CSCI 5525 Machine Learning Fall 2019

Lecture 20: Expectation and Maximization

Nov 12th 2019

Lecturer: Steven Wu *Scribe: Steven Wu*

Gaussian Mixture Model

Let us revisit the Gaussian mixture model (GMM):

- Draw a latent class Y such that $\mathbf{Pr}[Y = j] = \pi_j$
- Then draw X conditioned on Y: $X \mid Y = j \sim \mathcal{N}(\mu_j, \Sigma_j)$.

The parameter $\theta = ((\pi_1, \mu_1, \Sigma_1), \ldots, (\pi_k, \mu_k, \Sigma_k))$ and the probability density at each point x is

$$p_\theta(x) = \sum_{j=1}^{k} p_{\mu_j,\Sigma_j}(x)\ \pi_j$$

where p_{μ_j,Σ_j} denotes the multivariate Gaussian density function:

$$p_{\mu_j,\Sigma_j}(x) = \frac{1}{\sqrt{(2\pi)^d \text{det}(\Sigma_j)}} \exp\left(-\frac{1}{2}(x-\mu_j)^\intercal \Sigma_j^{-1}(x-\mu_j)\right)$$

The MLE problem becomes maximization of

$$L(\theta) = \sum_{i=1}^{n} \ln\left(\sum_{j=1}^{k} p_{\mu_j,\Sigma_j}(x_i)\ \pi_j\right) = \sum_{i=1}^{n} \ln\left[\sum_{j=1}^{k} \frac{\pi_j}{\sqrt{(2\pi)^d \text{det}(\Sigma_j)}} \exp\left(-\frac{1}{2}(x-\mu_j)^\intercal \Sigma_j^{-1}(x-\mu_j)\right)\right]$$

Unlike the MLE problem for coin flips, we cannot obtain a closed-form solution here. In fact, MLE for GMM is known to be NP-hard, but we will introduce a well-known heuristic in this lecture.

Expectation and Maximization

We will introduce the method of *expectation and maximization* (EM) for solving the MLE problem for GMM. We will introduce a set of auxiliary variables in matrix form $R \in \mathbb{R}^{n\times k} := R \in [0,1]^{n\times k} : R\mathbf{1}_k = \mathbf{1}_n$, such that each R_{ij} that defines the probability that each example x_i to the j-th Gaussian distribution. Let us define augmented likelihood as

$$L(\theta, R) = \sum_{i=1}^{n}\sum_{j=1}^{k} R_{ij} \ln\left(\sum_{j=1}^{k} \frac{p_\theta(x_i, y_i = j)}{R_{ij}}\right)$$

Note that we can write the original likelihood function as:

$$
\begin{aligned}
L(\theta) &= \sum_{i=1}^{n} \ln\left(p_\theta(x_i)\right) \\
&= \sum_{i=1}^{n} 1 \ln\left(p_\theta(x_i)\right) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{k} p_\theta(y_i = j \mid x_i) \ln\left(p_\theta(x_i)\right) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{k} p_\theta(y_i = j \mid x_i) \ln\left(\frac{p_\theta(x_i, y_i = j)}{p_\theta(y_i = j \mid x_i)}\right) \qquad \text{(Bayes rule: } p_\theta(x)\, p_\theta(y|x) = p_\theta(x, y)\text{)}
\end{aligned}
$$

This means, if we set $R_{ij} = p_\theta(y_i = j \mid x_i)$, then $L(\theta, R) = L(\theta)$.

The EM method performs the following alternating optimization over θ and R to maximize augmented likelihood function $L(\theta, R)$. the algorithm first initialize $(\pi_0)_j = 1/k$, $(\Sigma_0)_j = I$, and $(\mu_0)_j$ randomly. Then over iterations $t = 1, \ldots T$:

- **E-step**: set $(R_t)_{ij} := p_{\theta_{t-1}}(y_i = j|x_i)$. This means

$$
(R_t)_{ij} = p_{\theta_{t-1}}(y_i = j \mid x_i) = \frac{p_{\theta_{t-1}}(y_i = j, x_i)}{p_{\theta_{t-1}}(x_i)} = \frac{\pi_j p_{\mu_j, \Sigma_j}(x_i)}{\sum_{l=1}^{k} \pi_l p_{\mu_l, \Sigma_l}(x_i)}
$$

  (We omit the subscript t on the rigthmost expression.)

