SYDE 572 - Lab 1

Clusters and Classification Boundaries

Group 19

Jonathan Chen	20722167
Ellen Choi	20707132
Aman Mathur	20710307
Michael Eden	20678312

1 Introduction

Lab 1 of SYDE 572 investigates the development of clusters and classification boundaries in Matlab. Five classes, split into two cases, were analyzed to compare the effectiveness of MED, MICD, MAP, NN, and 5NN classifiers. This was accomplished by initially generating the 2D random clusters for the five classes using the **randn** function in Matlab. Subsequently, the classification boundaries between the classes using each classifier were plotted. Finally, the classification performance was analyzed by calculating the experimental error rate and the confusion matrix for each classifier.

2 Implementation and Results

2.1 Generating Clusters

The clusters were generated by applying a transformation to the **randn** function in Matlab. Since **randn** generates random numbers from a standard normal distribution, the transform found in *generateBivariateCluster.m* was applied to obtain the correlated, un-equal variance data. This involved multiplying the randomly generated data by a weight matrix, which was the inverse of the whitening-tranformed, orthonormal covariance transform, to obtain the new samples. Figures 1 and 2 show the clusters and standard deviation contours for Case 1 and Case 2, respectively.

The unit standard deviation contours were plotted using the provided *plotEllipse.m* function. The major axis was determined to be the eigenvector of the largest eigenvalue of the covariance matrix, while the angle from the horizontal was the inverse tangent of the second element of the eigenvector of the major axis, divided by the first ($\theta = atan(\frac{2ndElement}{1stElement})$). Finally, the axis lengths a and b were the square roots of the larger and smaller eigenvalues, respectively.

Visually, how does the unit contour relate to the cluster data?

Visually, the unit contours closely relate to the cluster data. The contours follow the general direction and spread of the data points. To verify that the contours were plotted correctly, the centres of the contours were compared against the means of the classes and the shape of the contours were compared against the covariance matrices, where the *a, b, and c* elements of each covariance matrix were verified against the plotted contours to check their directions, shapes, and sizes. All of the contours matched the expected bounding boxes, so the clusters and contours were verified to be plotted correctly.

