10-707 Deep Learning **Spring 2020**

Lecture 11: February 24, 2019

Lecturer: Andrej Risteski *Scribes: Saket Dingliwal, Alexander Pei*

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

11.1 Introduction

Markov Chain Monte Carlo(MCMC) and Variational Methods are two common techniques used in graphical models. MCMC methods usually involve doing a random walk with equilibrium distribution as the one we are trying to sample from. On the other hand, variational methods rely on solving some form of optimization problem. In this lecture, we are particularly interested in Variational Methods to solve the following two problems in graphical models

- **Inference**: Given the parameters θ of the model, the goal of inference is to sample/calculate the marginal distributions. For example sampling $p_\theta(x_1)$ or $p_\theta(x|z)$. The task is easy for fully-observable Bayesian Nets, RBM's while harder for undirected fully-observable models or latent variable Bayesian nets as computing the normalization is harder.
- **Learning**: Given the observed data, the goal of Learning is to find the values for parameters θ for the model. Typically, this involves maximizing the likelihood function, which is same as solving the following optimization problem.

$$\max_{\theta \in \Theta} \sum_{i=1}^{n} log\ p(x_i)$$

11.2 Inference

11.2.1 Inference in undirected models

The problem of inference often reduces to calculation of the partition function. For example consider an Ising model

$$P_\theta(x) = \frac{1}{Z(\theta)} exp(\sum_{i,j \in E} x_i x_j \theta_i j + \sum_{i \in V} x_i \theta_i)$$

any marginal computation for let's say $p_\theta(x_k = 1)$ can be thought of as computing partition functions for the denominator (the complete $Z(\theta)$) and another modified partition function for the numerator (another quadratic). Therefore, our goal here is to calculate/approximate the partition function. We reduce it to optimization problem using the principle given below.

Theorem 11.1 ***Gibbs Variational Principle:*** *Let $p(x) = \frac{1}{Z} exp(E(x))$ be a distribution over the domain $\mathcal{X}$. Then, Z is the solution to the following optimization problem:*

$$log\, Z = \max_{q:\ distribution\ over\ \mathcal{X}} H(q) + \mathbb{E}_{x \sim q}[E(x)]$$

where $H(\cdot)$ is entropy of the distribution and $E(\cdot)$ is any energy function.

Proof: The proof follows from non-negativity of KL divergence. Consider KL divergence between $p(x)$ as defined above and some $q(x)$ as

$$0 \leq KL(q||p) = \mathbb{E}_q \log q - \mathbb{E}_q \log p = -H(q) - \mathbb{E}_{x\sim q}[E(x)] + \log Z$$

$$H(q) + \mathbb{E}_{x\sim q}[E(x)] \leq \log Z$$

$$\max_{q\text{: distribution over } \mathcal{X}} H(q) + \mathbb{E}_{x\sim q}[E(x)] \leq \log Z$$

There is equality ie $KL(q||p) = 0$ when $p = q$, so we have

$$\max_{q\text{: distribution over } \mathcal{X}} H(q) + \mathbb{E}_{x\sim q}[E(x)] = \log Z$$

■

11.2.1.1 Examples

We have reduced the computation problem to optimization problem. Here, we see an example of how we can solve this optimization problem over the set of all distributions over $\mathcal{X}$. Even if $\mathcal{X}$ is a really simple domain, eg. $\mathcal{X} = \{-1, +1\}^n$, the trivial way to solve the problem would involve introducing 2^n variables to parameterise q which is hard. Therefore, we make assumptions and restrict the class of distributions. For example, let $p(x) = \prod_i \ p_i(x_i)$, then for each $i \in [n]$, we only need to specify $p(x_i) = 1$. Since the number of parameters are small, we can take gradient wrt variables $p_i(x_i)$ and do gradient descent. Consider the case where $n = 2$ below

$$\max_{q=\prod_i q_i} H(q) + \mathbb{E}_{x\sim q}[E(x)]$$

So the entropy term in the objective factorizes as shown below

$$H(q_1, q_2) = \sum_{x_1,x_2\in -1,+1} q_1(x_1)q_2(x_2)log(q_1(x_1)q_2(x_2))$$

changing product inside log into sum, we get

$$H(q_1, q_2) = \sum_{x_2\in -1,+1} q_2(x_2)H(q_1) + \sum_{x_1\in -1,+1} q_1(x_1)H(q_2) = H(q_1) + H(q_2)$$

