STA 243: Homework 3

- Homework due in Canvas: 05/29/2024 at 9:00 PM. Please follow the instructions provided in Canvas about homeworks, carefully.

1. Please download the MNIST handwritten digit dataset (training set images and training set labels) from `http://yann.lecun.com/exdb/mnist/`

   It contains 28×28-pixel images for the hand-written digits {0,1,...,9} by storing each pixel value ranging between 0 and 255, and their corresponding true labels.

   Load the data into Python. Preprocess the data by compressing each image to 1/4 of the original size in the following way: Divide each 28×28 image into 2×2 non-overlapping blocks. Calculate the mean pixel value of each 2×2 block, and create a new 14×14 image. This preprocessing step will drastically help your computation. We will be clustering the digits $\{0, 1, 2, 3, 4\}$ in this homework. For visualization purpose, we view each data sample as 14×14 matrix. For using in an algorithm, treat each sample as a vector - you just simply stack each column of the 14×14 matrix into a 196 dimensional vector.

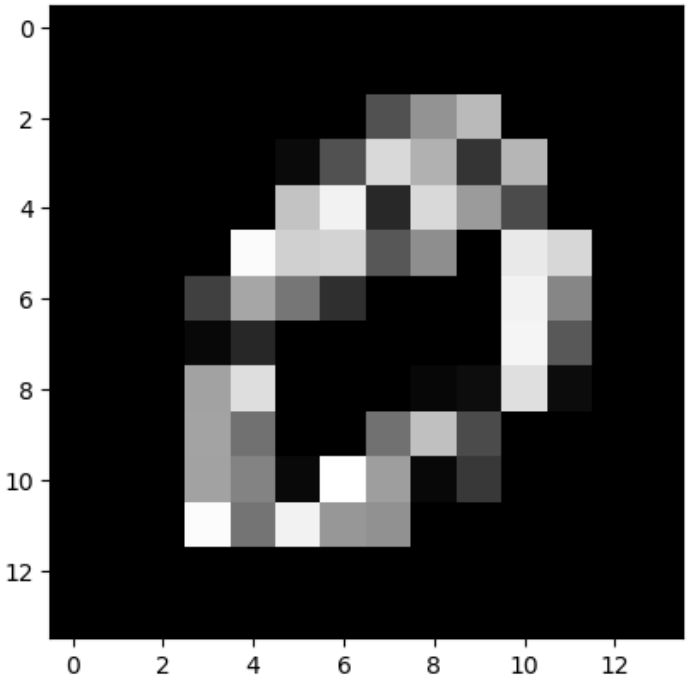

Figure 1: Sample of Reshaped Figure

2. (**20 Points**) Closely following the derivation in class, we now derive the EM algorithm for Gaussian mixture models. More specifically, we will use 2 models, the "mixture of spherical Gaussians" and the "mixture of diagonal Gaussians". By the end of this question, you should have derived two EM algorithms, one for each model. Below, the following denotes the meaning of each symbol:

   - each $\boldsymbol{\mu}_j$ is a d-dimensional vector representing the cluster center for cluster j
   - each $\boldsymbol{\Sigma}_j$ is a $d \times d$ matrix representing the covariance matrix for cluster j
   - $\sum_{j=1}^{k} \pi_j = 1$ and $\forall j, \pi_j \geq 0$ where all the π_j are the mixing coefficients.
   - Z_i represents the missing variable associated with $\boldsymbol{x}_i$ for $i \in \{1, \ldots, n\}$. It takes integer values $1, \ldots, k$.

   The parameters of the model to be estimated are $\boldsymbol{\theta} = (\{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, \pi_j\}_{j=1}^{k})$. To be more explicit, we will the notation $p(\boldsymbol{x}_i; \{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, \pi_j\}_{j=1}^{k})$ to denote the probability density function $p_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$. That is, $p(\boldsymbol{x}_i; \{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, \pi_j\}_{j=1}^{k}) = p_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$. The likelihood of the data for k clusters is:

$$
\begin{aligned}
L(\boldsymbol{\theta}) &= \prod_{i=1}^{n} p(\boldsymbol{x}_i; \{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, \pi_j\}_{j=1}^{k}) \qquad (1) \\
&= \prod_{i=1}^{n} \sum_{j=1}^{k} p(\boldsymbol{x}_i | Z_i = j; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) p(Z_i = j) \\
&= \prod_{i=1}^{n} \sum_{j=1}^{k} \frac{\pi_j}{(2\pi)^{d/2} |\boldsymbol{\Sigma}_j|^{1/2}} \exp\left(-\frac{1}{2} (\boldsymbol{x}_i - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_j) \right)
\end{aligned}
$$

