COMS 4771: Machine Learning
Homework 5, due April 28.

Note: This homework is optional and will only count towards extra credit.

In class, the form of the Expectation-Maximization (EM) algorithm we studied is the most common one, and is sometimes called "soft EM" since the E-step computes soft assignments of the hidden variables. There is a variant called "hard EM" where the E-step is replaced by a hard assignemnt of the hidden variables.

Specifically, suppose the observed data point is denoted by $\boldsymbol{X} \in \mathcal{X}$, the hidden variable is denoted by $Y \in [k] = \{1, 2, \ldots, k\}$, and their joint distribution is obtained from a parameterized model with parameters denoted by $\boldsymbol{\theta}$. Let $\Pr_{\boldsymbol{\theta}}[\cdot]$ denote probabilities in the distribution with parameters $\boldsymbol{\theta}$.

The (soft) EM algorithm as we studied it in class repeats the following two steps until convergence, starting from an arbitrary initialization of the parameters $\boldsymbol{\theta}$:

- **E-step:** For each training example $\boldsymbol{x}^{(i)}$ and for all $j \in [k]$, compute

$$q_j^{(i)} = \Pr_{\boldsymbol{\theta}}[Y = j \mid \boldsymbol{X} = \boldsymbol{x}^{(i)}]$$

- **M-step:** Update $\boldsymbol{\theta}$ as follows:

$$\boldsymbol{\theta} \leftarrow \arg\max_{\boldsymbol{\theta}'} \sum_{i=1}^{n} \sum_{j=1}^{k} q_j^{(i)} \cdot \ln \Pr_{\boldsymbol{\theta}'}[\boldsymbol{X} = \boldsymbol{x}^{(i)},\ Y = j]$$

In hard EM, the E-step above is replaced by a hard assignment of the hidden variable to the most likely value for the given data point under the current distribution (i.e. with parameters $\boldsymbol{\theta}$). We will call this the E'-step. The M-step stays the same.

The hard EM algorithm repeats the following two steps until convergence, starting from an arbitrary initialization of the parameters $\boldsymbol{\theta}$:

- **E'-step:** For every training example $\boldsymbol{x}^{(i)}$, compute

$$j^{(i)} = \arg\max_{j \in [k]} \Pr_{\boldsymbol{\theta}}[Y = j \mid \boldsymbol{X} = \boldsymbol{x}^{(i)}],$$

  breaking ties arbitrarily, and set

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

- **M-step:** The same as the M-step in the soft EM algorithm.

Now consider implementing the hard EM algorithm for the following specific scenario. We have a Gaussian mixture model where all the k Gaussians have the same covariance matrix, $\boldsymbol{I}$, the identity matrix. Thus, in particular, the parameters of the model are

$$\boldsymbol{\theta} = (\pi_1, \mu_1, \pi_2, \mu_2, \ldots, \pi_k, \mu_k),$$

where π_j are the mixing weights, and μ_j are the centers of the Gaussians. Samples are generated from this mixture model exactly as in class (viz. as described on slide 4 of the lecture.) The setting is a bit simpler in that covariance matrices are already known (i.e. $\boldsymbol{I}$).

Part 1 of assignment (weightage: 4%). For the mixture of Gaussians setting described above, work out the precise implementation of the E'- and M-steps.

Part 2 of assignment (weightage: 1%). The hard EM algorithm you derived above should be almost (but not exactly) the same as another algorithm we have already studied in class. Which algorithm? What is the difference between the two algorithms?