Similarly, we can handle the energy term $\mathbb{E}_{x\sim q}[E(x)]$, for example for the Ising model under the product assumption, we can write

$$\mathbb{E}_q[E(x)] = \sum_{ij} J_{ij}\mathbb{E}[x_i]\mathbb{E}[x_j]$$

and we know that $\mathbb{E}[x_j] = q_j(x_j)$, so taking the gradient is simple.
Similarly another example could be to find the parameters of mean and covariance matrix for Gaussians. Then the terms can be factorized and computed as follows

- Entropy for Gaussian $\mathcal{N}(\mu, \Sigma)$ has a closed form formula as given by $\frac{1}{2}log((2\pi e)^n det\Sigma)$
- Expectations wrt Gaussians can be estimated by drawing samples.

11.2.2 Inference in latent-variable Bayesian Networks

Oftentimes it is desirable to calculate the posterior distribution in a given problem. However, the denominator $p(x)$ of the posterior distribution is oftentimes an intractable integral. Referred to as the model evidence or marginal likelihood, it is usually not possible to integrate over all parameters.

$$p(z|x) = \frac{p(z)p(x|z)}{p(x)}$$

$$p(x) = \int_\theta p(x|\theta)p(\theta)\mathrm{d}\theta$$

Variation methods for posterior distributions estimate the posterior by introducing a new distribution $q(z|x)$. Similarly to the previous section, the divergence between the original posterior and this new distribution $q(z|x)$ is:

$$KL(q(z|x)||p(z|x)) = \mathbb{E}_{q(z|x)} \log q(z|x) - \mathbb{E}_{q(z|x)} \log p(z|x)$$

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

Therefore, choosing to our new posterior $q(z|x)$ to maximize the KL-divergence is written as:

$$\text{argmax}_{q(z|x)} KL(q(z|x)||p(z|x)) = \text{argmax}\{\mathbb{E}_{q(z|x)} \log q(z|x) - \mathbb{E}_{q(z|x)} \log p(z|x)\}$$

This approximate posterior can be optimized using coordinate ascent by updating a single coordinate under the mean-field approximation. We will keep $q_{-i}(z_i|x)$ fixed and optimize from $q_i(z_i|x)$.

$$KL(q(z|x)||p(z|x)) = \mathbb{E}_{q(z|x)} \log q(z|x) - \mathbb{E}_{q(z|x)} \log p(z|x)$$

$$= \sum_i \mathbb{E}_{q_i(z_i|x)} \log q_i(z_i|x) - \mathbb{E}_{q_i(z_i|x)} \left[\mathbb{E}_{q_{-i}(z_{-i}|x)} \log p(z_i, z_{-i}, x) \right]$$

$$= \mathbb{E}_{q_i(z_i|x)} \log q_i(z_i|x) - \mathbb{E}_{q_i(z_i|x)} \left[\log \tilde{p}(z_i, x) \right] + C$$

$$= KL(q_i(z_i|x)||\tilde{p}(z_i, x)) + C$$

The optimum for $q_i(z_i|x) = \tilde{p}(z_i, x))$ is:

$$\frac{\mathbb{E}_{q_{-i}(z_{-i}|x)} \log p(z_i, z_{-i}, x)}{\int_{z_i} \mathbb{E}_{q_{-i}(z_{-i}|x)} \log p(z_i, z_{-i}, x)}$$

11.3 Learning

11.3.1 Learning latent-variable directed graphical models

Again, our problem of learning for a directed graphical model begins with maximum likelihood:

$$\max_{\theta \in \Theta} \sum_{i=1}^{n} log\ p(x_i)$$

We want the maximum likelihood estimate of the parameters to best fit our given data. Using the variational principle again, we can write this as:

$$\log_\theta p(x) = \max_{q(z|x)\text{: distribution over} Z} H(q(z|x)) + \mathbb{E}_{q(z|x)}[\log p_\theta(x,z)]$$

$$= \max_{\theta \in \Theta} \max_{\{q_i(Z|X_i)\}} \sum_{i=1}^{n} H(q_i(z|x_i)) + \mathbb{E}_{q_i(z|x_i)}[\log p_\theta(x_i,z)]$$