   In a **mixture of diagonal Gaussians** model, $\boldsymbol{\Sigma}_j = \text{diag}(\sigma_{j_1}^2, \ldots, \sigma_{j_d}^2)$ is a diagonal matrix for all $j = 1, \ldots, k$. So, we only have d parameters to estimate for the covariance matrix of each cluster.

   In a **mixture of spherical Gaussians** model, $\boldsymbol{\Sigma}_j = \sigma_j^2 \boldsymbol{I}_d$ for all $j = 1, \ldots, k$. Here $\sigma_j > 0$ is a scalar and $\boldsymbol{I}_d$ is the $d \times d$ identity matrix. So we only have 1 parameter to estimate for the covariance matrix of each cluster.

   (i) First, we do some preliminary work that will be useful later on.

   - **Part a:** Write down the marginal distribution of the Z_i (Hint: What is the probability $p(Z_i = j)$).

     $$p(Z_i = j) = \pi_j$$

   - **Part b:** Calculate $p(Z_i = j | \boldsymbol{x}_i)$. (Hint: Bayes Rule)

$$
\begin{aligned}
p(Z_i = j|\boldsymbol{x}_i) &= \frac{p(Z_i = j, \boldsymbol{x}_i)}{p(\boldsymbol{x}_i)} \\
&= \frac{p(Z_i = j)p(\boldsymbol{x}_i|Z_i = j)}{p(\boldsymbol{x}_i)} \\
&= \frac{p\left(\mathbf{x}_i \mid Z_i = j\right) p\left(Z_i = j\right)}{\sum_{j'=1}^{k} p\left(\mathbf{x}_i \mid Z_i = j'\right) p\left(Z_i = j'\right)} \\
&= \frac{\pi_j \left|\boldsymbol{\Sigma}_j\right|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\left(\mathbf{x}_i - \boldsymbol{\mu}_j\right)^{\top} \boldsymbol{\Sigma}_j^{-1}\left(\mathbf{x}_i - \boldsymbol{\mu}_j\right)\right)}{\sum_{j'=1}^{k} \pi_{j'} \left|\boldsymbol{\Sigma}_{j'}\right|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\left(\mathbf{x}_i - \boldsymbol{\mu}_{j'}\right)^{\top} \boldsymbol{\Sigma}_{j'}^{-1}\left(\mathbf{x}_i - \boldsymbol{\mu}_{j'}\right)\right)}
\end{aligned}
$$

(ii). We now derive the E- and M- steps. We denote $\boldsymbol{\theta} := \{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, \pi_j\}_{j=1}^k$. From Equation (1), we can write down the log-likelihood and derive a lower bound:

$$
\begin{aligned}
\ell(\boldsymbol{\theta}) &= \sum_{i=1}^{n} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i) \\
&= \sum_{i=1}^{n} \log \left[\sum_{j=1}^{k} p_{\boldsymbol{\theta}}(\boldsymbol{x}_i, Z_i = j)\right] \\
&= \sum_{i=1}^{n} \log \left[\sum_{j=1}^{k} \mathbf{F}_{ij} \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_i, Z_i = j)}{\mathbf{F}_{ij}}\right],
\end{aligned}
$$

where $F_{ij} > 0$ for all i, j and satisfies

$$
\sum_{j=1}^{k} F_{ij} = 1 \quad for\ i = 1, \ldots, n.
$$

Note that if $i = 1$, $\mathbf{F}_{1j} \in \mathbb{R}^k$ corresponds to the the distributions that we pick in the slides, denoted as q_1. Similarly for all i from 1 to n.

