CSE6243: Advanced Machine Learning **Fall 2023**

Lecture 12: VAE and Diffusion models

Lecturer: Bo Dai *Scribes: Eshani Chauk, Harsha Karanth*

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

12.1 Recap

Energy Based Model Topics:

1. Contrastive Divergence: This optimization algorithm approximates the gradient of log-likelihood and updates parameters of a model to fit the data. It is mainly used for Restricted Boltzmann Machines (RBMs).

2. Score Matching: Other method than maximum likelihood to calculate probability distribution and it focuses on a non-parametric approach

Autoregressive model vs Restricted Boltzmann Machines (RBMs):

Autoregressive model: A model that learns from a list of timed steps and measurements/observations taken at each time step to predict the next time step.

Restricted Boltzmann Machines (RBMs): Unsupervised neural network that learns a probability distribution over a set of input.

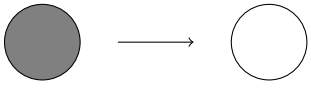
Autoregressive model vs RBMs Comparsion:

Autoregressive models	RBMs
No clear relationship between variables: $\prod_{i=1}^{d} p(x_i \mid x_{<i})$	Clear relationship between variables like x and h
Simpler to calculate (use MLE)	Harder to solve, need to use an estimator
Can produce sequences easier	Need to iterate many times to get one sample using MCMC

12.2 New Content

12.2.1 Latent Variable Model (Variational autoencoder)

x is variable we observe, and z is the variable that influences x but we cannot observe z



Z X

The probability of x (p(x)) is equal to

$$p(x) = \int p_\theta(x|z)p_\phi(z)dz$$

Energy based models are too simple to model the data. As a result, we use a latent variable model to model data.

An example of a LVM is the Gaussian Mixture Model

$$p(x|z) \sim N(\mu_z, \sigma_z)$$

Dogs Cats

Dogs and Cats represent clusters. We use a Gaussian model to to group the data into clusters.

12.2.2 Ineffective ways of learning parameters of LVM

We only have x, need to learn z

D=$\{x_i\}_{i=1}^n$

Perform MLE:

$$\max_{\theta,\phi} \frac{1}{n} \sum_{i=1}^{n} log(p(x_i))$$

$$\max_{\theta,\phi} \frac{1}{n} \sum_{i=1}^{n} log(\int p_\theta(x_i|z_i) * p_\phi(z_i)dz)$$

$$L(\theta, \phi) = log(\int p_\theta(x_i|z_i) * p_\phi(z_i)dz)$$

Attempt to calculate gradient:

$$L(\theta, \phi) = \frac{1}{n} \sum_{i=1}^{n} \frac{\int \Delta_\theta p_\theta(x_i|z_i)p_\phi(z_i)dz_i}{\int p_\theta(x_i|z_i)p_\phi(z_i)dz_i}$$

It is hard to calculate this because it is a ratio between integrals. Additionally, it is difficult to calculate the unbiased estimator for θ and ϕ

Using Score matching will result in same issue.

12.2.3 Using Auxiliary function in MLE and ELBO

We will use an auxiliary function to simplify the expression and avoid the ratio of integrals.

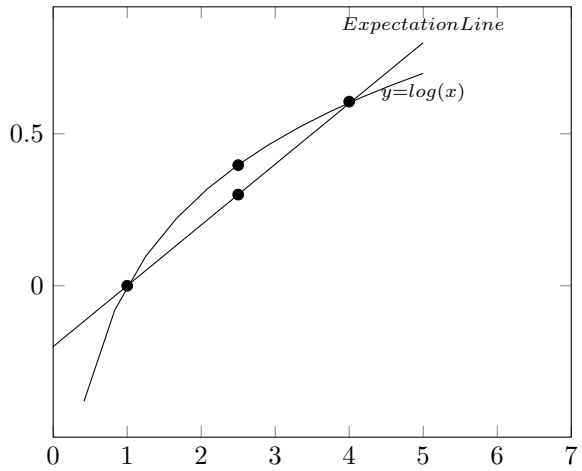
log p(x) =

$$log(\int p(x|z)p(z)dz)$$

$q(z|x)$ is the auxiliary function

$$log(\int \frac{p(x|z)p(z)}{q(x|z)}q(x|z)dz)$$

$$log(\mathbb{E}_{q(z|x)}\left[\frac{p(x,z)}{q(z|x)}\right])$$



We know the expectation of a log function is less than the log value:

