Homework 4 of Statistical Machine Learning

Wang Yikai, 2017310740

June 29, 2018

1 Probabilistic Graphical Models

1.1 Conditional Queries in a Bayesian Network

1. The probabilistic graph of the model is shown in Figure 1. There are:

$$P(G_1) = (0.5, 0.5),\ P(G_i|G_1) = \begin{pmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{pmatrix},\ (i = 2, 3)$$
$$p(X_i|G_i = 1) = \mathcal{N}(X_i|\mu = 55, \sigma^2 = 10),\ (i = 1, 2, 3)$$
$$p(X_i|G_i = 2) = \mathcal{N}(X_i|\mu = 65, \sigma^2 = 10),\ (i = 1, 2, 3)$$

The **Markov blanket** of G_3 contains G_1 and X_3.

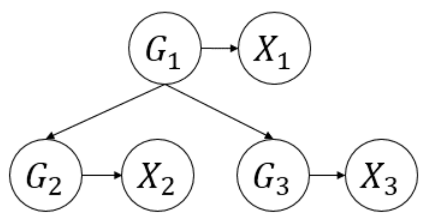

Figure 1: Probabilistic graph

2. Suppose X_2 is observed to be 50. In Python, uses **scipy.stats.norm.pdf([50], 55, np.sqrt(10))**, there is $P(X_2 = 50|G_2 = 1) = 0.03614$. Similarly, $P(X_2 = 50|G_2 = 2) = 1.6 \times 10^{-6}$. The posterior belief $P(G_1 = 1|X_2 = 50)$ is:

$$\begin{aligned}
P(G_1 = 1|X_2 = 50) &= \frac{P(G_1 = 1, X_2 = 50)}{P(X_2 = 50)} \\
&\propto P(G_1 = 1, X_2 = 50) \\
&= \sum_{i=1}^{2} P(G_1 = 1, G_2 = i, X_2 = 50) \\
&= \sum_{i=1}^{2} P(G_1 = 1)P(G_2 = i|G_1 = 1)P(X_2 = 50|G_2 = i) \\
&= 0.5 \times 0.9 \times 0.03614 + 0.5 \times 0.1 \times 1.6 \times 10^{-6} = 0.0163
\end{aligned}$$

$$
\begin{aligned}
P(G_1 = 2 | X_2 = 50) \propto P(G_1 = 2, X_2 = 50) &\\
&= \sum_{i=1}^{2} P(G_1 = 2) P(G_2 = i | G_1 = 2) P(X_2 = 50 | G_2 = i) \\
&= 0.5 \times 0.1 \times 0.03614 + 0.5 \times 0.9 \times 1.6 \times 10^{-6} = 0.0018
\end{aligned}
$$

As $P(G_1 = 1|X_2 = 50) + P(G_1 = 2|X_2 = 50) = 1$, there are:

$$P(G_1 = 1|X_2 = 50) = \frac{0.0163}{0.0163 + 0.0018} = 0.901$$

$$P(G_1 = 2|X_2 = 50) = \frac{0.0018}{0.0163 + 0.0018} = 0.099$$

3. Suppose we observe both $X_2 = 50$ and $X_3 = 50$. Then $P(G_1|X_2, X_3)$ is:

$$P(G_1 = 1|X_2 = 50, X_3 = 50) = \frac{P(G_1 = 1, X_2 = 50, X_3 = 50)}{P(X_2 = 50, X_3 = 50)} \propto P(G_1 = 1, X_2 = 50, X_3 = 50)$$

$$= \sum_{i=1}^{2} \sum_{j=1}^{2} P(G_1 = 1) P(G_2 = i|G_1 = 1) P(X_2 = 50|G_2 = i) P(G_3 = j|G_1 = 1) P(X_3 = 50|G_3 = j)$$

$$= P(G_1 = 1) \sum_{i=1}^{2} [P(G_2 = i|G_1 = 1) P(X_2 = 50|G_2 = i)] \sum_{j=1}^{2} [P(G_3 = j|G_1 = 1) P(X_3 = 50|G_3 = j)]$$

$$= 0.5 \times (0.9 \times 0.03614 + 0.1 \times 1.6 \times 10^{-6})^2 = 5.29 \times 10^{-4}$$

$$P(G_1 = 2|X_2 = 50, X_3 = 50) = 0.5 \times (0.1 \times 0.03614 + 0.9 \times 1.6 \times 10^{-6})^2 = 6.53 \times 10^{-6}$$

