Problem 1 (20 Points): Lloyd's Method

Given a dataset with seven data points $\{x_1, \cdots, x_7\}$ and the distances between all pairs of data points are in the following table.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
x_1	0	5	3	1	6	2	3
x_2	5	0	4	6	1	7	8
x_3	3	4	0	4	3	5	6
x_4	1	6	4	0	7	1	2
x_5	6	1	3	7	0	8	9
x_6	2	7	5	1	8	0	1
x_7	3	8	6	2	9	1	0

Assume the number of clusters $k = 2$, and the cluster centers are initialized to be x_3 and x_6.

1. **5 Points.** What's the two clusters formed at the end of the first iteration of Lloyd's algorithm?
2. **5 Points.** What's the two clusters formed at the end of the second iteration of Lloyd's algorithm?
3. **10 Points.** What's the two clusters formed when the Lloyd's algorithm converges?

Problem 2 (20 Points): Guassian Mixture Model (GMM): Latent Variable View

Consider a GMM in which the marginal distribution $p(\mathbf{z})$ for the latent variable $\mathbf{z}$ is given by

$$p(\mathbf{z}) = \prod_{k=1}^{K} \pi_k^{z_k},$$

where $\sum_{k=1}^{K} \pi_k = 1$; $\mathbf{z} = [z_1, z_2, \cdots, z_K]$ and z_k satisfies $z_k \in \{0, 1\}$ and $\sum_{k=1}^{K} z_k = 1$.
Moreover, the conditional distribution $p(\mathbf{x}|\mathbf{z})$ for the observed variable $\mathbf{x}$ is given by:

$$p(\mathbf{x}|\mathbf{z}) = \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)^{z_k}.$$

Prove that $p(\mathbf{x})$, obtaining by summing $p(\mathbf{z})p(\mathbf{x}|\mathbf{z})$ over all possible values of $\mathbf{z}$, is a GMM. That is,

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{z})p(\mathbf{x}|\mathbf{z}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k).$$

Problem 3 (20 Points): Mixture Models: Mixtures of Bernoulli Distributions

Consider a set of D binary variables x_i, where $i = 1, \cdots, D$, each of which is governed by a Bernoulli distribution with parameter μ_i, so that

$$p(\mathbf{x}|\boldsymbol{\mu}) = \prod_{i=1}^{D} \mu_i^{x_i}(1 - \mu_i)^{(1-x_i)},$$

where $\mathbf{x} = [x_1; x_2; \cdots, x_D]$ and $\boldsymbol{\mu} = [\mu_1, \mu_2; \cdots, \mu_D]$.

Now let us consider a finite mixture of these distributions given by

$$p(\mathbf{x}|M, \boldsymbol{\pi}) = \sum_{k=1}^{K} \pi_k p(\mathbf{x}|\boldsymbol{\mu}_k),$$

where $M = \{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_k\}$, $\boldsymbol{\pi} = \{\pi_1, \pi_2, \cdots, \pi_K\}$, and

$$p(\mathbf{x}|\boldsymbol{\mu}_k) = \prod_{i=1}^{D} \mu_{ki}^{x_{ki}} (1 - \mu_{ki})^{(1-x_{ki})}.$$

Show that:

1. **10 Points.** The mean of this mixture distribution is given by

$$\mathbb{E}[\mathbf{x}] = \sum_{k=1}^{K} \pi_k \boldsymbol{\mu}_k.$$

2. **10 Points.** The covariance of this mixture distribution is given by

$$\text{cov}[\mathbf{x}] = \sum_{k=1}^{K} \pi_k \left(\boldsymbol{\Sigma}_k + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T\right) - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{x}]^T,$$

where $\boldsymbol{\Sigma_k} = \text{dig}(\mu_{ki}(1 - \mu_{ki}))$.

Problem 4 (15 Points): Generating GMMs

In this problem, you will write code to generate a mixture of 3 Gaussians satisfying the following requirements, respectively. Please specify the mean vector and covariance matrix of each Gaussian in your answer.

1. **3 Points.** Draw a data set where a mixture of 3 spherical Gaussians (where the covariance matrix is the identity matrix times some positive scalar) can model the data well, but K-means cannot.
2. **4 Points.** Draw a data set where a mixture of 3 diagonal Gaussians (where the covariance matrix can have non-zero values on the diagonal, and zeros elsewhere) can model the data well, but K-means and a mixture of spherical Gaussians cannot.
3. **5 Points.** Draw a data set where a mixture of 3 Gaussians with unrestricted covariance matrices can model the data well, but K-means and a mixture of diagonal Gaussians cannot.

Problem 4 (25 Points): Implementing K-Means Clustering

In this problem, you will implement the K-means clustering algorithm on a synthetic data set. The default programming language is MATLAB, but you can choose any language that you are comfortable. There is code and data for this problem (see attached). The steps are as follows:

1. **2 Points.** Run `load 'X.dat';` to load the data file for clustering.
2. **15 Points.** Implement the `[clusters, centers] = k_means(X, k)` function. This function takes the $m \times n$ data matrix $\mathbf{X}$ and the number of clusters k as input. It should output a m element vector, `clusters`, which indicates which of the clusters each data point belongs to, and a $k \times n$ matrix, `centers`, which contains the centroids of each cluster. Run the algorithm on the data with $k = 3$ and $k = 4$.
3. **8 Points.** Plot the cluster assignments and centroids for each iteration of the algorithm using the `draw_clusters(X, clusters, centroids)` function. For each k, be sure to run the algorithm several times using different initial centroids.