$\mathbb{E}[log(f(z))] \leq log(\mathbb{E}[f(z)])$

so we can say

$$log(\mathbb{E}_{q(z|x)}\left[\frac{p(x,z)}{q(z|x)}\right]) \geq \mathbb{E}_{q(z|x)}\left[log(\frac{p(x,z)}{q(z|x)})\right]$$

This is called Evidence Based Lower Bound (ELBO). This is a key concept in Variational Bayesian Methods because it changes intractable inference problems to solvable optimization problems. We continue expanding this lower bound and use it as an approximation/value for log(p(x)).

To confirm we can use this value in our calculations, we can set $q(z|x) = \frac{p(x,z)}{p(x)}$

$\mathbb{E}_{q(z|x)}\left[log(\frac{p(x,z)}{q(z|x)})\right] = \mathbb{E}_{q(z|x)}\left[log(p(x))\right] = \int q(z|x)log(p(x))dz = log(p(x))\int q(z|x)dz$

$q(z|x)$ is a distribution so the integral of it is 1

$log(p(x)) \int q(z|x)dz = \log(\text{p(x)})$

Using ELBO, log(p(x)) will equal

$$\max_{q(z|x)} \mathbb{E}_{q(z|x)} \left[log(\frac{p(x,z)}{q(z|x)}) \right]$$

$$\max_{q(z|x)} \mathbb{E}_{q(z|x)} \left[log(p(x,z)) - log(q(z|x)) \right]$$

$$\max_{q(z|x)} \mathbb{E}_{q(z|x)} \left[log(p(x,z)) \right] - \int q(z|x) log(q(z|x))$$

$$\max_{q(z|x)} \mathbb{E}_{q(z|x)} \left[log(p(x,z)) \right] - H(q)$$

where $\text{H(q)} = \int q(z|x) log(q(z|x))$ and represents entropy

12.2.4 Adding λ Parameter to MLE

Recall that the original MLE equation is

$$\max_{\theta,\phi} \frac{1}{n} \sum_{i=1}^{n} \log p(x_i)$$

The MLE equation can be redefined with a new objective function such that we have the following equation:

$$\max_{\theta,\phi} \max_{q(z|x)} \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{q(z|x)} \left[\log \frac{p_\theta(x|z) p_\phi(z)}{q(z|x)} \right]$$

The reason q is used is that it helps to estimate the gradient that updates model parameters during the training process. A new parameter λ defines the q function, which can also be learned in training, and the MLE function is now

$$\max_{\theta,\phi} \max_{\lambda} \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{q_\lambda(z|x)} \left[\log \frac{p_\theta(x|z) p_\phi(z)}{q_\lambda(z|x)} \right]$$

where the likelihood $\mathcal{L}(\theta, \phi, \lambda)$ is given by

$$\frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{q_\lambda(z|x)} \left[\log \frac{p_\theta(x|z) p_\phi(z)}{q_\lambda(z|x)} \right]$$

Now, to update each of these parameters via gradient descent, the gradient with respect to each of these parameters must be calculated individually. Once the gradient is found, MC approximation or gradient descent can be used to get an unbiased estimator that updates the parameter.

Gradient w.r.t θ:

$$\nabla_\theta \mathcal{L}(\theta, \phi, \lambda) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{q(z|x)} \left[\nabla_\theta \log p_\theta(x|z) \right]$$

Gradient w.r.t. ϕ:

$$\nabla_\phi \mathcal{L}(\theta, \phi, \lambda) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{q(z|x)} \left[\nabla_\phi \log p_\phi(z) \right]$$

This gradient w.r.t. λ is derived with the following steps:

$$\mathbb{E}_{q_\lambda(z|x)} \left[-\log(q_\lambda(z|x)) \right]$$

$$= \nabla_\lambda \int q_\lambda(z|x)(-\log(q_\lambda(z|x)))dz$$

$$= -\int \nabla_\lambda q_\lambda(z|x) \log(q_\lambda(z|x))dz - \int q_\lambda(z|x)(-\nabla_\lambda \log(q_\lambda(z|x))dz$$