- **Part c:** Prove the following lower bound of the log-likelihood function:

$$
\ell(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log \left[\sum_{j=1}^{k} \mathbf{F}_{ij} \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_i, Z_i = j)}{\mathbf{F}_{ij}}\right] \geq \sum_{i=1}^{n} \sum_{j=1}^{k} \mathbf{F}_{ij} \log \left[\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_i, Z_i = j)}{\mathbf{F}_{ij}}\right]. \tag{2}
$$

  Hint: you can use the Jensen's inequality $\log \mathbb{E}X \geq \mathbb{E} \log X$ (You are not required to prove the Jensen's inequality).

  Note for convex real function, we have Jensen's inequality For a real convex function φ, numbers $x_1, x_2, \ldots, x_k$ in its domain, and positive weights a_j, Jensen's inequality can be stated as:

$$
\varphi\left(\frac{\sum a_j x_j}{\sum a_j}\right) \leq \frac{\sum a_j \varphi\left(x_j\right)}{\sum a_j}
$$

  Note that log function is convex real function, then apply Jensen's inequality with $\varphi(x) = \log(x)$, $a_j = F_{ij}$ and $x_j = \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_i, Z_i = j)}{\mathbf{F}_{ij}}$, $\sum_{j=1}^{k} F_{ij} = 1$, we have

$$\log\left[\sum_{j=1}^{k}\mathbf{F}_{ij}\frac{p_{\boldsymbol{\theta}}\left(\boldsymbol{x}_i, Z_i=j\right)}{\mathbf{F}_{ij}}\right] \geq \sum_{j=1}^{k}\mathbf{F}_{ij}\log\left[\frac{p_{\boldsymbol{\theta}}\left(\boldsymbol{x}_i, Z_i=j\right)}{\mathbf{F}_{ij}}\right]$$

This holds for every i. so that we can sum all i up:

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{n}\log\left[\sum_{j=1}^{k}\mathbf{F}_{ij}\frac{p_{\boldsymbol{\theta}}\left(\boldsymbol{x}_i, Z_i=j\right)}{\mathbf{F}_{ij}}\right] \geq \sum_{i=1}^{n}\sum_{j=1}^{k}\mathbf{F}_{ij}\log\left[\frac{p_{\boldsymbol{\theta}}\left(\boldsymbol{x}_i, Z_i=j\right)}{\mathbf{F}_{ij}}\right]$$

- **Part d:** (E-Step) We define

$$Q(\mathbf{F}, \boldsymbol{\theta}) := \sum_{i=1}^{n}\sum_{j=1}^{k}\mathbf{F}_{ij}\log\left[\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_i, Z_i=j)}{\mathbf{F}_{ij}}\right] \tag{3}$$

to be the lower bound function of $\ell(\boldsymbol{\theta})$. Let $\boldsymbol{\theta}'$ be a fixed value of $\boldsymbol{\theta}$ (e.g., $\boldsymbol{\theta}'$ could be the parameter value of the current iteration). Recall from Part c that we designed $Q(\mathbf{F}, \boldsymbol{\theta})$ to be a lower bound for $\ell(\boldsymbol{\theta})$. We want to make this lower bound as tight as possible. Prove that $\ell(\boldsymbol{\theta}') = Q(\mathbf{F}, \boldsymbol{\theta}')$ when

$$\mathbf{F}_{ij} = p_{\boldsymbol{\theta}'}(Z_i = j|\boldsymbol{x}_i). \tag{4}$$

From **Part c**, we have

$$Q(F, \theta') \leq \sum_{i=1}^{n}\log\sum_{j=1}^{k}F_{ij}\frac{p_{\theta'}(x_i, Z_i=j)}{F_{ij}}$$

And according to Jensen's inequality, the equation holds if and only if

$$p_{\theta'}(x_i, Z_i=j) \propto F_{ij}$$

Note that we have constrain of $\sum_{j=1}^{k}F_{ij}=1$, Therefore:

$$F_{ij} = \frac{p_{\theta'}(x_i, Z_i=j)}{\sum_{t=1}^{k}p_{\theta'}(x_i, Z_i=t)} = \frac{p_{\theta'}(x_i, Z_i=j)}{p_{\theta'}(x_i)} = p_{\theta'}(Z_i=j|x_i)$$

Q.E.D.

