COMS 4771: Machine Learning
Homework 5 solutions

Part 1.

E'-step: The density of a Gaussian $\mathcal{N}(\mu, \boldsymbol{I})$ at $\boldsymbol{x}$ is given by $(2\pi)^{-d/2}\exp(-\frac{1}{2}\|\boldsymbol{x}-\mu\|_2^2)$. Using this, we have

$$\begin{aligned}\Pr_{\boldsymbol{\theta}}[Y = j \mid \boldsymbol{X} = \boldsymbol{x}^{(i)}] &= \frac{\Pr_{\boldsymbol{\theta}}[\boldsymbol{X} = \boldsymbol{x}^{(i)} \mid Y = j] \cdot \Pr_{\boldsymbol{\theta}}[Y = j]}{\Pr_{\boldsymbol{\theta}}[\boldsymbol{X} = \boldsymbol{x}^{(i)}]} \\ &= \frac{\pi_j \cdot (2\pi)^{-d/2}\exp(-\frac{1}{2}\|\boldsymbol{x}^{(i)}-\mu_j\|_2^2)}{\Pr_{\boldsymbol{\theta}}[\boldsymbol{X} = \boldsymbol{x}^{(i)}]}\end{aligned}$$

Since $\Pr_{\boldsymbol{\theta}}[\boldsymbol{X} = \boldsymbol{x}^{(i)}]$ is independent of j, to find the most likely value for Y, we can ignore the denominator of the ratio and simply maximize $\pi_j \cdot (2\pi)^{-d/2}\exp(-\frac{1}{2}\|\boldsymbol{x}^{(i)}-\mu_j\|_2^2)$. Maximizing this is equivalent to maximizing its logarithm, and dropping terms which don't depend on j, we end up with the following:

$$j^{(i)} = \arg\max_{j\in[k]} \ln(\pi_j) - \frac{1}{2}\|\boldsymbol{x}^{(i)}-\mu_j\|_2^2 = \arg\min_{j\in[k]} \|\boldsymbol{x}^{(i)}-\mu_j\|_2^2 - 2\ln(\pi_j).$$

In other words, $j^{(i)}$ is the index of the center μ_j that's closest to $\boldsymbol{x}^{(i)}$, except that we add a "handicap" of $-2\ln(\pi_j)$ to the squared distance of $\boldsymbol{x}^{(i)}$ to center j. Finally, we set

$$q_j^{(i)} = \begin{cases} 1 & \text{if } j = j^{(i)} \\ 0 & \text{otherwise.} \end{cases}$$

M-step. Since the M-step is the same as in the soft EM algorithm, the derived formulae are also the same. In this case, the formulae become a bit simpler since the covariance matrices are fixed to $\boldsymbol{I}$.

Let $S_j = \{x^{(i)} \in S \mid q_j^{(i)} = 1\}$ be the set of training points assigned to center j. Then we have the following updated parameters in the M-step:

$$\pi_j = \frac{1}{n}\sum_{i=1}^n q_j^{(i)} = \frac{|S_j|}{n},$$

and

$$\mu_j = \frac{1}{|S_j|}\sum_{i=1}^n \boldsymbol{x}^{(i)} q_j^{(i)} = \frac{1}{|S_j|}\sum_{\boldsymbol{x}\in S_j} \boldsymbol{x}.$$

Part 2. The hard EM algorithm derived above is very similar to the Lloyd's algorithm for k-means. The only difference is in Lloyd's algorithm, we assign each point to the closest center, whereas in the hard EM algorithm above we have an additional $-2\ln(\pi_j)$ added to the squared distance to center j. The new centers are computed exactly the same way in both algorithms, while the hard EM algorithm also needs to compute the mixing weights π_j.

Lloyd's algorithm can be derived as a hard EM algorithm where not only the covariance matrices but also the mixing weights are assumed to be fixed or known in advance, and the only parameters to be estimated are the centers μ_j.