As $P(G_1 = 1|X_2 = 50, X_3 = 50) + P(G_1 = 2|X_2 = 50, X_3 = 50) = 1$, there are:

$$P(G_1 = 1|X_2 = 50, X_3 = 50) = \frac{5.29 \times 10^{-4}}{5.29 \times 10^{-4} + 6.53 \times 10^{-6}} = 0.988$$

$$P(G_1 = 2|X_2 = 50, X_3 = 50) = \frac{6.53 \times 10^{-6}}{5.29 \times 10^{-4} + 6.53 \times 10^{-6}} = 0.012$$

1.2 Conditional Random Fields

1. The undirected graph and the factor graph of the CRF are shown as Figure 2, 3:

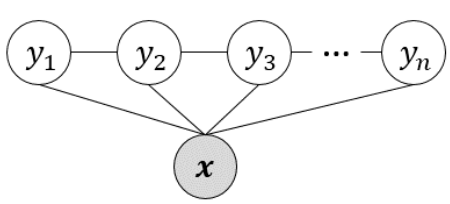

Figure 2: Undirected Graph

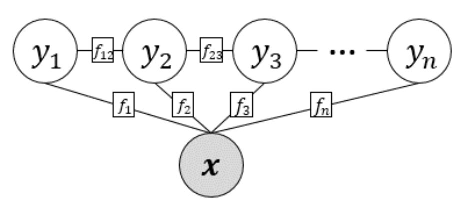

Figure 3: Factor Graph

2. The cliques of CRF is illustrated as Figure 4.

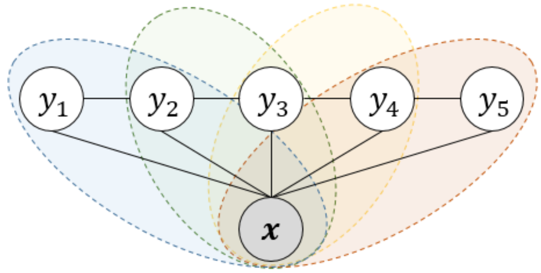

Figure 4: Cliques of CRF

The junction tree of CRF is illustrated as Figure 5.

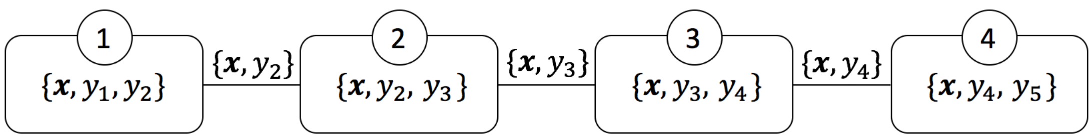

Figure 5: Junction Tree of CRF

Denote each clique in the junction tree as $C_i, i = 1, 2, 3, 4$. First, set initial factors at each cluster as products. For example, $\psi_1(\boldsymbol{x}, y_1, y_2) = \phi(\boldsymbol{x}, y_1)\phi(y_1, y_2)$.