(iii). Once the E-step is derived, we now derive the M-step. First, we plug

$$\mathbf{F}_{ij} = p_{\boldsymbol{\theta}^{(t)}}(Z_i = j|\boldsymbol{x}_i)$$

into Equation 3 and define the lower bound function at $\boldsymbol{\theta}^{(t)}$ to be

$$Q(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}) := \sum_{i=1}^{n}\sum_{j=1}^{k} p_{\boldsymbol{\theta}^{(t)}}(Z_i = j|\boldsymbol{x}_i) \log\left[\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_i, Z_i = j)}{p_{\boldsymbol{\theta}^{(t)}}(Z_i = j|\boldsymbol{x}_i)}\right]. \tag{5}$$

- **Part e** (M-step for mixture of spherical Gaussians) In the M-step, we aim to find $\boldsymbol{\theta}^{(t+1)}$ that maximize the lower bound function $Q(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta})$. Under the **mixture of spherical Gaussians** model, derive the M-step updating equations for $\boldsymbol{\mu}_j^{(t+1)}, \boldsymbol{\Sigma}_j^{(t+1)}$ and $\pi_j^{(t+1)}$. We have:

$$\begin{aligned}
Q(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}) &= \sum_{i=1}^{n}\sum_{j=1}^{k} p_{\boldsymbol{\theta}^{(t)}}(Z_i = j|\boldsymbol{x}_i) \log\left[\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_i, Z_i = j)}{p_{\boldsymbol{\theta}^{(t)}}(Z_i = j|\boldsymbol{x}_i)}\right] \\
&= \sum_{i=1}^{n}\sum_{j=1}^{k} \mathbf{F}_{ij}^{(t)} \log\frac{p\left(\boldsymbol{x}_i, Z_i = j\right)}{\mathbf{F}_{ij}^{(t)}} \\
&= \sum_{i=1}^{n}\sum_{j=1}^{k} \mathbf{F}_{ij}^{(t)} \log\frac{\frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_j|^{1/2}} \exp\left(-\frac{1}{2}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)^\top \boldsymbol{\Sigma}_j^{-1}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)\right) \cdot \pi_j}{\mathbf{F}_{ij}^{(t)}}
\end{aligned}$$

If we take the derivative with respect to $\boldsymbol{\mu}_j$, we find: First, simplify the logarithmic term:

$$\log\frac{\frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_j|^{1/2}} \exp\left(-\frac{1}{2}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)^\top \left(\boldsymbol{\Sigma}_j\right)^{-1}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)\right) \cdot \pi_j}{\mathbf{F}_{ij}^{(t)}}$$

equals

$$\log\left(\frac{1}{(2\pi)^{d/2}\left|\boldsymbol{\Sigma}_j\right|^{1/2}}\right) + \log\left(\exp\left(-\frac{1}{2}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)^\top \left(\boldsymbol{\Sigma}_j\right)^{-1}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)\right)\right) + \log(\pi_j) - \log(\mathbf{F}_{ij}^{(t)})$$

which simplifies to

$$-\frac{d}{2}\log(2\pi) - \frac{1}{2}\log\left|\boldsymbol{\Sigma}_j\right| - \frac{1}{2}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)^\top \left(\boldsymbol{\Sigma}_j\right)^{-1}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right) + \log(\pi_j) - \log(\mathbf{F}_{ij}^{(t)})$$

Since $\boldsymbol{\mu}_j$ appears only in the third term, we only need to differentiate this term:

$$\nabla_{\boldsymbol{\mu}_j}\left(-\frac{1}{2}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)^\top \left(\boldsymbol{\Sigma}_j\right)^{-1}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)\right)$$

Recall that for a vector $\boldsymbol{a}$ and a matrix $\mathbf{A}$,

$$\nabla_{\boldsymbol{a}}\left(\boldsymbol{a}^\top \mathbf{A}\boldsymbol{a}\right) = 2\mathbf{A}\boldsymbol{a}$$

Thus,

$$\nabla_{\boldsymbol{\mu}_j}\left(-\frac{1}{2}\left(\boldsymbol{x}_i-\boldsymbol{\mu}_j\right)^\top\left(\boldsymbol{\Sigma}_j\right)^{-1}\left(\boldsymbol{x}_i-\boldsymbol{\mu}_j\right)\right)=\left(\boldsymbol{\Sigma}_j\right)^{-1}\left(\boldsymbol{x}_i-\boldsymbol{\mu}_j\right)$$