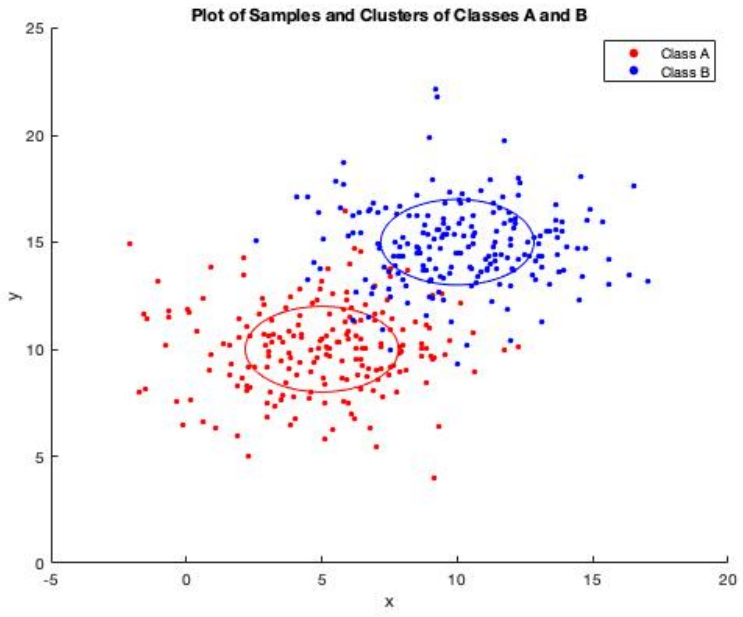

Figure 1: Clusters and Standard Deviation Contours of Classes A and B

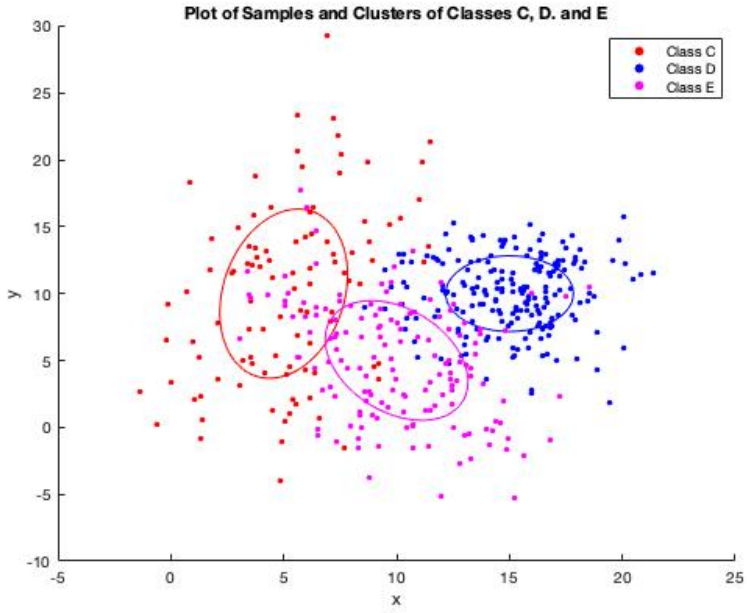

Figure 2: Clusters and Standard Deviation Contours of Classes C, D, and E

2.2 Classifiers

The classifiers were evaluated numerically using a 2D grid of points in Matlab. The grid was created by taking 500 evenly spaced points between the minimum and maximum values of each case, using the *linspace* function in Matlab. Each point in the 2D grid was classified using the Minimum Euclidean Distance (MED), Minimum Intra-Class Distance (MICD), Maximum a Posteriori (MAP), Nearest Neighbour (NN), and 5-

Nearest Neighbour (5NN) classifiers. Then, contours from the resulting decision boundaries were plotted on top of the original random clusters.

To create the decision boundaries for each classifier, the following discriminant functions were implemented in Matlab. For MED, the discriminant function was:

$$g_k(x) = -\vec{z_k}^T \vec{x} + \frac{1}{2}\vec{z_k}^T \vec{z_k} \tag{1}$$

where k is the class number, $\vec{z_k}$ is the class mean, and $\vec{x}$ is the point of interest. This was implemented as the Euclidean distance in Matlab in functions *med.m* and *ed.m*, where each point was assigned to the class that minimizes the discriminant function (i.e., the class with the mean with the smallest Euclidean distance from $\vec{x}$).

For MICD, the discriminant function was:

$$g_k(x) = (\vec{x} - \vec{z_k})^T S_k^{-1} (\vec{x} - \vec{z_k}) \tag{2}$$

where S_k^{-1} is the inverse covariance matrix of class k and $\vec{z_k}$ and $\vec{x}$ are the same as defined before. Similarly, each point was assigned to the class that minimized the discriminant function. MICD was implemented in *micd.m* for the two class case and *micd3.m* for the three class case.

For MAP, the discriminant function was:

$$\frac{P(\vec{x}|A)}{P(\vec{x}|b)} = \frac{P(B)}{P(A)} \tag{3}$$

where each point is assigned to the class with the greatest probability. MAP was implemented in *map.m* for the two class case and *map3.m* for the three class case.

For kNN, the decision function was:

$$g_k(x) = -\vec{z_k}^T \vec{x} + \frac{1}{2}\vec{z_k}^T \vec{z_k} \tag{4}$$

where $\vec{z_k}$ is the dynamically computed class prototype. NN can be considered a special case of kNN where $k = 1$, so the more general kNN method was implemented with k as a configurable parameter. The implementation may be found in *knn.m*. As in the other cases, the pattern is assigned to the class which minimizes the discriminant function. For a given pattern $\vec{x}$, the class prototype $\vec{z_k}$ is derived as follows:

1. Compute the Euclidean distance from $\vec{x}$ to all points belonging to that class.
2. Choose the k points with the minimum Euclidean distance from $\vec{x}$.
3. Compute the class prototype $\vec{z_k}$ as the mean of these k points.

Comment on the classification boundaries. How do the different boundaries compare?