- **M-step**: set $\theta_t = \arg\max_{\theta \in \Theta} L(\theta; R_t)$

$$
\pi_j = \frac{\sum_{i=1}^{n} R_{ij}}{\sum_{i=1}^{n}\sum_{l=1}^{k} R_{il}} = \frac{\sum_{i=1}^{n} R_{ij}}{n} \tag{1}
$$

$$
\mu_j = \frac{\sum_{i=1}^{n} R_{ij} x_i}{\sum_{i=1}^{n} R_{ij}} = \frac{\sum_{i=1}^{n} R_{ij} x_i}{n\pi_j} \tag{2}
$$

$$
\Sigma_j = \frac{\sum_{i=1}^{n} R_{ij}(x_i - \mu_j)(x_i - \mu_j)^\intercal}{n\pi_j} \tag{3}
$$

  (We omit all the subscripts t above.)

By using first-order condition and also Lagrange duality, one can show that the choices in (1), (2) and (3) solve the problem of $\arg\max_{\theta\in\Theta} L(\theta; R_t)$. We will leave it as an exercise.

Theorem 0.1. *Let $(R_0, \theta_0) \in \mathbb{R}^{n\times k} \times \Theta$ be initialized arbitrarily. Let (R_t, θ_t) by given by EM:*

$$
(R_t)_{ij} := p_{\theta_{t-1}}(y = j \mid x_i) \qquad \theta_t := \arg\max_{\theta\in\Theta} L(\theta; R_t)
$$

Then for all t,

$$L(\theta_t; R_t) \leq \max_{R \in \mathbb{R}^{n \times k}} L(\theta_t; R) = L(\theta_t; R_{t+1}) = L(\theta_t) \leq L(\theta_{t+1}; R_{t+1}) \tag{4}$$

In particular, this implies $L(\theta_t) \leq L(\theta_{t+1})$.

Proof. Let us first prove the easy steps in (4) from left to right.

- First, $L(\theta_t; R_t) \leq \max_{R \in \mathbb{R}^{n \times k}} L(\theta_t; R)$ follows from maximization over R.
- $L(\theta_t; R_{t+1}) = L(\theta_t)$ follows from the definition of R_{t+1} and augmented likelihood.
- $L(\theta_t) \leq L(\theta_{t+1}; R_{t+1})$ follows from maximization over θ.

Now we just need to show $\max_{R \in \mathbb{R}^{n \times k}} L(\theta_t; R) = L(\theta_t; R_{t+1})$. We will rely on a useful tool from convex analysis called the *Jensen's inequality*: for any concave function $f \colon \mathbb{R}^d \to \mathbb{R}$, any $a_1, \ldots, a_k$, and any weights $\lambda_1, \ldots, \lambda_k \geq 0$ such that $\sum_{j=1}^k \lambda_j = 1$, the following inequality holds

$$\sum_{j=1}^m \lambda_j f(a_j) \leq f\left(\sum_{j=1}^m \lambda_j a_j\right)$$

Using this tool, we can bound the augmented likelihood as follows

$$\begin{aligned}
L(\theta_t, R) &= \sum_{i=1}^n \sum_{j=1}^k R_{ij} \ln \frac{p_{\theta_t}(x_i, y_i = j)}{R_{ij}} \\
&\leq \sum_{i=1}^n \ln\left(\sum_{j=1}^k R_{ij} \frac{p_{\theta_t}(x_i, y_i = j)}{R_{ij}}\right) && \text{(Jensen's inequality)} \\
&\leq \sum_{i=1}^n \ln\left(\sum_{j=1}^k p_{\theta_t}(x_i, y_i = j)\right) \\
&\leq \sum_{i=1}^n \ln\left(p_{\theta_t}(x_i)\right) = L(\theta_t) = L(\theta_t, R_{t+1})
\end{aligned}$$

This means $L(\theta_t, R) \leq L(\theta_t, R_{t+1})$ for any R, and so $\max_R L(\theta_t, R) = L(\theta_t, R_{t+1})$. □

Choosing the number k. The number k is another hyperparameter. We can follow the same approach in supervised learning, and tune it with a validation set. As we increase k, we will gradually increase the log-likelihood on the training set, but the log-likehood on the validation set will stop increasing at some point.

k-Means Clustering A related unsupervised learning method is k-means clustering. We won't go into details in this course. k-means is another alternating optimization method that aims to minimize the following k-means objective

$$\phi(\mu_1, \dots, \mu_k) = \sum_{i=1}^{n} \min_j \|x_i - \mu_j\|^2$$

The method introduces an "hard" assigment matrix $A \in \{0,1\}^{n \times k}$, and alternatively optimizes A and μ_j's:

- For each x_i, define $\mu(x_i)$ to be a closest center:

$$\|x_i - \mu(x_i)\| = \min_j \|x_i - \mu_j\|$$

- For each i, set $A_{ij} = \mathbf{1}[\mu(x_i) = \mu_j]$.

k-means can also provide initialization for the EM method.