Plugging this back into the original summation, we get:

$$\sum_{i=1}^{n}\mathbf{F}_{ij}^{(t)}\left(\left(\boldsymbol{\Sigma}_j\right)^{-1}\boldsymbol{x}_i-\left(\boldsymbol{\Sigma}_j\right)^{-1}\boldsymbol{\mu}_j\right)$$

Therefore, the detailed derivation is as follows:

$$\begin{aligned}
&\nabla_{\boldsymbol{\mu}_j}\sum_{i=1}^{n}\sum_{j=1}^{k}\mathbf{F}_{ij}^{(t)}\log\frac{\frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_j|^{1/2}}\exp\left(-\frac{1}{2}\left(\boldsymbol{x}_i-\boldsymbol{\mu}_j\right)^\top\left(\boldsymbol{\Sigma}_j\right)^{-1}\left(\boldsymbol{x}_i-\boldsymbol{\mu}_j\right)\right)\cdot\pi_j}{\mathbf{F}_{ij}^{(t)}}\\
=&\sum_{i=1}^{n}\sum_{j=1}^{k}\mathbf{F}_{ij}^{(t)}\left(-\frac{1}{2}\left(\boldsymbol{x}_i-\boldsymbol{\mu}_j\right)^\top\left(\boldsymbol{\Sigma}_j\right)^{-1}\left(\boldsymbol{x}_i-\boldsymbol{\mu}_j\right)\right)\\
=&\sum_{i=1}^{n}\mathbf{F}_{ij}^{(t)}\left(\left(\boldsymbol{\Sigma}_j\right)^{-1}\boldsymbol{x}_i-\left(\boldsymbol{\Sigma}_j\right)^{-1}\boldsymbol{\mu}_j\right)
\end{aligned}$$

Setting this to zero and solving for $\boldsymbol{\mu}_j$ therefore yields the update rule

$$\boldsymbol{\mu}_j^{(t+1)}:=\frac{\sum_{i=1}^{n}\mathbf{F}_{ij}^{(t)}\boldsymbol{x}_i}{\sum_{i=1}^{n}\mathbf{F}_{ij}^{(t)}}$$

Note that if we take the derivative with respect to Σ_j, we will find:

$$Q=-\frac{d}{2}\log(2\pi)-\frac{1}{2}\log\left|\boldsymbol{\Sigma}_j\right|-\frac{1}{2}\left(\boldsymbol{x}_i-\boldsymbol{\mu}_j\right)^\top\left(\boldsymbol{\Sigma}_j\right)^{-1}\left(\boldsymbol{x}_i-\boldsymbol{\mu}_j\right)+\log(\pi_j)-\log(\mathbf{F}_{ij}^{(t)})$$

$$\begin{aligned}
\frac{\partial Q}{\partial\boldsymbol{\Sigma}_j}&=\frac{\partial}{\partial\boldsymbol{\Sigma}_j}\left(-\frac{d}{2}\log(2\pi)\right)+\frac{\partial}{\partial\boldsymbol{\Sigma}_j}\left(-\frac{1}{2}\log\left|\boldsymbol{\Sigma}_j\right|\right)\\
&\quad+\frac{\partial}{\partial\boldsymbol{\Sigma}_j}\left(-\frac{1}{2}\left(\boldsymbol{x}_i-\boldsymbol{\mu}_j\right)^\top\boldsymbol{\Sigma}_j^{-1}\left(\boldsymbol{x}_i-\boldsymbol{\mu}_j\right)\right)+\frac{\partial}{\partial\boldsymbol{\Sigma}_j}\left(\log(\pi_j)\right)+\frac{\partial}{\partial\boldsymbol{\Sigma}_j}\left(-\log(\mathbf{F}_{ij}^{(t)})\right)\\
&=-\frac{1}{2}\boldsymbol{\Sigma}_j^{-1}+\frac{1}{2}\boldsymbol{\Sigma}_j^{-1}\left(\boldsymbol{x}_i-\boldsymbol{\mu}_j\right)\left(\boldsymbol{x}_i-\boldsymbol{\mu}_j\right)^\top\boldsymbol{\Sigma}_j^{-1}\\
&=\frac{1}{2}\boldsymbol{\Sigma}_j^{-1}\left[\left(\boldsymbol{x}_i-\boldsymbol{\mu}_j\right)\left(\boldsymbol{x}_i-\boldsymbol{\mu}_j\right)^\top-\boldsymbol{\Sigma}_j\right]\boldsymbol{\Sigma}_j^{-1}
\end{aligned}$$