By running the sum-product algorithm on the junction tree, we can get $p(y_3, \boldsymbol{x}; \boldsymbol{w})$:

- In C_1, eliminate y_1 by sending $\delta_{1\to 2}(\boldsymbol{x}, y_2) = \sum_{y_1} \psi_1(\boldsymbol{x}, y_1, y_2)$ to C_2;
- In C_2, eliminate y_2 by sending $\delta_{2\to 3}(\boldsymbol{x}, y_3) = \sum_{y_2} \delta_{1\to 2}(\boldsymbol{x}, y_2)\psi_2(\boldsymbol{x}, y_2, y_3)$ to C_3;
- In C_4, eliminate y_5 by sending $\delta_{4\to 3}(\boldsymbol{x}, y_4) = \sum_{y_5} \psi_4(\boldsymbol{x}, y_4, y_5)$ to C_3;
- In C_3, eliminate y_4 to get $p(y_3, \boldsymbol{x}; \boldsymbol{w}) = \sum_{y_4} \delta_{2\to 3}(\boldsymbol{x}, y_3)\delta_{4\to 3}(\boldsymbol{x}, y_4)\psi_3(\boldsymbol{x}, y_3, y_4)$.

After getting $p(y_3, \boldsymbol{x}; \boldsymbol{w})$, there is $p(\boldsymbol{x}; \boldsymbol{w}) = \sum_{y_3} p(y_3, \boldsymbol{x}; \boldsymbol{w})$. Then by $p(y_3|\boldsymbol{x}; \boldsymbol{w}) = \frac{p(y_3, \boldsymbol{x}; \boldsymbol{w})}{p(\boldsymbol{x}; \boldsymbol{w})}$, we can get $p(y_3|\boldsymbol{x}; \boldsymbol{w})$.

3. Given the following parametric form of CRF:

$$p(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{w}) = \frac{1}{Z(\boldsymbol{x}; \boldsymbol{w})} \exp\left\{\boldsymbol{w}^T \sum_{i=2}^{n} \boldsymbol{f}(\boldsymbol{x}, y_i, y_{i-1})\right\}$$

where

$$Z(\boldsymbol{x}; \boldsymbol{w}) = \sum_{\boldsymbol{y}' \in \mathcal{L}^n} \exp\left\{\boldsymbol{w}^T \sum_{i=2}^{n} \boldsymbol{f}(\boldsymbol{x}, y'_i, y'_{i-1})\right\}$$

To learn the parameters of the CRF, we maximize the conditional log likelihood of training data $\mathcal{D} = \{(\boldsymbol{x}^i, \boldsymbol{y}^i)\}_{i=1}^N$ over parameters $\boldsymbol{w}$ using gradient descent:

$$\mathcal{L}(\boldsymbol{w}) = \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}} \log p(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{w}) = \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}} \left\{ \boldsymbol{w}^T \sum_{i=2}^n \boldsymbol{f}(\boldsymbol{x}, y_i, y_{i-1}) - \log Z(\boldsymbol{x};\boldsymbol{w}) \right\}$$

Compute its gradient:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}} \left\{ \sum_{i=2}^n \boldsymbol{f}(\boldsymbol{x}, y_i, y_{i-1}) - \frac{1}{Z(\boldsymbol{x};\boldsymbol{w})} \frac{\partial Z(\boldsymbol{x};\boldsymbol{w})}{\partial \boldsymbol{w}} \right\}$$

where $\frac{1}{Z(\boldsymbol{x};\boldsymbol{w})} \frac{\partial Z(\boldsymbol{x};\boldsymbol{w})}{\partial \boldsymbol{w}}$ is:

