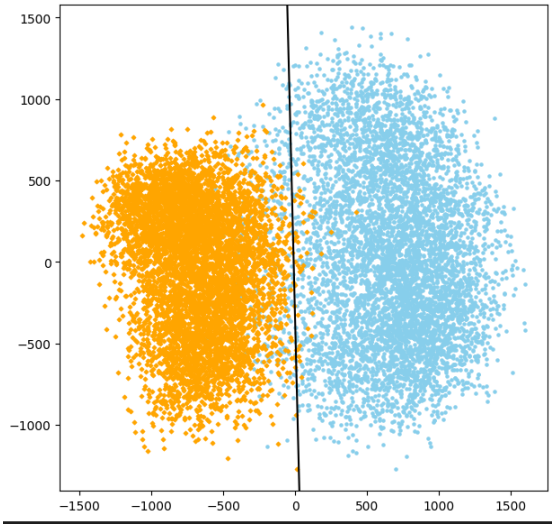
SYDE 675 : Introduction to Pattern Recognition
Assignment1

Tzu-Ting, Huang / 20988611

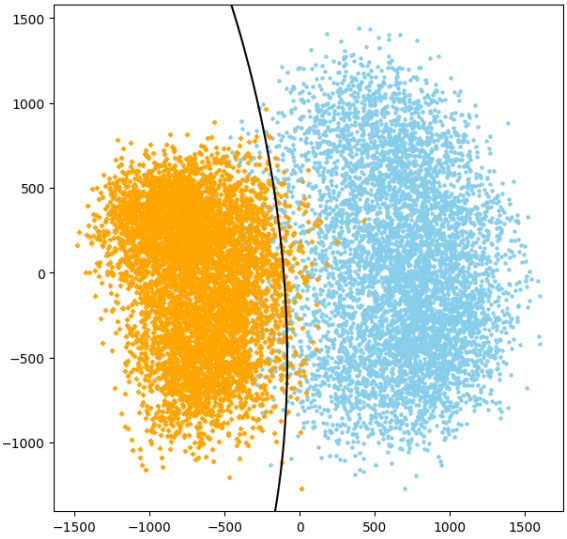
Exercise 1: MED and MMD Classifiers

1. **Implement the algorithm for the MED and MMD classifiers on the training set of the MNIST dataset. Determine the decision boundary for the two classifiers. Plot the boundary along with the training data.**

MED



MMD



2. **Use the test set of the two classes in the MNIST dataset and make label predictions for all the test vectors for the two classes using the MED and MMD classifiers. Report the classification accuracy of the classifiers using:**

   Accuracy_med_test: 0.9764056224899599
   Accuracy _mmd_test: 0.9819277108433735

3. **Which of the two classifiers is better? Explain.**

   The MMD classifier, which considers the covariance of the data, can better handle the high dimensionality and reduce the sensitivity to outliers. Also, the MNIST data contains some noise and outliers, so the MMD classifier performs higher accuracy than the MED classifier, which only considers the mean.

4. **Now, for the same two classes in the MNIST dataset, use PCA to convert the images to 20 x 1 vectors. Repeat steps 1-3 for this dataset. For step 1, can you plot the decision boundary for this dataset? Explain.**

   Accuracy_med_test: 0.981425702811245
   Accuracy _mmd_test: 0.9979919678714859

   It is not feasible to visualize a decision boundary when working with a dataset that has 20 dimensions. Typically, a decision boundary is plotted in two or three dimensions.

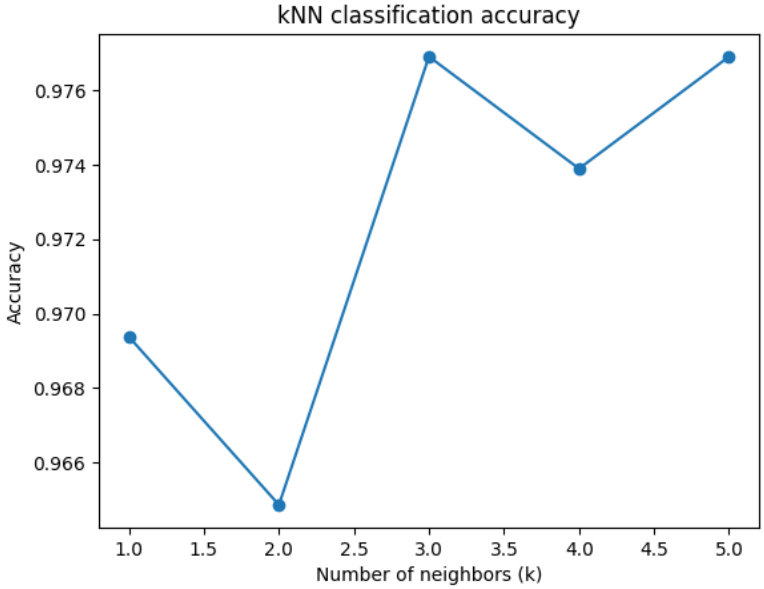
Exercise 2: Nearest Neighbor Classification and Regression

