CSE512 - Machine Learning - Homework 4

Yang Wang, 110014939
Computer Science Department
Stony Brook University
yang.wang@stonybrook.edu

1. One round of EM for a GMM

To learn a GMM with hidden variables, we need to maximize the observed data log-likelihood $L_X(\theta)$ with respect to the model's parameters $\theta = \{\pi_c, \mu_c, \sigma_c\}_{c=1}^{C}$, subject to $\sum_c \pi_c = 1$.

$$
\begin{aligned}
L_X(\theta) &= \log p(X|\theta) \\
&= \sum_i \log p(x^i|\theta) \\
&= \sum_i \log \sum_{z^i} p(x^i, z^i|\theta)
\end{aligned}
$$

This function is hard to maximize directly, instead we can iteratively maximize its tight lower bound function $Q(\theta, \theta^{cur})$.

$$
\begin{aligned}
\theta^{cur} &\leftarrow \arg\max_\theta Q(\theta, \theta^{cur}) \\
&= \arg\max_\theta \sum_i \sum_{z^i} p(z^i|x^i, \theta^{cur}) \log \frac{p(x^i, z^i|\theta)}{\cancel{p(z^i|x^i, \theta^{cur})}} \\
&= \arg\max_\theta \sum_i \sum_{z^i} p(z^i|x^i, \theta^{cur}) \log\{p(z^i|\theta)p(x^i|z^i, \theta)\} \\
&= \arg\max_\theta Q'(\theta, \theta^{cur})
\end{aligned}
$$

1.1. M step

1.1.1

In the **M step**, we need to **update the parameters** by maximizing the following function, subject to $\sum_c \pi_c = 1$.

$$
\begin{aligned}
Q'(\theta, \theta^{cur}) &= \sum_{i=1}^{3} \sum_{c=1}^{2} R_{i,c} \log\{\pi_c \times \frac{1}{\sqrt{2\pi}\sigma_c} \exp(-\frac{(x^i - \mu_c)^2}{2\sigma_c^2})\} \\
&= \sum_{i=1}^{3} \sum_{c=1}^{2} R_{i,c} \{\log \pi_c - \log \sigma_c - \frac{(x^i - \mu_c)^2}{2\sigma_c^2} + \log \frac{1}{\sqrt{2\pi}}\}
\end{aligned}
$$

The update rule for parameters is as follows.

$$
\pi_c^* = \frac{\sum_{i=1}^3 R_{i,c}}{3}
$$

$$
\mu_c^* = \frac{\sum_{i=1}^3 R_{i,c} x^i}{\sum_{i=1}^3 R_{i,c}}
$$

$$
\sigma_c^* = \sqrt{\frac{\sum_{i=1}^3 R_{i,c}(x^i - \mu_c)^2}{\sum_{i=1}^3 R_{i,c}}}
$$

1.1.2

$$
\pi_1 = \frac{\sum_{i=1}^3 R_{i,1}}{3} = \frac{1.4}{3}
$$

$$
\pi_2 = \frac{\sum_{i=1}^3 R_{i,2}}{3} = \frac{1.6}{3}
$$

1.1.3

$$
\mu_1 = \frac{\sum_{i=1}^3 R_{i,1} x^i}{\sum_{i=1}^3 R_{i,1}} = \frac{5}{1.4}
$$

$$
\mu_2 = \frac{\sum_{i=1}^3 R_{i,2} x^i}{\sum_{i=1}^3 R_{i,2}} = \frac{26}{1.6}
$$

1.1.4

$$
\sigma_1 = \sqrt{\frac{\sum_{i=1}^3 R_{i,1}(x^i - \mu_1)^2}{\sum_{i=1}^3 R_{i,1}}} = 4.0658
$$

$$
\sigma_2 = \sqrt{\frac{\sum_{i=1}^3 R_{i,2}(x^i - \mu_2)^2}{\sum_{i=1}^3 R_{i,2}}} = 4.8412
$$

1.2. E step

1.2.1

In the **E step**, we need to **update the probability distributions** $\{R_{i,c}\}$ **of the hidden variables** $\{z^i\}$.