The classification boundaries for the MED, MICD, and MAP classifier for Case 1 are displayed in Figure 3. All three classifiers produce linear decision boundaries. The MED line uses only the sample means and is the simplest of the three. This MED line is perpendicular (negative reciprocal slope) to the line between the means of the two classes. On the other hand, the MICD and MAP classifiers produce a decision boundary that considers information beyond just the sample mean. In Figure 3, the MICD and MAP decision boundaries overlap and are indistinguishable from each other. This is as expected because both the covariance matrices and the probabilities of Classes A and B are equivalent. As a whole, the MICD and MAP decision boundaries have a slope with a lower magnitude than MED and is more accurate.

In the three class case (Figure 4), the MED classifier continues to be the simplest with linear contours between the three clusters of data. The decision boundary passes through several of the unit standard contours

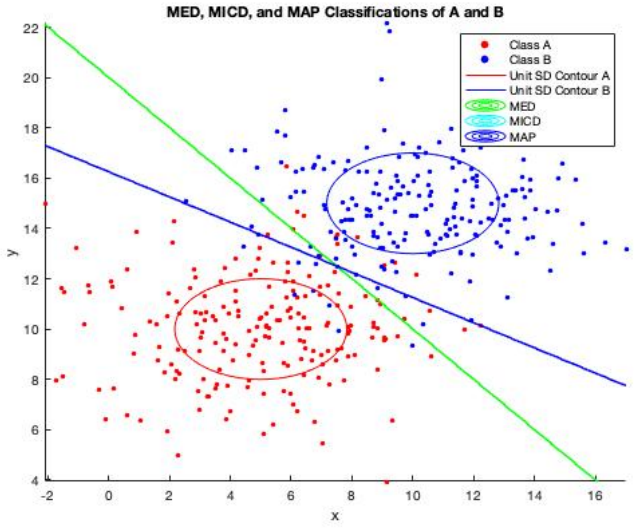

Figure 3: MED, MICD and MAP Classifications of A and B

and is thus the least accurate. Both the MICD and MAP classifiers produced non-linear boundaries that adjust based on the covariances of each of the classes, and are thus more accurate than the MED classifier. Additionally, the MICD decision boundary passes through all intersections of the cluster contours. Finally, the MAP decision boundary follows a similar shape to MICD, but is slightly more accurate because it accounts for the differences in covariance and probability of each of the classes.

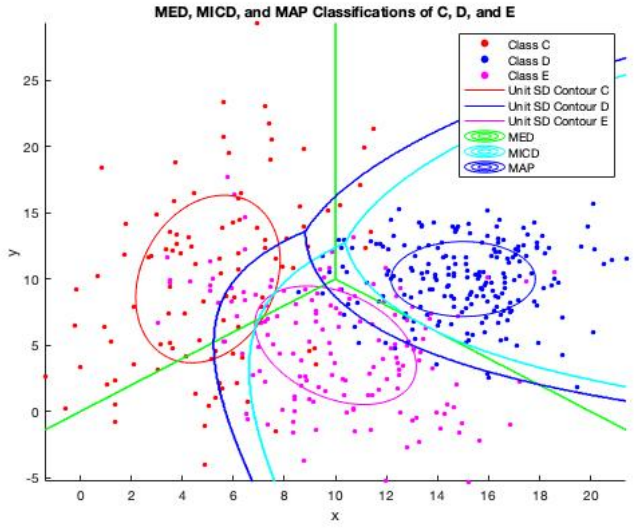

Figure 4: MED, MICD and MAP Classifications of C, D and E

The decision boundary for the NN classifier for Case 1 is displayed in Figure 5. As in MED, MICD, and MAP, one segment of the decision boundary passes between the two clusters. Unlike MED, MICD, and MAP, NN creates additional sub-boundaries to account for outlier values, so NN tends to overfit the data.