1. **Implement the k-nearest neighbour classifier on the MNIST dataset with 2 x1 vectors for the same two classes used in the previous exercise. Use euclidean distance as the distance metric. Compute the kNN solution for each integer k from 1 to 5.**

   Accuracy: 1.0
   Accuracy: 0.9835463125365406
   Accuracy: 0.9836298337927002
   Accuracy: 0.9793702497285559
   Accuracy: 0.9800384197778335

2. **Use the test set of the two classes and find the classification accuracy for all kNN classifiers. Plot the accuracy for each value of k.**

   [0.9693775100401606, 0.964859437751004, 0.9769076305220884, 0.9738955823293173, 0.9769076305220884]



3. **Which k value seems to be producing the best results? Why?**

   When k equals 3 or 5, the results are optimal in the result. However, when k is smaller, it becomes sensitive to noise in the data. Additionally, odd values of k generally perform better than even values. This is because even values may lead to tie situations, which can sometimes make the decision-making process ambiguous, especially in binary classification.

4. **How does the kNN classifier compare against the MED and MMD classifiers in the previous exercise?**

   The kNN classifier is more flexible than both the MED and MMD classifiers, particularly when dealing with complex decision boundaries. However, the kNN classifier can be computationally expensive when handling large datasets and is more susceptible to noise.

5. **Now let's do kNN regression, which is similar to classification, but instead of aggregating labels by taking the majority, we average the y values of the k nearest neighbours. For this, use the training set of the d = 20 mystery dataset F available on the course website, and compute the kNN regression solution with each integer k from 1 to 3. Report the mean squared error**

using the test set of the mystery dataset F for the three kNN classifiers. Which k value seems to be giving you the best result? Explain.

Mean Squared Error for k=1: 5.345166127093706
Mean Squared Error for k=2: 4.390415367943646
Mean Squared Error for k=3: 3.8738548973640072

k=3 yields the lowest MSE of 3.8738548973640072, so it is the best for this dataset. The reason k=3 performs better might be due to it considering more neighboring points, which can lead to a more robust estimation of the target variable.

Exercise 3: ML and MAP Classifiers

1. Find the sample mean and covariance for the training set of the two classes in the MNIST dataset and estimate the probability of the two classes as Gaussian distributions. Based on this, develop an ML classifier and report the classification accuracy on the test set of the two classes.

ML classification accuracy: 0.981425702811245

2. Now, let's assume that the prior probabilities for the two classes are p(C1) = 0.58 and p(C2) = 0.42. Using these prior probabilities, and the means and covariances of the two classes, develop an MAP classifier and report the classification accuracy on the test set.

MAP classification accuracy: 0.9774096385542169

3. Based on the results, do you think that assuming the probability distributions of the two classes as Gaussian was correct? Explain.

In both MAP and ML classifications, I've used the Gaussian probability density function (pdf) as a modeling choice. The high accuracy achieved by both classifiers suggests that the features of classes 3 and 4 may indeed follow a Gaussian distribution. However, further statistical tests would be needed for confirmation.

4. (5 pts) Compare the ML, MAP, MED, MMD, and kNN classifiers based on the classification accuracy. Which classifier is the best? Could the inferior classifiers be better for different datasets? Explain.

ML Accuracy: 0.9814
MAP Accuracy: 0.9774
MED Accuracy: 0.9764
MMD Accuracy: 0.9819
kNN (k=1): 0.9694
kNN (k=2): 0.9649
kNN (k=3): 0.9769
kNN (k=4): 0.9739
kNN (k=5): 0.9769

When it comes to high accuracy, the MMD and ML classifiers outperform others and their accuracy is very close. The MMD classifier has a slightly higher accuracy than the ML classifier, making it the best classifier for this particular case. However, it's important to note that a classifier that performs well on one dataset may not perform as well on another. For instance, the kNN classifier may work better on a dataset with complex decision boundaries, while the ML or MAP classifiers would be more suitable for a dataset where the class distributions are known and follow the assumed distribution.