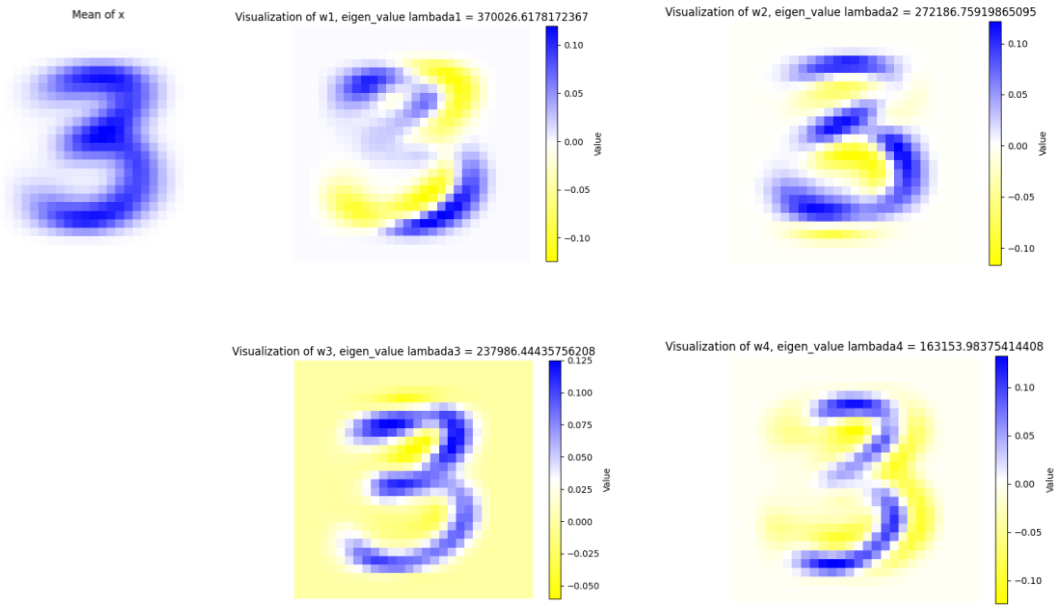
# Pattern Recognition Experiment 3

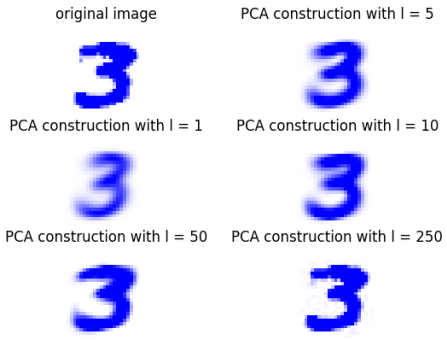
EECS26 111060024 蔡孟伶

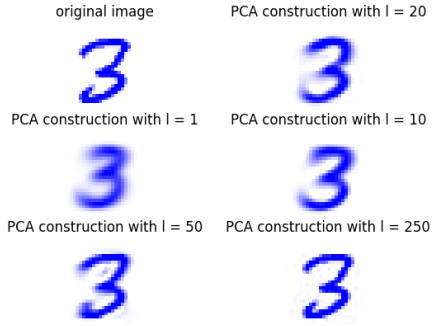
1. Download the MNIST data set of handwritten digits from:
   https://www.kaggle.com/datasets/oddrationale/mnist-in-csv
   (a) Create a data matrix $\mathbf{X} = [\mathbf{x}_1 - \boldsymbol{\mu}_\mathbf{x}, \mathbf{x}_2 - \boldsymbol{\mu}_\mathbf{x}, \dots, \mathbf{x}_N - \boldsymbol{\mu}_\mathbf{x}] \in R^{d \times N}$ from $N = 2000$ randomly sampled instances of the digit "3" from the training dataset (mnist_train.csv). The dimension of each instance is $d = 784$.
   (b) Compute the principal components of the covariance matrix of $\mathbf{X}$ as well as the corresponding variances (eigenvalues). Please refer to P. 26 of Ch6 slides to show similar results (e.g., the PCA bases and the dimension-reduced reconstructions)
   (c) Evaluate the reconstruction error with different settings of the reduced dimension $l = 1, 10, 50, 250$ and draw your conclusion.