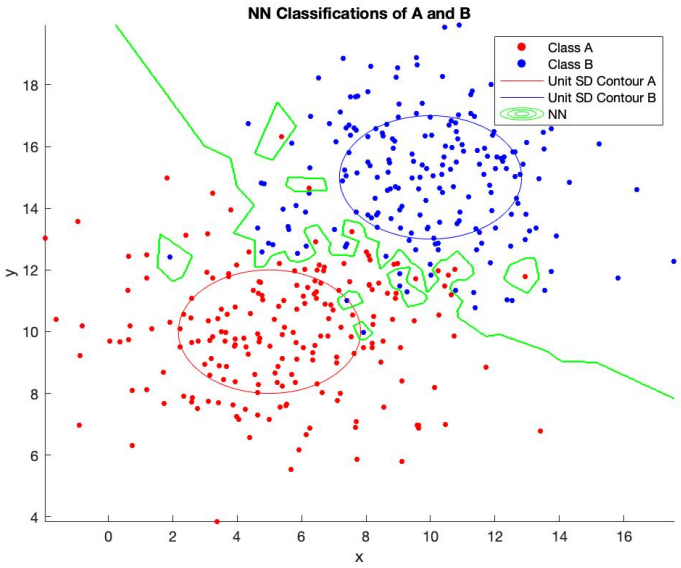

Figure 5: NN Classifications of A and B

The decision boundary for the 5NN classifier for Case 1 is displayed in Figure 6. Unlike with NN, the decision boundary mostly consists of one curve which passes between the two clusters. 5NN is less sensitive to outliers than NN because it computes the class prototype as the mean of the 5 points nearest to a given pattern. However, the decision boundary appears to wrap around groups of points, signifying that overfitting has still occurred, though to a lesser extent than in the case of NN.

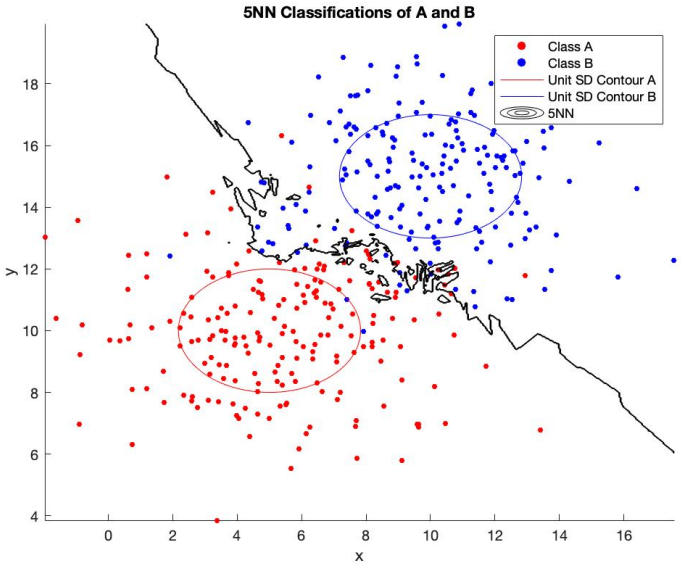

Figure 6: 5NN Classifications of A and B

The decision boundary for the NN classifier for Case 2 is displayed in Figure 7. Unlike MED, MICD, and MAP, the decision boundary does not clearly separate the 3 classes. Likely because there is significant overlap between the distributions, the NN decision boundary behaves erratically and passes through each distribution rather than between the distributions. Significant overfitting has occurred, so the NN classifier would likely perform poorly in this case.

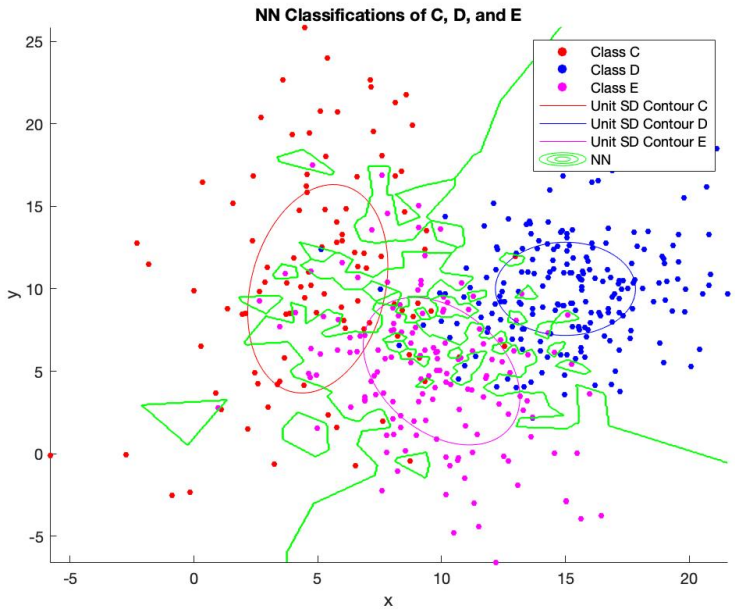

Figure 7: NN Classifications of C, D and E

The decision boundary for the 5NN classifier for Case 2 is displayed in Figure 8. Again, the decision boundary does not clearly separate the 3 classes. There are fewer isolated sub-boundaries than in the case of NN. Significant overfitting has occurred, although to a lesser extent than with NN.

