208891 Probabilistic Graphical Models **Spring 2019**

Lecture 14: February 24

Lecturer: Donlapark Pornnopparath

Continued from last time, we have shown that the KL-divergence is nonnegative.

$$D_{KL}(p\|q) = -\int p(x)\log\frac{q(x)}{p(x)}\,dx \geq 0,$$

with equality holds if and only if p/q is a constant. Since that constant has to be one, we obtain

$$D_{KL}(p\|q) = 0 \quad \iff \quad p \equiv q.$$

14.1 EM for learning latent variable models

Consider a graphical model $\boldsymbol{Z} \to \boldsymbol{X}$ where $\boldsymbol{Z}$ consists of latent variables. After the variables $\boldsymbol{X} = \boldsymbol{x}$ are observed, the parameters can be learned using the EM algorithm as follows:

- *E-Step*: Compute $q(\boldsymbol{z}) = p(\boldsymbol{z}|\boldsymbol{x}, \theta_t)$
- *M-step*: $\theta_{t+1} = \arg\max_\theta Q(\theta_t, \theta)$ where

$$\begin{aligned} Q(\theta_t, \theta) &= \mathbb{E}_{q(\boldsymbol{z})} \log p(\boldsymbol{x}, \boldsymbol{z}|\theta) \\ &= \mathbb{E}_{q(\boldsymbol{z})} \log\frac{p(\boldsymbol{x}, \boldsymbol{z}|\theta)}{q(\boldsymbol{z})} + \mathbb{E}_{q(\boldsymbol{z})} \log q(\boldsymbol{z}). \end{aligned}$$

Since the last term is a constant that does not depend on θ, we can instead try to maximize

$$\begin{aligned} \mathcal{L}(q, \theta) &= \mathbb{E}_{q(\boldsymbol{z})} \log\frac{p(\boldsymbol{x}, \boldsymbol{z}|\theta)}{q(\boldsymbol{z})} \\ &= -D_{KL}(q(\boldsymbol{z})\|p(\boldsymbol{z}|\boldsymbol{x}, \theta)) + \log p(\boldsymbol{x}|\theta) \\ &\leq \log p(\boldsymbol{x}|\theta). \end{aligned}$$

Because of this property, $\mathcal{L}(q, \theta)$ is called the evidence lower bound (ELBO).

We will now show that $\{p(\boldsymbol{x}|\theta_t)\}_t$ is an increasing sequence. Since θ_{t+1} maximizes the ELBO, we have

$$\begin{aligned} \mathcal{L}(q, \theta_{t+1}) &\geq \mathcal{L}(q, \theta_t) \\ &\geq p(\boldsymbol{x}|\theta_t) - D_{KL}(q(\boldsymbol{z})\|p(\boldsymbol{z}|\boldsymbol{z}, \theta_t)) \\ &= p(\boldsymbol{x}|\theta_t). \end{aligned}$$

It follows that

$$p(\boldsymbol{x}|\theta_{t+1}) \geq \mathcal{L}(q, \theta_{t+1}) \geq p(\boldsymbol{x}|\theta_t)$$

as desired. Therefore, $\{p(\boldsymbol{x}|\theta_t)\}_t$ is a sequence that is bounded above by $\max_\theta p(\boldsymbol{x}|\theta)$, implying that the EM algorithm must converge. Figure 14.1 illustrates how the ELBO and θ_t change over the course of EM algorithm. However, one must be careful as the converging point can be either a local minimum, a local minimum or a saddle, depending on the choice of the initial parameters. Thus it is recommended to use EM with random restarts and choose the best parameters.

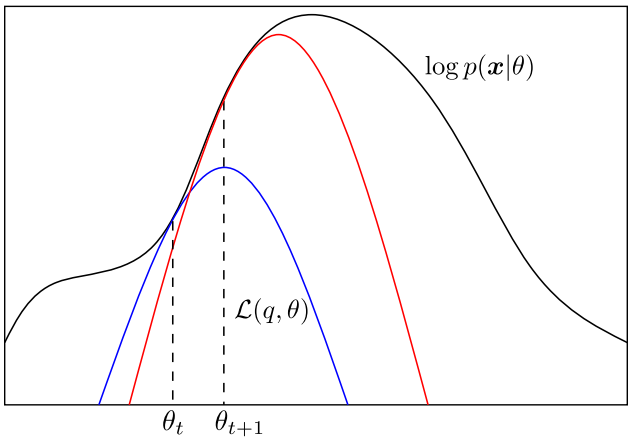

Figure 14.1: ELBO and parameter updates after an EM iteration.