In practice, we replace $\boldsymbol{\mu}_j$ by $\boldsymbol{\mu}_j^{(t+1)}$, and set this derivation to zero i.e.

$$\boldsymbol{\Sigma}_j^{(t+1)}=\sum_{i=1}^{n}\mathbf{F}_{ij}^{(t)}\left(\boldsymbol{x}_i-\boldsymbol{\mu}_j^{(t+1)}\right)\left(\boldsymbol{x}_i-\boldsymbol{\mu}_j^{(t+1)}\right)^\top/\sum_{i=1}^{n}\mathbf{F}_{ij}^{(t)}$$

And note that in a **mixture of spherical Gaussian** model, $\boldsymbol{\Sigma}_j=\sigma_j^2\boldsymbol{I}_d$ for all $j=1,\ldots,k$ specificly:

$$\frac{\partial Q}{\partial \boldsymbol{\sigma}_j} = \sum_{i=1}^{n} \mathbf{F}_{ij}^{(t)} \left(-\frac{d}{2\sigma_j^2} + \frac{(x_i - \mu_j)^\top (x_i - \mu_j)}{2\sigma_j^4} \right) \tag{6}$$

So that set this to zero, replace μ by $\mu_j^{(t+1)}$:

$$\sigma_j^{2\ (t+1)} = \frac{1}{d} \frac{\sum_{i=1}^{n} \mathbf{F}_{ij}^{(t)} (x_i - \mu_j^{(t+1)})(x_i - \mu_j^{(t+1)})^\top}{\sum_{i=1}^{n} \mathbf{F}_{ij}^{(t)}} \tag{7}$$

Now we derive the M-step update for the parameters π_j. Grouping together only the terms that depend on π_j, we find that we need to maximize

$$\sum_{i=1}^{n} \sum_{j=1}^{k} \mathbf{F}_{ij}^{(t)} \log \pi_j$$

with the constraint that $\sum_{j=1}^{k} \pi_j = 1$. To do this, we construct the Lagrangian

$$\mathcal{L}(\pi) = \sum_{i=1}^{n} \sum_{j=1}^{k} \mathbf{F}_{ij}^{(t)} \log \pi_j + \beta \left(\sum_{j=1}^{k} \pi_j - 1 \right)$$

where β is the Lagrange multiplier. Taking derivatives, we find

$$\frac{\partial}{\partial \pi_j} \mathcal{L}(\pi) = \sum_{i=1}^{n} \frac{\mathbf{F}_{ij}^{(t)}}{\pi_j} + \beta$$

Setting this to zero and solving, we get

$$\pi_j^{(t+1)} = \frac{\sum_{i=1}^{n} \mathbf{F}_{ij}^{(t)}}{-\beta}$$

Using the constraint that $\sum_j \pi_j = 1$ and the fact $\mathbf{F}_{ij}^{(t)} = p_{\boldsymbol{\theta}^{(t)}}(Z_i = j \mid \boldsymbol{x}_i)$, we easily find that

$$-\beta = \sum_{i=1}^{n} \sum_{j=1}^{k} \mathbf{F}_{ij}^{(t)} = \sum_{i=1}^{n} 1 = n$$

We therefore have our M-step updates for the parameters π_j :

$$\pi_j^{(t+1)} := \frac{1}{n} \sum_{i=1}^{n} \mathbf{F}_{ij}^{(t)}$$

- **Part f** (M-step for mixture of diagonal Gaussians) Under the **mixture of diagonal Gaussians** model, derive the M-step updating equations for $\boldsymbol{\mu}_j^{(t+1)}, \boldsymbol{\Sigma}_j^{(t+1)}$ and $\pi_j^{(t+1)}$.