11.3.2 Variational methods for posterior distributions

ELBO (Evidence lower bound)

Similar to before, we are interested in introducing a new distrubtion q to approximate the posterior distribution $p(z|x)$. The name comes directly from lower bounding the model evidence $p(x) = \int p(x,z)dz$ by performing the marginalization of z. Let $p(z,x)$ be a joint distribution over latent variables and observables. Then:

$$\log p(x) = \max_{q(z|x)\text{: distribution over} Z} H(q(z|x)) + \mathbb{E}_{q(z|x)}[\log \theta(x,z)]$$

Again by using Bayes rule for the posterior distribution using the joint, the above equation is can expressed by the Gibbs variational principle with the energy function $E(x) = \log p(x,z)$

Gibbs variational principle

Let $p(x) = \frac{1}{z}\exp(E(x))$ be a distribution over a domain X. Then, Z is the solution to the following optimization problem:

$$\log Z = \max_{q\text{: distribution over} X} H(q) + \mathbb{E}_{x\ q}[E(x)]$$

This method provides us with an approximation to the posterior in a tractable way.

Expectation-maximization/ variational inference

Expectation-maximization can help us estimate model parameters when the model contains latent variables that we cannot observe directly. From before, the MLE objective is:

$$\max_{\theta\in\Theta}\max_{\{q_i(Z|X_i)\}}\sum_{i=1}^{n}H(q_i(z|x_i))+\mathbb{E}_{q_i(z|x_i)}[\log p_\theta(x_i,z)]$$

We wish to iteratively update $\theta^t, q_i^t(z|x_i)$. The EM algorithm is composed of two steps. The first step, the **Expectation (E)-step** is:

Keep fixed, set $q_i^t(z|x_i)$ to maximize the objective above.

The second step, the **Maximization (M)- step** is:

Keep $q_i^t(z|x_i)$ fixed, set θ^{t+1} to maximize the objective above.

Every single step improves the objective, however the optimization space is not always convex so this algorithm can get stuck in local minimum.

11.3.2.1 Examples

This example will cover the mixture of spherical Gaussians given by

$$p=\sum_{i=1}^{k}\frac{1}{k}\mathcal{N}(\mu_i,I_d)$$

E-step - assigning all of the points to the nearest cluster. This assignment is soft, meaning that each point will have a probability of membership to each cluster.

By using Bayes rule,

$$p_{\theta^t}(z=k|x_i)\propto p(x_i|z=k)\propto\exp\{-||x_i-\mu_k^t||^2\}$$

$$p_{\theta^t}(z=k|x_i)=\frac{\exp\{-||x_i-\mu_k^t||^2\}}{\sum_{k'}\exp\{-||x_i-\mu_{k'}^t||^2\}}$$

M-step Updating the centers of the clusters based on the memberships assigned during the E-step

$$\max_{\theta\in\Theta}\sum_{i=1}^{n}H(q_i^t(z|x_i))+\mathbb{E}_{q_i^t(z|x_i)}[\log p_\theta(x_i,z)]$$

Cancelling out the terms that do not depend on θ leaves:

$$\max_{\theta\in\Theta}\sum_{i=1}^{n}\mathbb{E}_{q_i^t(z|x_i)}[\log p_\theta(x_i|z)]$$

This is equal to:

$$\max_{\theta}-\sum_{i=1}^{n}\sum_{k=1}^{K}q_i^t(z=k|x_i)||x_i-\mu_k||^2$$

Solving for the new cluster centers by taking the partial derivative with respect to θ and setting it to zero gives:

$$\mu_K^{t+1} = \sum_{i=1}^{n} \frac{\exp\{-||x_i - \mu_k^t||^2\}}{\sum_{k'} \exp\{-||x_i - \mu_{k'}^t||^2\}}$$

Which is updating the new cluster centerse to be the average of the memberships of all the points assigned to it.