$$
\begin{aligned}
R_{i,c} &= p(z^i = c | x^i, \theta^{cur}) \\
&= \frac{p(z^i = c, x^i | \theta^{cur})}{\sum_{c=1}^2 p(z^i = c, x^i | \theta^{cur})} \\
&= \frac{J_{i,c}}{\sum_{c=1}^2 J_{i,c}}
\end{aligned}
$$

$$
\begin{aligned}
\text{where, } \ J_{i,c} &= p(z^i = c, x^i | \theta^{cur}) \\
&= p(z^i = c | \theta^{cur}) p(x^i | z^i = c, \theta^{cur}) \\
&= R_{i,c}^{cur} \times \frac{1}{\sqrt{2\pi}\sigma_c} \exp(-\frac{(x^i - \mu_c)^2}{2\sigma_c^2})
\end{aligned}
$$

1.2.2

After performing E step,

$$R = \begin{pmatrix} 1 & 0 \\ 0.3435 & 0.6565 \\ 0 & 1 \end{pmatrix}$$

2. HMM with tied mixtures

2.1. parameters of this HMM model

A typical HMM model has three types of parameters:

1. Start Probability: $P(X_1)$
2. Transition Probability: $P(X_{t+1}|X_t)$
3. Emission Probability: $P(O_t|X_t)$

In this case of tied-mixture HMM, we have

1. Start Probability: $S_j,\;\; j \in \{1, \cdots, M\}$
   (1) $S_j = P(X_1 = j)$
   (2) $\sum_{j=1}^{M} S_j = 1$
2. Transition Probability: $T_{ij},\;\; i, j \in \{1, \cdots, M\}$
   (1) $T_{ij} = P(X_{t+1} = j|X_t = i)$
   (2) $\sum_{j=1}^{M} T_{ij} = 1$
3. Emission Probability: $w_{jk}, \mu_k, \Sigma_k,\;\; j \in \{1, \cdots, M\},\;\; k \in \{1, \cdots, K\}$
   (1) $P(O_t|X_t = j) = \sum_{k=1}^{K} w_{jk}\mathcal{N}(O_t|\mu_k, \Sigma_k)$
   (2) $\sum_{k=1}^{K} w_{jk} = 1$

2.2. E step

Forward Algorithm:

$$\begin{aligned} \alpha_1^{jk} &= P(o_1, X_1 = j, Z_1 = k) \\ &= P(o_1|Z_1 = k)P(Z_1 = k|X_1 = j)P(X_1 = j) \\ &= \mathcal{N}(o_1|\mu_k, \Sigma_k)w_{jk}S_j \end{aligned}$$

$$\begin{aligned} A_1^{j} &= P(o_1, X_1 = j) \\ &= \sum_k P(o_1, X_1 = j, Z_1 = k) \\ &= \sum_{k=1}^{K} \alpha_1^{jk} \end{aligned}$$

$$
\begin{aligned}
\alpha_t^{jk} &= P(o_{1:t}, X_t = j, Z_t = k) \\
&= \sum_i P(o_{1:t-1}, X_{t-1} = i, o_t, X_t = j, Z_t = k) \\
&= \sum_i P(o_{1:t-1}, X_{t-1} = i) P(o_t, X_t = j, Z_t = k | o_{1:t-1}, X_{t-1} = i) \\
&= P(o_t | Z_t = k) P(Z_t = k | X_t = j) \sum_i P(o_{1:t-1}, X_{t-1} = i) P(X_t = j | X_{t-1} = i) \\
&= \mathcal{N}(o_t | \mu_k, \Sigma_k) w_{jk} \sum_{i=1}^{M} A_{t-1}^i T_{ij}
\end{aligned}
$$

$$
\begin{aligned}
A_t^j &= P(o_{1:t}, X_t = j) \\
&= \sum_k P(o_{1:t}, X_t = j, Z_t = k) \\
&= \sum_{k=1}^{K} \alpha_t^{jk}
\end{aligned}
$$

