ECE 283: Homework 3

Topics: Unsupervised Learning, part 1 (K Means, Gaussian mixtures and the EM algorithm)
Reading: Handout 5 (Sections 1-3 are needed for doing the homework; the rest is theoretical background that is highly recommended reading); Bishop, Chapter 9.

Experiments with 2D data: For this, let us use the same data statistics as in HW1 and HW2, except that we discard the class labels. Thus, we have four Gaussian components, as follows:
Component A: $\pi_A = \frac{3}{8}$, $\mathbf{m}_A = (0,0)^T$, $\mathbf{C}_A$ with eigenvalue, eigenvector pairs:
$\lambda_1 = 4$, $\mathbf{u}_1 = (\cos\theta, \sin\theta)^T$, $\lambda_2 = 1$, $\mathbf{u}_2 = (-\sin\theta, \cos\theta)^T$, with $\theta = 0$.
Component B: $\pi_B = \frac{1}{8}$, $\mathbf{m}_B = (6,4)^T$, $\mathbf{C}_B$ with eigenvalue, eigenvector pairs:
$\lambda_1 = 1$, $\mathbf{u}_1 = (\cos\theta, \sin\theta)^T$, $\lambda_2 = 4$, $\mathbf{u}_2 = (-\sin\theta, \cos\theta)^T$, with $\theta = \frac{\pi}{3}$.
Component C: $\pi_C = \frac{1}{3}$, $\mathbf{m}_C = (2,3)^T$, $\mathbf{C}_C$ with eigenvalue, eigenvector pairs:
$\lambda_1 = 1$, $\mathbf{u}_1 = (\cos\theta, \sin\theta)^T$, $\lambda_2 = 2$, $\mathbf{u}_2 = (-\sin\theta, \cos\theta)^T$, with $\theta = \frac{\pi}{4}$.
Component D: $\pi_D = \frac{1}{6}$, $\mathbf{m}_B = (2,-2)^T$, $\mathbf{C}_D$ with eigenvalue, eigenvector pairs:
$\lambda_1 = 4$, $\mathbf{u}_1 = (\cos\theta, \sin\theta)^T$, $\lambda_2 = 1$, $\mathbf{u}_2 = (-\sin\theta, \cos\theta)^T$, with $\theta = \frac{\pi}{6}$.
Remark: When referring to components below, we map (A,B,C,D) to (1,2,3,4).

1) Generate $N = 200$ data samples from the preceding model, saving both the data point $\mathbf{x}_i$ and $\mathbf{z}_i \in \{0,1\}^4$, the one-hot encoding of which component the data point belongs to. Implement the K-means algorithm with different values of $K = 2, 3, 4, 5, 6$.

(a) For each K, start with several different random initializations, and choose the run that leads to the smallest mean squared error. Let $\{\mathbf{m}_k, k = 1, ..., K\}$ denote the cluster centers, and for each data point, compute $\mathbf{a}_i \in \{0,1\}^K$, the one-hot encoding of which cluster the data point is assigned to.

(b) Plot the empirical probabilities $P[a_i[k] = 1 | z_i[l] = 1]$, $l = 1, 2, 3, 4$, $k = 1, ..., K$ in a $4 \times K$ table, indicating how the "ground truth" components map to the clusters you learn.

(c) Plot the distortion $\Delta^2(K)$ as a function of K. Do you see a way to predict the "right value" of K?

2) For the same data samples, repeat 1) with a single K-means++ initialization. Comment on whether you see any differences in the results.

3) Using the results of 1) or 2) as a starting point, implement the EM algorithm to estimate the mean and covariance for a Gaussian mixture model with the "right" value of K as determined from either 1) or 2). Plot the average values of $p(k|i), k = 1, ..., K$ for data points drawn from each ground truth component, $l = 1, 2, 3, 4$, in a $4 \times K$ table. Does allowing learnt elliptical covariances provide a better fit than the spherical covariance implicitly assumed in K means?