(a)




(b) Result image




```
please enter the l'th largest principal component you want:5
Do you want to generate the mean of X, too? Type yes or no:yes
average error of  l = 5, 1747899.4100599536
average error of  l = 1, 2534948.0778994295
average error of  l = 10, 1340574.9526771726
average error of  l = 50, 449195.8168071182
average error of  l = 250, 38221.24519930661
```

```
please enter the l'th largest principal component you want:20
Do you want to generate the mean of X, too? Type yes or no:no
average error of  l = 20, 912937.3250813447
average error of  l = 1, 2559921.692627765
average error of  l = 10, 1339409.6914112442
average error of  l = 50, 448202.1114251014
average error of  l = 250, 38397.92579721615
```

(c) Conclusion:

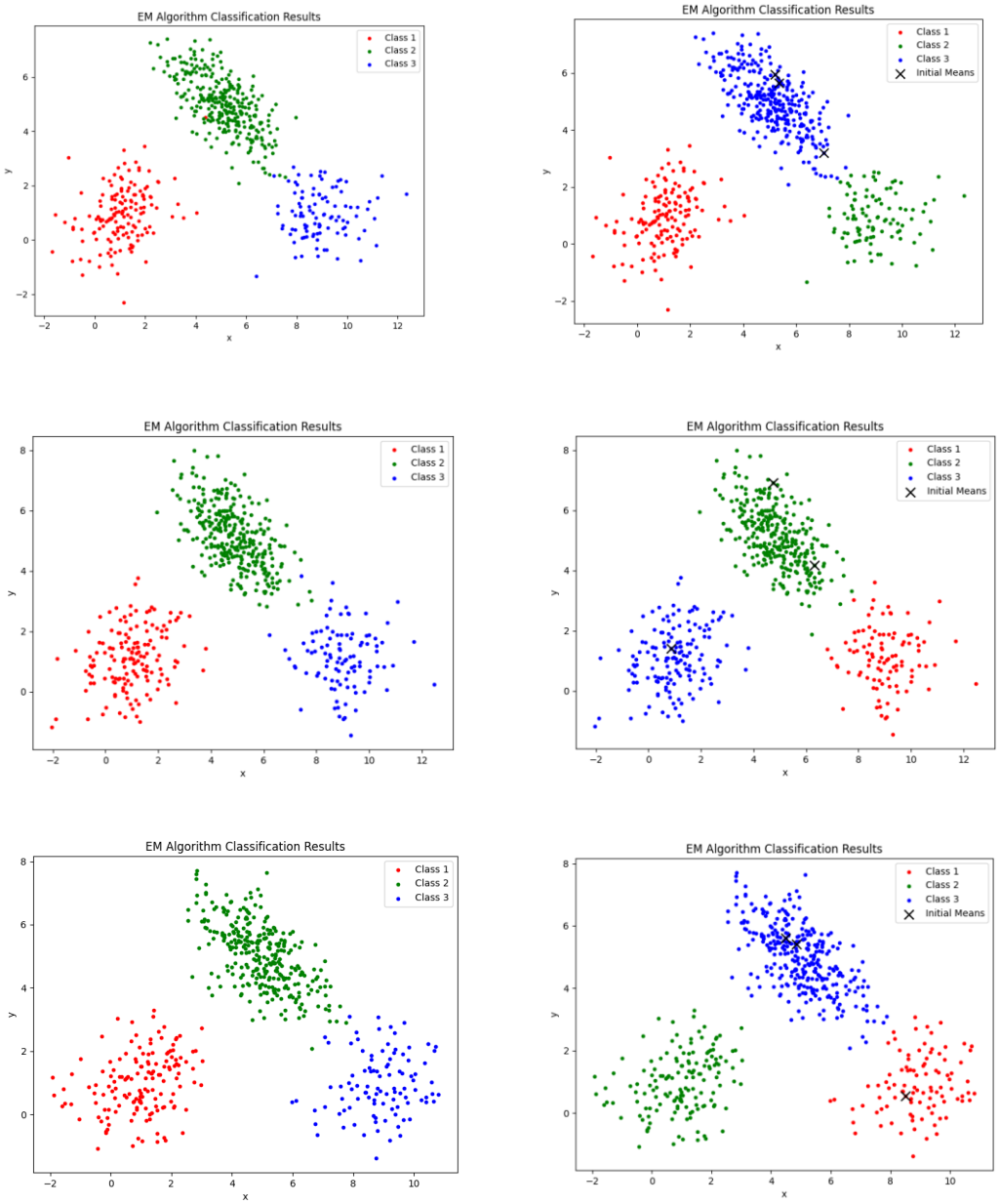
From the two cases tested, we can see that the average construction error of dimension l decreased intensely as l increased. The formula of average construction error is

$$J(w) = \frac{1}{N}\sum_{i=1}^{N} ||x_i - Wz_i||^2$$

This reconstruction error is the average of the whole dataset's data, with each summing up the pixel-wise difference's square to the reconstructed image. The relationship between dimension l and reconstruction error is reasonable since we start from the eigenvector corresponding to the biggest eigenvector, as more feature involved, the output should become closer the original input.

