimate the maximum likelihood of models with latent variables. It can be used to estimate the θ and q_z by maximizing the ELBO.

The algorithm iteratively maximizes $\text{ELBO}(q_z,\theta)$ through two steps: the E-step, which optimizes q_z with fixed θ, and the M-step, which optimizes θ with fixed q_z. These two steps are repeated until convergence. maximizes $\text{ELBO}(q_z,\theta)$.

1. **E-step:** Optimize $\text{ELBO}(q_z,\theta)$ over q_z with fixed θ. The optimal $q_z^*(\theta)$ satisfies:

$$\begin{aligned} q_z^*(\theta) &= \arg\max_{q_z} \text{ELBO}(q_z,\theta) \\ &= \arg\min_{q_z} \text{KL}(q_z \parallel p(z|X;\theta)) \\ &\approx p(z|X;\theta). \end{aligned}$$

2. **M-step:** Optimize over θ with fixed q_z. The new parameter θ^* is:

$$\theta^* = \arg\max_{\theta} \mathbb{E}_q[\log p(X,z;\theta)] - \mathbb{E}_q[\log q(z)].$$

where $\mathbb{E}_q[\log q(z)]$ is fixed. The two steps are repeated until convergence.

9.3.1 Example 1

9.3.1.1 Problem Definition

Type of Problem: Maximum Likelihood Estimation (MLE) in probabilistic models with unobserved latent variables Z, given observed data X.
Goal: Maximize the log-likelihood of the observed data $\log P(X; \theta)$.

Inputs

- Observed data $X = x$
- Latent variables Z
- Parameters to estimate θ

Mathematical Formulation

Q-Function Definition

For discrete Z:

$$Q(\theta \mid \theta^{(k)}) = \sum_Z P(Z \mid X = x; \theta^{(k)}) \log P(X, Z; \theta)$$

For continuous Z:

$$Q(\theta \mid \theta^{(k)}) = \int_Z P(Z \mid X = x; \theta^{(k)}) \log P(X, Z; \theta) \, dZ$$

9.3.1.2 Steps

1. **Initialize**: Set initial parameters $\theta^{(0)}$ (randomly or heuristically).
2. **Iterate until convergence** ($k = 0, 1, 2, \ldots$):
   - **E-step (Expectation)**:
     Compute the posterior distribution of Z:
     $$q(Z) = P(Z \mid X = x; \theta^{(k)})$$
     Construct the Q-function (expected complete-data log-likelihood):
     $$Q(\theta \mid \theta^{(k)}) = \mathbb{E}_{Z \sim q(Z)} \left[\log P(X, Z; \theta) \mid X = x\right]$$
   - **M-step (Maximization)**:
     Update parameters by maximizing the Q-function:
     $$\theta^{(k+1)} = \arg\max_\theta Q(\theta \mid \theta^{(k)})$$

9.3.2 Example 2

9.3.2.1 Problem Definition

Observations $Y_1, \ldots, Y_n$ are generated from a mixture of two exponential distributions:

$$P_\theta(y) = \frac{1}{2} e^{-y} + \frac{\theta}{2} e^{-\theta y}, \quad y > 0,$$

where the latent variable $J_i \in \{0, 1\}$ indicates the component:

$$J_i = \begin{cases} 0, & Y_i \sim \text{Exp}(1) \text{ (rate 1)}, & \text{w.p. } \frac{1}{2}, \\ 1, & Y_i \sim \text{Exp}(\theta) \text{ (rate } \theta), & \text{w.p. } \frac{1}{2}. \end{cases}$$

Goal: Estimate θ using the EM algorithm.