14.2 Gaussian mixture model (GMM) revisited

In GMM, we try to fit a group of Gaussians $N(\mu_k, \Sigma_k)$ to the data D. The latent variable $Z \in \{z_1, \ldots, z_K\}$, distributed as $\text{Mult}(\pi_1, \ldots, \pi_K)$, assigns a probability that $\boldsymbol{x} \in D$ belongs to each group .

$$p(\boldsymbol{x}|z_k, \theta) = N(\mu_k, \Sigma_k),$$

where $\theta = \{\mu_k, \Sigma_k, \pi_k\}_{k=1}^K$. In the E-step, we try to maximize

$$\sum_{k=1}^{K} \sum_{\boldsymbol{x} \in D} p(z_k|\boldsymbol{x}, \theta_t) \log p(\boldsymbol{x}|z_k, \theta) + \sum_{k=1}^{K} \sum_{\boldsymbol{x} \in D} p(z_k|\boldsymbol{x}, \theta_t) \log p(z_k|\theta).$$

Each term in the first summation can be optimized separately.

$$\begin{aligned}
\mu_k, \Sigma_k &= \arg\max_{\mu,\Sigma} \sum_{\boldsymbol{x} \in D} p(z_k|\boldsymbol{x}, \theta_t) \log p(\boldsymbol{x}|z_k, \theta) \\
&= c_k \arg\max_{\mu,\Sigma} \sum_{\boldsymbol{x} \in D} \frac{p(z_k|\boldsymbol{x}, \theta_t)}{c_k} \log p(\boldsymbol{x}|z_k, \theta) \\
&= c_k D_{KL}(q(\boldsymbol{x}) \| p(\boldsymbol{x}|z_k, \theta)) + c_k \mathbb{E}_{q(\boldsymbol{x})} \log q(\boldsymbol{x}),
\end{aligned}$$

where $q(\boldsymbol{x}) = c_k^{-1} p(z_k|\boldsymbol{x}, \theta_t)$ is a probability mass function on D and $c_k = \sum_{x \in D} p(z_k|\boldsymbol{x}, \theta_t)$ is the normalizing factor. Since the second term is a constant which does not depend on θ, we only need to consider the first term. Since $p(\boldsymbol{x}|z_k, \theta)$ is in the exponential family, one can show that the KL-divergence which is minimized when the sufficient of the two distributions match i.e.

$$\mu_k = \mathbb{E}_{q(\boldsymbol{x})} \boldsymbol{x} = \sum_{\boldsymbol{x} \in D} \boldsymbol{x} q(\boldsymbol{x})$$

and

$$\Sigma_k = \mathbb{E}_{q(\boldsymbol{x})} \boldsymbol{x}\boldsymbol{x}^T - \mu_k \mu_k^T = \sum_{\boldsymbol{x} \in D} \boldsymbol{x}\boldsymbol{x}^T q(\boldsymbol{x}) - \mu_k \mu_k^T.$$

Similarly, by optimizing the distributions over z_k's we have that

$$\pi_k = \frac{\sum_{\boldsymbol{x} \in D} p(z_k|\boldsymbol{x}, \theta_t)}{\sum_{k=1}^{K} \sum_{\boldsymbol{x} \in D} p(z_k|\boldsymbol{x}, \theta_t)} = \frac{1}{|D|} \sum_{\boldsymbol{x} \in D} p(z_k|\boldsymbol{x}, \theta_t).$$

14.3 Exact inference: variable elimination

After resolving all parameters of a Bayesian network, we can use it to compute the probability of an event. For example, in the spam filtering task, we might want to find the marginal probability that an email is a spam.

$$p(y = 1) = \sum_{w_1} \cdots \sum_{w_j} p(y, w_1, \ldots, w_j).$$

For many complex distributions, computing the exact probability is NP-hard. This is not the case for a sparse Bayesian network.