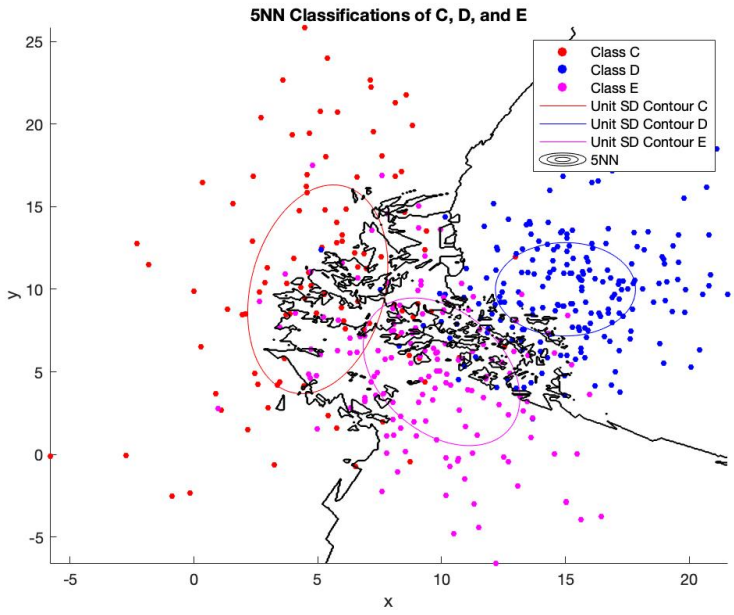

Figure 8: 5NN Classifications of C, D and E

3 Error Analysis

Compare the results. Which error is smallest? What do you observe in the confusion matrices for CASE2?

The probability of error for each class was calculated using the following equation:

$$P(\epsilon) = \frac{\#ofmisclassified}{total} \tag{5}$$

Using the values obtained from the confusion matrices, the probability of error for each class and case are calculated to be the following and summarized in Table 1.

Case 1

MED: $P(\epsilon) = (11 + 19)/400 = 0.075$
MICD: $P(\epsilon) = (10 + 12)/400 = 0.055$
MAP: $P(\epsilon) = (10 + 12)/400 = 0.055$
NN: $P(\epsilon) = (21 + 20)/400 = 0.0875$
5NN: $P(\epsilon) = (13 + 10)/400 = 0.0625$

Case 2

MED: $P(\epsilon) = (4 + 21 + 1 + 21 + 27 + 13)/450 = 0.19333$
MICD: $P(\epsilon) = (2 + 12 + 4 + 28 + 26 + 12)/450 = 0.18667$
MAP: $P(\epsilon) = (4 + 19 + 0 + 12 + 13 + 24)/450 = 0.16000$
NN: $P(\epsilon) = (7 + 35 + 4 + 20 + 26 + 27)/400 = 0.26444$
5NN: $P(\epsilon) = (3 + 22 + 4 + 26 + 26 + 11)/400 = 0.20444$

Table 1: Error Rate for Each Classifier

Classifier	**Case 1**	**Case 2**
MED	0.075	0.19333
MICD	0.055	0.18667
MAP	0.055	0.16000
NN	0.0875	0.26444
5NN	0.0625	0.20444