Experiments with data in higher dimensions: Let us now see what happens when we increase the number of dimensions to d (d to be played with, but the nominal value is $d = 30$), while keeping the "effective dimension" smaller than d. We will do this by drawing a small number of random vectors of dimension d, and then taking random linear combinations of them to generate the data. We will also add a bit of white noise so the vectors we generate do have energy over all d dimensions.

4) Write the following program to generate a random vector $\mathbf{u}$ in d dimensions as follows:

The components of $\mathbf{u}$ are i.i.d., with

$$P[u[i] = 0] = 2/3, \quad P[u[i] = +1] = 1/6, \quad P[u[i] = -1] = 1/6$$

Let $\{\mathbf{u}_j, j = 1, ..., 6\}$ be i.i.d. draws from your program in 4). Draw them once, and then fix them. Check that the vectors are quasi-orthogonal. If the normalized correlation between two of them is too large, then purge one of them and draw another vector. We use these vectors to generate data samples coming from a Gaussian mixture distribution, as follows.

5) Write a program to use the fixed vectors from 4) to generate d-dimensional data samples for a Gaussian mixture distribution with 3 *equiprobable* components, as follows. In order to generate any given data point $\mathbf{X}$, we will use i.i.d. draws from a standard Gaussian ($N(0, 1)$) distribution that we will denote $\{V_m\}$, and we will also draw a "noise vector" $\mathbf{N} \sim N(0, \sigma^2 \mathbf{I}_d)$ (default value $\sigma^2 = 0.01$). A sample from each of the three components can now be described as follows (remember, a given data sample belongs to exactly one of these components):
Component 1: Generate $\mathbf{X} = 2\mathbf{u}_1 + 0.5V_1\mathbf{u}_2 + V_2\mathbf{u}_3 + \mathbf{N}$.
Component 2: Generate $\mathbf{X} = 1.5\mathbf{u}_4 + V_1\mathbf{u}_5 + 0.8V_2\mathbf{u}_6 + \mathbf{N}$.
Component 3: Generate $\mathbf{X} = \mathbf{u}_6 + V_1(\mathbf{u}_1 - \mathbf{u}_2) + 0.7V_2\mathbf{u}_5 + \mathbf{N}$.
Note that the vectors $\{\mathbf{u}_j\}$ stay the same across data samples, but the random numbers V_1 and V_2, and the noise vector $\mathbf{N}$ are drawn afresh for each sample.

6) Generate $N =$?? (to be determined) data samples from the preceding model, saving both the data point $\mathbf{x}_i$ and $\mathbf{z}_i \in \{0, 1\}^3$, the one-hot encoding of which component the data point belongs to. Implement the K-means algorithm with different values of $K = 2, 3, 4, 5$. For each K, either use K-means++, or start with several different random initializations, and choose the run that leads to the smallest distortion Δ^2. Let $\{\mathbf{m}_k, k = 1, ..., K\}$ denote the cluster centers, and for each data point, compute $\mathbf{a}_i \in \{0, 1\}^K$, the one-hot encoding of which cluster the data point is assigned to. Plot the empirical probabilities $P[a_i[k] = 1 | z_i[l] = 1]$, $l = 1, 2, 3$, $k = 1, ..., K$ in a $3 \times K$ table, indicating how the "ground truth" components map to the clusters you learn.

7) Try to provide geometric insight as to how the cluster centers found by K-means relate to the vectors $\{\mathbf{u}_j\}$ in the model.
Remark: For example, you can plot the normalized inner products between the cluster centers and the vectors $\{\mathbf{u}_j\}$.

8) Run the EM algorithm with several different values of K. Comment on whether and how the eigenvectors of the covariance matrices you find relate to the parameters of the Gaussian mixture model.

9) **Optional bonus problem:** Play with deterministic annealing for this dataset, and report on your findings.