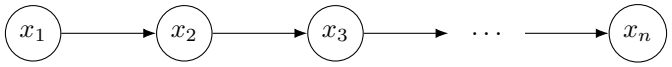

Figure 14.2: A Markov chain

Example 14.1. Consider a Markov chain in Figure 14.2 where $x_i \in \{1, 2, \ldots, k\}$. The joint pdf pf all variables is given by

$$p(x_1, \ldots, x_n) = p(x_1)p(x_2|x_1) \ldots p(x_n|x_{n-1}).$$

Suppose that we want to compute $p(x_n)$.

$$p(x_n) = \sum_{x_{n-1}} \ldots \sum_{x_1} p(x_1)p(x_2|x_1) \ldots p(x_n|x_{n-1}).$$

The right-hand side can be simplified by pushing each term through the summations

$$p(x_n) = \sum_{x_{n-1}} p(x_n|x_{n-1}) \sum_{x_{n-2}} \ldots \sum_{x_1} p(x_2|x_1)p(x_1).$$

Then we can "eliminate" each variable by marginalizing them out, starting from the root node: by defining $\tau_2(x_2) = \sum_{x_1} p(x_2|x_1)p(x_1)$ and $\tau_i(x_i) = \sum_{x_{i-1}} p(x_i|x_{i-1})\tau(x_{i-1})$ for $i = 3, \ldots, n$, we have

$$\begin{aligned}
p(x_n) &= \sum_{x_{n-1}} p(x_n|x_{n-1}) \sum_{x_{n-2}} \ldots \sum_{x_1} p(x_2|x_1)p(x_1) \\
&= \sum_{x_{n-1}} p(x_n|x_{n-1}) \sum_{x_{n-2}} \ldots \sum_{x_2} p(x_3|x_2)\tau_2(x_2) \\
&\vdots \\
&= \sum_{x_{n-1}} p(x_n|x_{n-1})\tau_{n-1}(x_{n-1}) \\
&= \tau_n(x_n).
\end{aligned}$$

This is more efficient than brute-forcing as it takes only $O(nk^2)$. ◇

Since any Bayesian network has topological ordering, we can easily extend the method shown in Example 14.1 to more general networks.

Variable elimination (VE) algorithm

Let $\mathcal{X} = \{X_1, X_2, \ldots, X_n\}$ be the set of all variables with topological ordering $X_1 > X_2 > \ldots > X_n$. To compute the probability $p(\boldsymbol{x}_o)$ where $\boldsymbol{x}_o$ are observed values of $\mathcal{X}_o \subseteq \mathcal{X}$, we perform the following steps:

1. Factorize the joint probability $p(X_1, \ldots, X_n) = \prod_{i=1}^{n} p(X_i|\mathrm{Pa}(X_i))$.
2. Write $p(\mathcal{X}_o) = \sum_{X_i \notin \mathcal{X}_o} p(X_1, \ldots, X_n)$ and plug in $\mathcal{X}_o = \boldsymbol{x}_o$.
3. Marginalize each $X_i \notin \mathcal{X}_o$ in order.

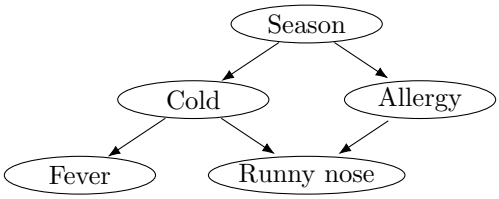

Figure 14.3: Medical diagnosis between cold and allergy.

Example 14.2. Suppose that we would like to compute $p(s, c)$ in Figure 14.3. Let the topological ordering be $S > C > A > R > F$. Then the variables A, R, F would be marginalized in that order. We can perform the VE algorithm as follows:

$$\begin{aligned} p(s,c) &= p(s,c) \sum_f p(f|c) \sum_r \sum_a p(r|c,a) p(a|s) \\ &= p(s,c) \sum_f p(f|c) \sum_r \tau_1(r). \end{aligned}$$

◇

To compute a conditional probability $p(\mathcal{X}_o = \boldsymbol{x}_o | E = e)$ where E consists of observed evidence variables, we write

$$p(\mathcal{X}_o = \boldsymbol{x}_o | E = e) = \frac{p(\mathcal{X}_o = \boldsymbol{x}_o, E = e)}{p(E = e)}$$

and perform VE algorithm on both $p(\mathcal{X}_o = \boldsymbol{x}_o, E = e)$ and $p(E = e)$.