Table 2: Case 1 MED Confusion Matrix

		Predicted		
		A	**B**	**Total**
Actual	**A**	189	11	200
	B	19	181	200
		208	192	

Table 3: Case 1 MICD Confusion Matrix

		Predicted		
		A	**B**	**Total**
Actual	**A**	190	10	200
	B	12	188	200
		202	198	

Table 4: Case 1 MAP Confusion Matrix

		Predicted		
		A	**B**	**Total**
Actual	**A**	190	10	200
	B	12	188	200
		202	198	

Table 5: Case 1 NN Confusion Matrix

		Predicted		
		A	**B**	**Total**
Actual	**A**	181	19	200
	B	16	184	200
		197	203	

Table 6: Case 1 5NN Confusion Matrix

		Predicted		
		A	**B**	**Total**
Actual	**A**	185	15	200
	B	10	190	200
		195	205	

Table 1 above shows the error values for each classifier for both cases. For Case 1, MAP and MICD combined to have the smallest error. This was as expected because MAP and MICD formed the same boundary, as described in Section 2. Using the Euclidean distance metric to classify data points yielded similar results to the classifiers that are more computationally complex, however, there were incremental improvements as we started to account for more information (MED to MAP) and the probability of error decreased accordingly. In general for Case 1, there were only a few data points where the Euclidean distance caused an issue in classification. When using the nearest-neighbours approach, the error was greater than MED, GED and MAP since the NN classifier creates additional sub-boundaries to account for extreme/outlier values in the class and is therefore more sensitive to noise and prone to overfitting. However, the 5NN had a smaller error than NN because the prototype in 5NN is defined as the sample mean of the 5 samples within the cluster closest to $\vec{x}$, so the 5NN classifier is less sensitive to noise and outliers.

Table 7: Case 2 MED Confusion Matrix

		Predicted			
		C	**D**	**E**	**Total**
Actual	**C**	75	4	21	100
	D	1	178	21	200
	E	27	13	110	150
		103	195	152	

Table 8: Case 2 MICD Confusion Matrix

		Predicted			
		C	**D**	**E**	**Total**
Actual	**C**	86	2	12	100
	D	4	168	28	200
	E	26	12	112	150
		116	182	152	

Table 9: Case 2 MAP Confusion Matrix

		Predicted			
		C	**D**	**E**	**Total**
Actual	**C**	77	4	19	100
	D	0	188	12	200
	E	13	24	113	150
		90	216	144	

Table 10: Case 2 NN Confusion Matrix

		Predicted			
		C	**D**	**E**	**Total**
Actual	**C**	58	7	35	100
	D	4	176	20	200
	E	26	27	97	150
		88	210	152	

Table 11: Case 2 5NN Confusion Matrix

		Predicted			
		C	**D**	**E**	**Total**
Actual	**C**	75	3	22	100
	D	4	170	26	200
	E	26	11	113	150
		105	184	161	

For the second case, where there were three classes to categorize the points into, MAP had the lowest error. For Case 1, MICD and MAP had the same decision boundary, whereas for Case 2, there was a distinct difference since the covariance matrices of the classes were no longer the same. As expected, the MAP classified had the lowest probability of error because it considers the probabilities and the covariances of each of the distributions. As shown in the tables above, the number of correctly classified points decrease when we only use the Euclidean distance as the main metric. In particular, the MED classifier has a higher error percentage for Class E classification, when more samples are overlapping with other classes. In general, the NN and 5NN perform worse in the 3 class case because there is a greater amount of overlap in data points, which causes misclassifications to become easier for models that overfit such as NN and 5NN.

4 Summary and Conclusion

In summary, all of the classifiers had lower error rates in the 2 class case than the 3 class case, with the MICD and MAP classifiers having the lowest error. The MAP classifier was also identical in the two class case to the MICD classifier because the probabilities and covariance matrices were the same. The NN and 5NN classifiers had slightly higher error values because these classifiers were more prone to overfitting to the training dataset and were thus more sensitive to noise and outliers.

As for the 3 class case, the MAP classifier had the lowest error, followed by MICD, MED, 5NN, and then NN. The reason for this is because the MAP classifier considers the most information (the probabilities and covariance matrices of each class), followed by the MICD classifier which considers slightly less information (just the covariance matrices), and finally the MED classifier which considers the least amount of information (just the means of each class). Similar to the two class case, the NN and 5NN error rates were higher because they were more sensitive to noise and outliers, but 5NN is slightly less sensitive to noise because the prototype definition considers the sample mean of the 5 nearest points, as opposed to all points. If the generated clusters were more long, thin, and tendril-like, as opposed to Gaussian in nature, then the NN classifiers would likely perform better.