(EM algorithm for Gaussian mixtures)

2. Generate 150 2-D samples from one Gaussian $N(\mu_1, \Sigma_1)$, another 300 samples from $N(\mu_2, \Sigma_2)$, and 100 samples from $N(\mu_3, \Sigma_3)$, with mean vectors $\mu_1 = \begin{bmatrix}1\\1\end{bmatrix}$, $\mu_2 = \begin{bmatrix}5\\5\end{bmatrix}$, $\mu_3 = \begin{bmatrix}9\\1\end{bmatrix}$ and covariance matrices $\Sigma_1 = \begin{bmatrix}1 & 0.4\\0.4 & 1\end{bmatrix}$, $\Sigma_2 = \begin{bmatrix}1 & -0.6\\-0.6 & 1\end{bmatrix}$, $\Sigma_3 = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$, respectively.
   (a) Use EM algorithm and the generated samples to estimate the unknown parameters $\boldsymbol{\mu}_i, \Sigma_i, P_i$ $(i = 1,2,3)$. Please specify your experimental settings (e.g., initialization, stopping criterion) in the report.
   (b) Repeat the mixture density estimation by EM when the mean vectors are $\begin{bmatrix}1\\1\end{bmatrix}, \begin{bmatrix}2\\2\end{bmatrix}, \begin{bmatrix}3\\1\end{bmatrix}$, respectively.
   (c) Compare the results (in terms of confusion matrices or 2-D visualization) and draw your conclusions.




(a) Initialization and stopping criterion setting for the EM algorithm:

Three parameters set at initialization of EM are the mean, covariance matrix and the Prior probability. At first, mean is determined by randomly sampling three(since we want three groups) data from the dataset. Covariance matrix is initialized as 2 * 2 identity matrix for each class, and the prior is initialized as 1 / K, which is equal prior.

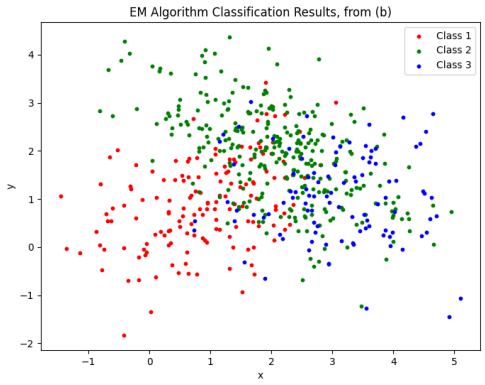
There are two stopping criterions for the EM algorithm. First, a maximum of iteration = 100 is set, and second, when the new parameters are updated at M-step is close to the old parameters. The threshold is set as 10^-4, when the total of np.linalg.norm(X) of mean, covariance and prior is small, the loop

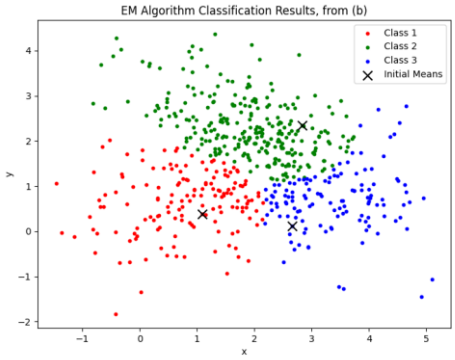
stops early. The norm operation is shown below.

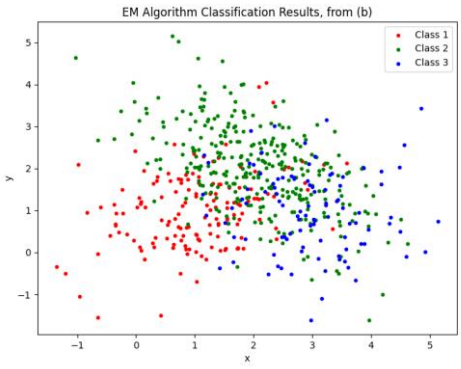
$$norm(X) = \sqrt{\sum_i {X_i}^2}$$

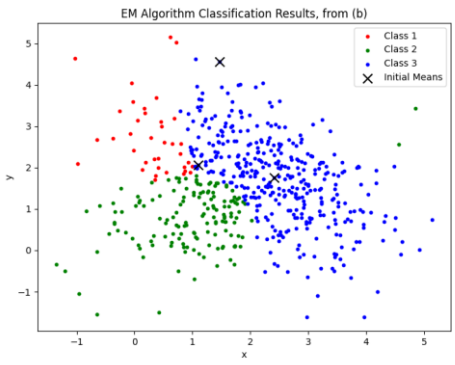
(c) Conclusion:

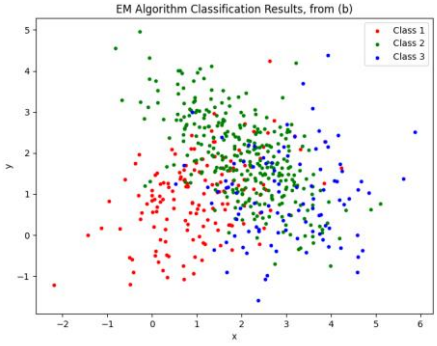
From the three test run shown above, we can see that although the initialization of mean may not be very nice or well-separated, EM could still find the cluster of the data. However, it only clusters data into group but didn't guarantee the correctness of classification class. Though the data that belongs to class 1 at ground truth is grouped together, it may be classified as group 2 depending on the initialization sample data. For dataset (b), we can see the following result. The clustering result is not as clear as dataset (a), since there are many overlapping data that are hard to be separate by using usual linear methods. EM intends to cluster data points that are close to one group, making it harder to deal with overlapping data.

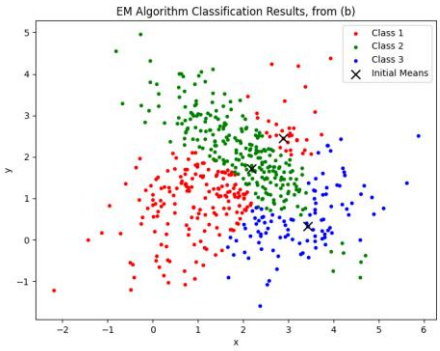




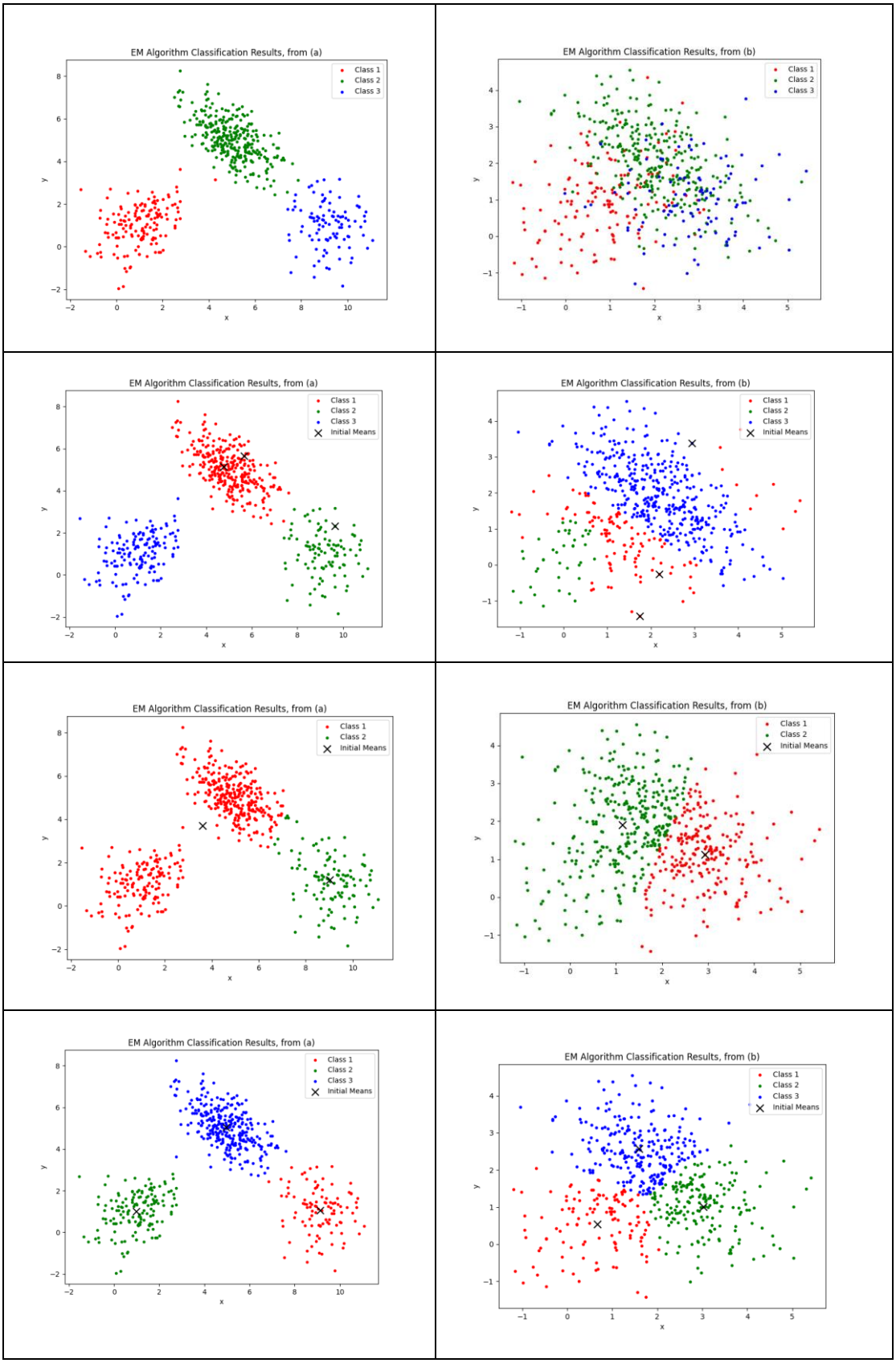

(K-means algorithm)
Use K-means algorithm on the two data sets you generate in Problem 2, for $K = 2,3,4$. Compare the results and draw your conclusions.

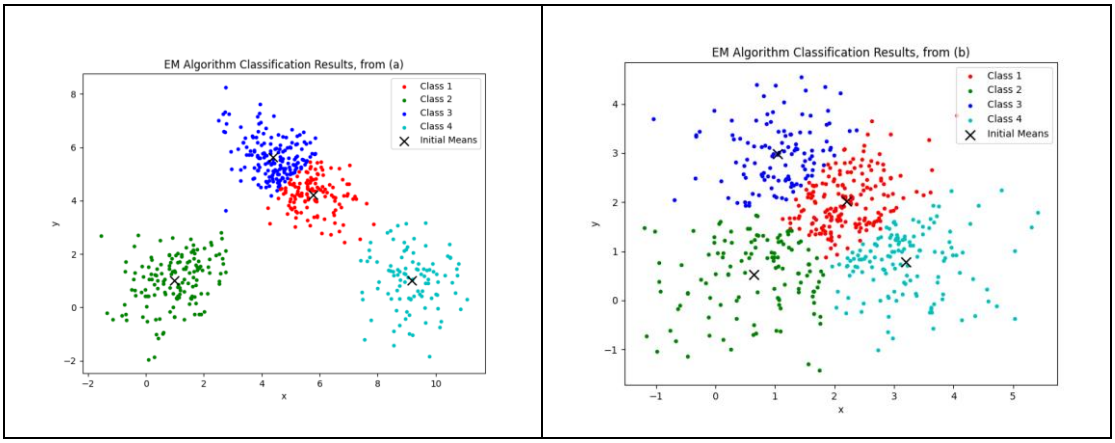
Setting the random seed in shared_dataset.py, we will be able to generate same dataset even executing the file separately. Below will show the result of problem 2: EM algorithm along with problem 3: K-means. Two dataset are on two columns, and the result of each row is in order of ground_truth, EM_algorithm, K-means(K = 2), K-means(K = 3), and K-means(K = 4).

Conclusion:

From the result, I think the result of EM algorithm and K-means(K=3) under dataset 1 are both acceptable, while for dataset 2, K-means(K=3) seems to cluster more data from the same group in ground truth than EM does in this case. However, we should still seek for better representation data or else such clustering doesn't really fit the dataset. It is also a problem worth trying of what K to decide, since in real world, we don't have the ground truth label. Another observation on the K-means result is that it didn't take into account of the prior of each class, from the K-means result(K = 2, K= 3, K=4), I see that each class has similar portion. However, inside EM algorithm, we take into account prior variable and update it each time, therefore for some dataset with larger prior differences among classes, I think EM algorithm might perform better on each cluster's portion compared with K-means. Last, since the principal of clustering inside K-means is about distance, it is not likely to have results like what we have in EM algorithm for dataset(b). Some red data are separated from other red ones by blue data. When facing some distribution to have one class caught the others

in the middle, K-means might not be appropriate for that.




EM Algorithm Classification Results, from (a)
Class 1
Class 2
Class 3
Class 4
Initial Means
x
y
EM Algorithm Classification Results, from (b)
Class 1
Class 2
Class 3
Class 4
Initial Means
x
y