Since we do not use any assumption of $\boldsymbol{\Sigma}_j^{(t)}$ to get $\pi_j^{(t+1)}$ and $\mu_j^{(t+1)}$, the conclusion of (e) still holds:

$$\pi_j^{(t+1)} := \frac{1}{n} \sum_{i=1}^{n} \mathbf{F}_{ij}^{(t)}$$

$$\boldsymbol{\mu}_j^{(t+1)} := \frac{\sum_{i=1}^n \mathbf{F}_{ij}^{(t)} \boldsymbol{x}_i}{\sum_{i=1}^n \mathbf{F}_{ij}^{(t)}}$$

For Σ_j:

$$\begin{aligned}
\frac{\partial Q}{\partial \boldsymbol{\Sigma}_j} &= \frac{\partial}{\partial \boldsymbol{\Sigma}_j}\left(-\frac{d}{2}\log(2\pi)\right) + \frac{\partial}{\partial \boldsymbol{\Sigma}_j}\left(-\frac{1}{2}\log|\boldsymbol{\Sigma}_j|\right) \\
&\quad + \frac{\partial}{\partial \boldsymbol{\Sigma}_j}\left(-\frac{1}{2}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)^\top \boldsymbol{\Sigma}_j^{-1}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)\right) + \frac{\partial}{\partial \boldsymbol{\Sigma}_j}\left(\log(\pi_j)\right) + \frac{\partial}{\partial \boldsymbol{\Sigma}_j}\left(-\log(\mathbf{F}_{ij}^{(t)})\right) \\
&= -\frac{1}{2}\boldsymbol{\Sigma}_j^{-1} + \frac{1}{2}\boldsymbol{\Sigma}_j^{-1}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)^\top \boldsymbol{\Sigma}_j^{-1} \\
&= \frac{1}{2}\boldsymbol{\Sigma}_j^{-1}\left[\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)^\top - \boldsymbol{\Sigma}_j\right]\boldsymbol{\Sigma}_j^{-1}
\end{aligned}$$

And set this to zero and replace μ by $\mu_j^{(t+1)}$:

$$\boldsymbol{\Sigma}_j^{(t+1)} = \sum_{i=1}^n \mathbf{F}_{ij}^{(t)}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j^{(t+1)}\right)\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j^{(t+1)}\right)^\top / \sum_{i=1}^n \mathbf{F}_{ij}^{(t)}$$

Hint: Almost all parts above are provided in the class slides. You are required to understand it, replicate it and fill in the missing parts, if any, from the class slides.

3. (**30 Points**) We now implement the EM algorithm from last question to cluster the MNIST data set.

(i) Program the EM algorithm you derived for mixture of spherical Gaussians. Assume 5 clusters. Terminate the algorithm when the fractional change of the log-likelihood goes under 0.0001. (Try 3 random initializations and present the best one in terms of maximizing the likelihood function).

(ii) Program the EM algorithm you derived for mixture of diagonal Gaussians. Assume 5 clusters. Terminate the algorithm when the fractional change in the log-likelihood goes under 0.0001. (Try 3 random initializations and present the best one in terms of maximizing the likelihood function).

Note that to assign a sample $\boldsymbol{x}_i$ to a cluster j, you first calculate $\mathbf{F}_{ij}$ using the parameters from the last iteration of EM algorithm you implemented. Next, assign sample $\boldsymbol{x}_i$ to the cluster j for which $\mathbf{F}_{ij}$ is maximum, i.e., the probability of sample i belonging to cluster j is maximum. Recall that the dataset has the true labels for each classes. Calculate the error of the algorithm (for the two different model). Here, error is just the number of mis-clustered samples divided by the total number of samples. In your opinion, were mixture models of Gaussian distributions suitable for modeling the MNIST data?

Hint: For these implementations, you will run into three different problems. Apply these following hints to solve each problem.

- 1) Use the log-sum-exp trick to avoid underflow on the computer. You will run into this problem when computing the log-likelihood. That is, when you calculate $\log \sum_j \exp^{a_j}$ for some sequence of variables a_j, calculate instead $A + \log \sum_j \exp^{a_j - A}$ where $A = \max_j a_j$.
- 2) Some pixels in the images do not change throughout the entire dataset. (For example, the top-left pixel of each image is always 0, pure white.) To solve this, after updating the covariance matrix $\mathbf{\Sigma}_j$ for the mixture of diagonal Gaussians, add $0.05\boldsymbol{I}_d$ to $\mathbf{\Sigma}_j$ (ie: add 0.05 to all the diagonal elements).
- 3) Be mindful of how you initialize $\mathbf{\Sigma}_j$. Note that for a diagonal matrix $\mathbf{\Sigma}_j$, the determinant $|\mathbf{\Sigma}_j|$ is the product of all the diagonal elements. Setting each diagonal element to a number too big at initialization will result in overflow on the computer.