$$\begin{aligned}
\frac{1}{Z(\boldsymbol{x};\boldsymbol{w})} \frac{\partial Z(\boldsymbol{x};\boldsymbol{w})}{\partial \boldsymbol{w}} &= \frac{1}{Z(\boldsymbol{x};\boldsymbol{w})} \frac{\partial \sum_{\boldsymbol{y}'\in\mathcal{L}^n} \exp\left\{\boldsymbol{w}^T \sum_{i=2}^n \boldsymbol{f}(\boldsymbol{x}, y'_i, y'_{i-1})\right\}}{\partial \boldsymbol{w}} \\
&= \frac{1}{Z(\boldsymbol{x};\boldsymbol{w})} \sum_{\boldsymbol{y}'\in\mathcal{L}^n} \exp\left\{\boldsymbol{w}^T \sum_{i=2}^n \boldsymbol{f}(\boldsymbol{x}, y'_i, y'_{i-1})\right\} \sum_{i=2}^n \boldsymbol{f}(\boldsymbol{x}, y'_i, y'_{i-1}) \\
&= \sum_{\boldsymbol{y}'\in\mathcal{L}^n} \frac{1}{Z(\boldsymbol{x};\boldsymbol{w})} \exp\left\{\boldsymbol{w}^T \sum_{i=2}^n \boldsymbol{f}(\boldsymbol{x}, y'_i, y'_{i-1})\right\} \sum_{i=2}^n \boldsymbol{f}(\boldsymbol{x}, y'_i, y'_{i-1}) \\
&= \sum_{\boldsymbol{y}'\in\mathcal{L}^n} p(\boldsymbol{y}'|\boldsymbol{x};\boldsymbol{w}) \sum_{i=2}^n \boldsymbol{f}(\boldsymbol{x}, y'_i, y'_{i-1}) \\
&= \mathbb{E}_{p(\boldsymbol{y}'|\boldsymbol{x};\boldsymbol{w})} \left[\sum_{i=2}^n \boldsymbol{f}(\boldsymbol{x}, y'_i, y'_{i-1}) \right]
\end{aligned}$$

Therefore $\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}}$ is:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} &= \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}} \left\{ \sum_{i=2}^n \boldsymbol{f}(\boldsymbol{x}, y'_i, y'_{i-1}) - \mathbb{E}_{p(\boldsymbol{y}'|\boldsymbol{x};\boldsymbol{w})} \left[\sum_{i=2}^n \boldsymbol{f}(\boldsymbol{x}, y'_i, y'_{i-1}) \right] \right\} \\
&= \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}} \sum_{i=2}^n \left\{ \boldsymbol{f}(\boldsymbol{x}, y'_i, y'_{i-1}) - \mathbb{E}_{p(\boldsymbol{y}'|\boldsymbol{x};\boldsymbol{w})} \left[\boldsymbol{f}(\boldsymbol{x}, y'_i, y'_{i-1}) \right] \right\}
\end{aligned}$$

4. Choose y_n as the root node, and apply forward propagation(O(n)) and backward propagation(O(n)). Therefore the whole process of belief propagation is O(n).

Forward propagation:

$$\begin{aligned}
&\alpha_0(y_0|\boldsymbol{x}) = 1(y_0 = start) \\
&\alpha_i(y_i|\boldsymbol{x}) = \delta_i^T(\boldsymbol{X}, y_i, y_{i-1})\alpha_{i-1}(y_{i-1}|\boldsymbol{x}),\ i = 1, \cdots, n+1
\end{aligned}$$

where

$$\delta_i(\boldsymbol{X}, y_i, y_{i-1}) = \exp\left\{\boldsymbol{w}^T \boldsymbol{f}(\boldsymbol{x}, y_i, y_{i-1})\right\}$$

Backward propagation:

$$\begin{aligned}
&\beta_{n+1}(y_{n+1}|\boldsymbol{x}) = 1(y_{n+1} = stop) \\
&\beta_i(y_i|\boldsymbol{x}) = \beta_{i+1}(y_{i+1}|\boldsymbol{x})\delta_{i+1}^T(\boldsymbol{x}, y_{i+1}, y_i),\ i = 1, \cdots, n+1
\end{aligned}$$

Here y_i has three probable values, which means α_i and β_i are both vectors with 3 dimensions and δ_i is a 3×3 matrix.