Mathematical Formulation

Q-Function Definition

$$Q(\theta|\theta^{(k)}) = \sum_{i=1}^{n} E[\log p(Y_i, J_i; \theta)|Y_i = y_i; \theta^{(k)}]$$

9.3.2.2 Steps

1. **Initialize**: Set initial parameters $\theta^{(0)}$ (randomly or heuristically).
2. **Iterate until convergence** ($k = 0, 1, 2, \ldots$):

   - **E-step (Posterior Weights)**:
     For each $Y_i = y_i$, compute the posterior probability $w_i^{(k)} = P(J_i = 1 \mid Y_i = y_i; \theta^{(k)})$:

     $$w_i^{(k)} = \frac{\frac{\theta^{(k)}}{2} e^{-\theta^{(k)} y_i}}{\frac{1}{2} e^{-y_i} + \frac{\theta^{(k)}}{2} e^{-\theta^{(k)} y_i}} = \frac{\theta^{(k)} e^{-\theta^{(k)} y_i}}{e^{-y_i} + \theta^{(k)} e^{-\theta^{(k)} y_i}}.$$

     Construct the Q-function:

     $$Q(\theta \mid \theta^{(k)}) = \sum_{i=1}^{n} \left[w_i^{(k)} \log \left(\frac{\theta}{2} e^{-\theta y_i} \right) + (1 - w_i^{(k)}) \log \left(\frac{1}{2} e^{-y_i} \right) \right].$$

     Simplifying (ignoring constants w.r.t θ):

     $$Q(\theta \mid \theta^{(k)}) = C + \sum_{i=1}^{n} w_i^{(k)} \left(\ln \theta - \theta y_i \right).$$

   - **M-step: Parameter Update** Maximize Q-function by taking derivative w.r.t θ:

     $$\frac{\partial Q}{\partial \theta} = \sum_{i=1}^{n} w_i^{(k)} \left(\frac{1}{\theta} - y_i \right) = 0 \quad \Rightarrow \quad \theta^{(k+1)} = \frac{\sum_{i=1}^{n} w_i^{(k)}}{\sum_{i=1}^{n} w_i^{(k)} y_i}.$$

9.4 Variational Auto-Encoder

Suppose we want to infer the parameters θ in the following model:

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

a latent variable z is sampled from the prior distribution $z \sim \mathcal{N}(0, I)$, where I denotes the identity matrix. This sampled z is then passed through a decoder neural network $D(z; \theta)$ (parameterized by θ) to generate the output X. Specifically, the generated X follows the conditional distribution $\mathcal{N}(D(z; \theta), I)$, where the mean is determined by the decoder's output and the covariance remains the identity matrix I.

Since $D(\cdot)$ is a neural network, the model is non-linear, and the posterior probability $P(z|X)$ does not have a closed-form expression. Thus, we cannot use the EM algorithm to infer the parameters θ in this model. This necessitates the adoption of the variational approach mentioned earlier, introducing an approximate posterior $q(z)$. In variational autoencoders, $q(z)$ is constrained to the family of isotropic Gaussian distributions and can be expressed as:

$$q(z_i|X_i) = \mathcal{N}(\mu(X; \theta), \Sigma(X; \theta)),$$

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 (ELBO). The ELBO can be decomposed as:

$$\text{ELBO} = \mathbb{E}_{z\sim q(z|X)}\left[\log P(X|z;\theta)\right] - \text{KL}(q(z|X) \parallel p(z;\theta)).$$

This consists of two parts:

1. **Reconstruction Loss:** The term $\mathbb{E}_{z\sim q(z|X)}\left[\log P(X|z;\theta)\right]$ can be approximated by sampling z from $q(z|X)$. Since $P(X|z;\theta)$ follows a Gaussian distribution, the log-likelihood of this term can be expressed as:

   $$\log P(X|z;\theta) \propto -\frac{1}{2}\left\|X - D(z;\theta)\right\|^2$$

   Therefore, the reconstruction loss can be approximated using the mean squared error (MSE) loss.

2. **Variational Regularization Term:** The second term minimizes the KL divergence $\text{KL}(q(z|X) \parallel p(z;\theta))$. Since both q and p are normal distributions, their KL divergence has a closed-form expression.

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