Type 1 (Diagonal)

Seed: 6657

label	0	1	2	3	4
cluster					
0	105	55	981	971	292
1	2883	1688	663	2915	3571
2	1270	182	2180	1106	1307
3	1663	129	2116	1081	649
4	2	4688	18	58	23

Table 1: Confusion Matrix for Seed: 6657, Type: 1 (Diagonal)

Prediction Error Rate: 0.8341

Seed: 7470

label	0	1	2	3	4
cluster					
0	1470	26	409	420	249
1	2774	6332	687	2818	3574
2	98	84	109	98	626
3	810	68	864	1550	1046
4	771	232	3889	1245	347

Table 2: Confusion Matrix for Seed: 7470, Type: 1 (Diagonal)

Prediction Error Rate: 0.6794(BEST)

Seed: 9999

label	0	1	2	3	4
cluster					
0	356	50	1277	815	213
1	1	0	0	0	0
2	81	42	622	666	313
3	949	4134	3013	1818	639
4	4536	2516	1046	2832	4677

Table 3: Confusion Matrix for Seed: 9999, Type: 1 (Diagonal)

Prediction Error Rate: 0.7558

Type 2 (Spherical)

Seed: 6657

label	0	1	2	3	4
cluster					
0	5202	0	461	133	63
1	153	81	800	329	5625
2	4	2803	47	21	41
3	1	3619	49	124	58
4	563	239	4601	5524	55

Table 4: Confusion Matrix for Seed: 6657, Type: 2 (Spherical)

Prediction Error Rate: 0.8199

Seed: 7470

label	0	1	2	3	4
cluster					
0	4911	0	81	74	29
1	6	6008	148	188	143
2	121	41	383	211	5507
3	374	231	1601	4916	36
4	511	462	3745	742	127

Table 5: Confusion Matrix for Seed: 7470, Type: 2 (Spherical)

Prediction Error Rate: 0.4658 (BEST)

Seed: 9999

label	0	1	2	3	4
cluster					
0	5122	0	222	87	44
1	1	0	0	0	0
2	173	131	671	419	5600
3	620	511	4923	5401	61
4	7	6100	142	224	137

Table 6: Confusion Matrix for Seed: 9999, Type: 2 (Spherical)

Prediction Error Rate: 0.6297

In our EM alorithms, we choose to intialize mean with sample mean adding a Gaussian random variable. For variance, we use the sample variance of each variable, multiplied by a Gaussian random variable with mean 1. We also add 0.05 to prevent numeric issues.

We point out the best results for diagonal method and spherical method in the following chart:

	Type 1 (Diagonal)	Type 2 (Spherical)
Seed	7470	7470
Prediction Error Rate	0.6794	0.4658
Log-Likelihood	-12950218.11532	-23852618.59406
π	[0.084, 0.53, 0.033, 0.141, 0.212]	[0.164, 0.215, 0.205, 0.24, 0.176]

Table 7: Comparison of Best Results for Type 1 (Diagonal) and Type 2 (Spherical)

Pledge:
Please sign below (print full name) after checking (✓) the following. If you cannot honestly check each of these responses, please email me at kbala@ucdavis.edu to explain your situation.

- I pledge that I am a honest student with academic integrity and I have not cheated on this homework.
- These answers are my own work.
- I did not give any other students assistance on this homework (beyond what is allowed as per syllabus).
- I understand that to submit work that is not my own and pretend that it is mine is a violation of the UC Davis code of conduct and will be reported to Student Judicial Affairs.
- I understand that suspected misconduct on this homework will be reported to the Office of Student Support and Judicial Affairs and, if established, will result in disciplinary sanctions up through Dismissal from the University and a grade penalty up to a grade of "F" for the course.

Signature: Jingzhi Sun