The right side becomes 0 the integral of the distribution is 1, and taking the derivative of that is 0. The left side is reduced with the following steps:

$$-\int \nabla_\lambda q_\lambda(z|x) \log(q_\lambda(z|x))dz$$

$$= -\int \frac{\nabla_\lambda q_\lambda(z|x)}{q_\lambda(z|x)} q_\lambda(z|x) \log(q_\lambda(z|x))dz$$

$$= -\int \nabla_\lambda \log(q_\lambda(z|x)) q_\lambda(z|x) \log(q_\lambda(z|x))dz$$

$$= -\mathbb{E}_{q_\lambda(z|x)} \left[(\nabla_\lambda \log(q_\lambda(z|x))) \log(q_\lambda(z|x)) \right]$$

Thus, the gradient w.r.t. λ is:

$$\nabla_\lambda \mathcal{L}(\theta, \phi, \lambda) = \mathbb{E}_{q_\lambda(z|x)} \left[\nabla_\lambda \log(q_\lambda(z|x)) \log \frac{p_\theta(x|z) p_\lambda(z)}{q_\lambda(z|x)} \right]$$

The VAE model relies on random sampling from a distribution determined by the θ, ϕ and λ parameters. However, when backtracking to update parameters with the gradients found above, it is not possible to perform this update on all of the parameters due to the random sampling. Thus, the parametrization trick is introduced.

12.2.5 Reparametrization Trick for Backpropagation

This method involves parametrizing how z is sampled. A new function $z = T(x, \varepsilon, \lambda)$ is defined where T is the equivalent of random sampling that allows for backpropagation through all of the parameters. x is the input, λ is the parameter, and ε is the noise. It is also know that $q \sim p_0(q)$. Then, the MLE equation can be modified once again:

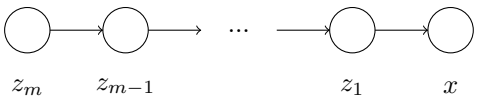
$$\max_{\theta,\phi} \max_{\lambda} \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{q_\lambda(z|x)} \left[\log p(x, z) - \log q_\lambda(z|x) \right]$$

$$= \mathbb{E}_{p_0(\varepsilon)} \left[\log p(x, T(x, \varepsilon, \lambda)) - \log p_0(T(x, \varepsilon, \lambda)) \right]$$

$$= \mathbb{E}_{\varepsilon} \left[\nabla_\lambda \log p(x, T(x, \varepsilon, \lambda)) - \nabla_\lambda \log p_0(T(x, \varepsilon, \lambda)) \right]$$

12.2.6 Denoising Diffusion Process with VAE

A noise vector z_m can be reconstructed into an image by using a VAE to denoise the vector. This process is represented with the following graph where z_m is the original noisy vector and the latent variables z_{m-1} to z_1 remove noise to create x:



The latent variable model is

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

$$= \int p(x|z_1)p(z_1|z_2)...p(z_{m-1}|z_m)p(z_m)dz_{1...m}$$

This $p(x)$ is not a diffusion model yet, but a special parametrization based on ELBO is used to make it a diffusion model:

$$\mathbb{E}_{q(z_{1...m}|x)}\left[\log \frac{p(x, z_{1...m})}{q(z_{1...m}|x)}\right]$$

$$q(z_{1...m}|x) = \prod_{i=1}^{n} q(z_{i+1}|z_i) \text{ where } q(z_{i+1}|z_i) = \mathcal{N}(\sqrt{1-\beta_i}z_i, \beta_i)$$

$$p(z_i|z_{i-1}) = \mathcal{N}(\mu(z_{i+1}, i+1), \eta_{i+1}I)$$

$$z_{i+1} = \sqrt{1-\beta_i}z_i + \varepsilon_i \text{ where } \varepsilon_i \sim \mathcal{N}(0, \beta_i I)$$

This topic is covered more in-depth in Lecture 13.