According to the definition:

$$\begin{aligned}
Z(\boldsymbol{x}; \boldsymbol{w}) &= \alpha_n^T(\boldsymbol{x}) \cdot \mathbf{1} = \mathbf{1}^T \cdot \beta_1(\boldsymbol{x}) \\
p(y_i|\boldsymbol{x}) &= \frac{\alpha_i^T(y_i|\boldsymbol{x})\beta_i(y_i|\boldsymbol{x})}{Z(\boldsymbol{x}; \boldsymbol{w})} \\
p(y_i, y_{i-1}|\boldsymbol{x}) &= \frac{\alpha_{i-1}^T(y_i|\boldsymbol{x})\delta_i(\boldsymbol{x}, y_i, y_{i-1})\beta_i(y_i|\boldsymbol{x})}{Z(\boldsymbol{x}; \boldsymbol{w})}
\end{aligned}$$

Therefore the expectation is:

$$\sum_{i=2} \mathbb{E}_{p(\boldsymbol{y'}|\boldsymbol{x},\boldsymbol{w})}[f(\boldsymbol{x}, y_i', y_{i-1}')] = \sum_{i=2}^{n} \sum_{y_i', y_{i-1}'} p(y_i', y_{i-1}'|\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x}, y_i', y_{i-1}')$$

2 Deep Generative Models: Class-conditioned VAE

1. The variational lower bound of Class-conditioned VAE is:

$$\begin{aligned}
\log p_\theta(x|y) &\geq \log p_\theta(x|y) - D_{KL}\left[q_\phi(z|x,y) \| p_\theta(z|x,y)\right] \\
&= \log p_\theta(x|y) - \mathbb{E}_{z\sim q_\phi(z|x,y)}\left[\log \frac{q_\phi(z|x,y)}{p_\theta(z|x,y)}\right] \\
&= \log p_\theta(x|y) - \mathbb{E}_{z\sim q_\phi(z|x,y)}\left[\log q_\phi(z|x,y) - \log p_\theta(x,z|y) + \log p_\theta(x|y)\right] \\
&= \mathbb{E}_{z\sim q_\phi(z|x,y)}\left[\log p_\theta(x,z|y) - \log q_\phi(z|x,y)\right] \\
&= \mathcal{L}(\theta, \phi)
\end{aligned}$$

Design the variational posterior as:

$$q_\phi(z|x,y) = N(z|\mu_z(x,y;\phi), \sigma_z^2(x,y;\phi))$$

2. The code is submitted, simply run main.py is ok.
3. Results(in the next page)

The generated result is illustrated in Figure 6.

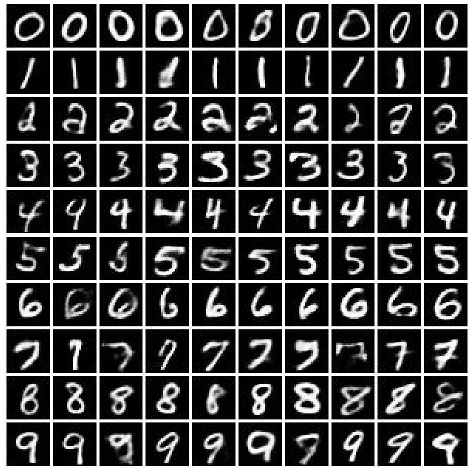

Figure 6: Generated Result

The variational lower bound of the training and testing process is illustrated in Figure 7.

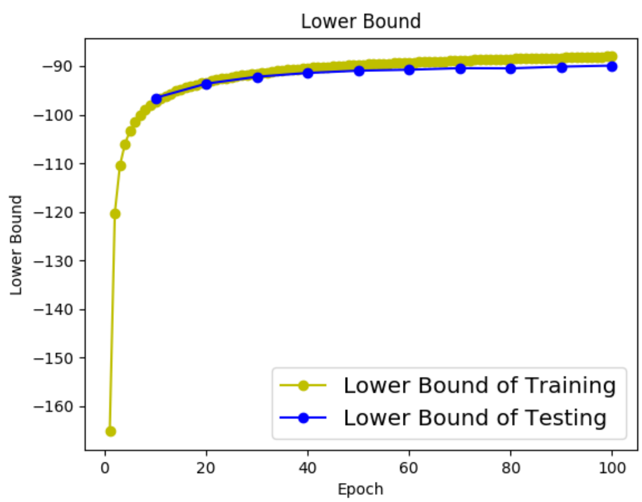

Figure 7: Lower Bound