Backward Algorithm:

$$
B_T^j = 1
$$

$$
\begin{aligned}
B_t^j &= P(o_{t+1:T} | X_t = j) \\
&= \sum_i \sum_k P(o_{t+1:T}, X_{t+1} = i, Z_{t+1} = k | X_t = j) \\
&= \sum_{i=1}^{M} T_{ji} B_{t+1}^i \sum_{k=1}^{K} w_{ik} \mathcal{N}(o_{t+1} | \mu_k, \Sigma_k)
\end{aligned}
$$

Then, we update the following probability distributions.

$$\tau = P(o_{1:T}) = \sum_{i=1}^{M} A_t^i B_t^i, \ \ \forall t$$

$$\gamma_t^{jk} = P(X_t = j, Z_t = k | o_{1:T}) = \frac{\alpha_t^{jk} B_t^j}{\tau}$$

$$\Gamma_t^j = P(X_t = j | o_{1:T}) = \sum_{k=1}^{K} \gamma_t^{jk} = \frac{A_t^j B_t^j}{\tau}$$

$$\begin{aligned}
\phi_t^k &= P(Z_t = k | o_{1:T}) = \sum_{j=1}^{M} \gamma_t^{jk} = \frac{\sum_{j=1}^{M} \alpha_t^{jk} B_t^j}{\tau} \\
\xi_t^{ij} &= P(X_t = i, X_{t+1} = j | o_{1:T}) \\
&= \frac{A_t^i T_{ij} B_{t+1}^j \sum_{k=1}^{K} w_{jk} \mathcal{N}(o_{t+1} | \mu_k, \Sigma_k)}{\tau}
\end{aligned}$$

2.3. M step

In the **M step**, we update the model's parameters θ.

$$S_j = \Gamma_1^j$$

$$T_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t^{ij}}{\sum_{t=1}^{T-1} \Gamma_t^i}$$

$$w_{jk} = \frac{\sum_{t=1}^{T-1} \gamma_t^{jk}}{\sum_{t=1}^{T-1} \Gamma_t^i}$$

$$\mu_k = \frac{\sum_{t=1}^{T} \phi_t^k o_t}{\sum_{t=1}^{T} \phi_t^k}$$

$$\Sigma_k = \frac{\sum_{t=1}^{T} \phi_t^k (o_t - \mu_k)(o_t - \mu_k)^T}{\sum_{t=1}^{T} \phi_t^k}$$

3. Linear-Chain Hidden CRF for gesture recognition

3.1. Derive the formula to compute the gradient w.r.t. w

$$\frac{\partial L(w)}{\partial w} = \lambda w - \frac{1}{n}\sum_{i=1}^{n}(D_{y^i} - D)$$

$$D = \frac{\partial \log Z(X)}{\partial w} = \sum_y D_y P_y$$

$$D_y = \frac{\partial \log Z(y,X)}{\partial w}, \text{ considered as known}$$

$$P_y = P(y|X,w) = \frac{Q_y}{\sum_y Q_y}, \text{ where } Q_y = \sum_{s_t} \alpha_t(y,s_t)\beta_t(y,s_t), \ \forall t$$

3.2. Derive the formula to compute the objective

$$L(w) = \frac{\lambda}{2} w^T w - \frac{1}{n}\sum_{i=1}^{n} \log P_{y^i}$$

$$P_y = P(y|X,w) = \frac{Q_y}{\sum_y Q_y}, \text{ where } Q_y = \sum_{s_t} \alpha_t(y,s_t)\beta_t(y,s_t), \ \forall t$$

3.3. 3Classes dataset: objective value

I choose $nState = 10, \ \lambda = 0.001, \ maxIter = 150$.
The training objective value is 0.1973; the validation objective value is 0.2969.

3.4. 3Classes dataset: accuracy

Under the same setting where $nState = 10, \ \lambda = 0.001, \ maxIter = 150$,
The training accuracy is 95.95%; the validation accuracy is 93.32%.
The confusion matrix for validation data:

	Class 5	Class 6	Class 7
Class 5	173	2	3
Class 6	14	157	5
Class 7	7	4	159

3.5. 3Classes dataset: test accuracy

Under the same setting where $nState = 10, \ \lambda = 0.001, \ maxIter = 150$,
I achieved test accuracy of 93.76% on Kaggle.

3.6. 3Classes dataset: Kaggle competition

I achieved test accuracy of 93.76% on Kaggle.