Statistical Pattern Recognition - Option 4

Sam Leese

The MATLAB code for the first two parts of the coursework can be found in *kmeansadapt.m*, and the small function that performs the centering for part 3 is *centering.m*. There is also a script in which the PCA and k-means are performed in the opposite order which was mentioned in the extension, called *pcafirst.m*.

1. **K-Means clustering:**
   After importing the data set, 3 initial guesses are made for the cluster centres between the minimum and maximum values of each feature of the data. This was done using the rand() function in MATLAB. The iris data used has four features so each cluster centre is a 1x4 array, and each 1x4 row of the data corresponds to a point. Now the algorithm runs through each row of the data, computes the euclidean distance to each of the centres and then assigns the point to the closest cluster centre. The euclidean distance was found using the following method.

   $$V = datapoint(i) - clustercentre(j)$$

   $$dist = sqrt(V \times V^T)$$

   Once this is done for all the points, the next step is to find the new cluster centres by finding the mean of each clustered group of data points. If these new cluster centres are the same as the previous ones the program goes back to the beginning until there is no change in the position of the centres. However if due to a poor initial centre guess one or more centres has no points assigned to it, the program starts again with reinitialised centres.
   The initialisation of the centres can have quite an impact on the characteristics of the clustering and how well it finally clusters the points, as they are random initial guesses. Some examples of the different results can be found in part 2.
   (Extension): As the correct classifications are available, the performance of the clustering algorithm can be analysed. The score of the classifier can be found relatively quickly by finding the confusion matrix of the actual classes (50 of each type of plant) and the classifier results, and looking at the diagonal entries. As the initial cluster centres are random, the classes will more than likely not be labelled the same, so the largest entries may not be in the diagonal.

It can be assumed that the total number of correctly clustered points will be the maximum value for the sum of the diagonals for all different permutations of the row order. The score is then found by dividing this number by the total number of points to find a proportion of the points that were correctly classified. After running the program 200 times, there were 4 different scores achieved, which would depend on where the cluster centres are initialised. These scores were: 0.5133, 0.5600, 0.8867 and 0.8933. It shows how there are certain cases where the initial location of the means will make a large difference if perhaps depending on which of the other two centres the third one is. Then there are smaller discrepancies in each of the two different clustering results.

2. **PCA:**

PCA aims to process the data into a form that is easier to visualise making it useful for finding patterns in high dimensional data. This is done by reducing the number of features of the data down into usually one or two of the principal features, making it possible to plot the data. These features are identified by first finding the covariance matrix of the data. This was done using the method of multiplying the data matrix by the transpose of itself rather than using the in-built MATLAB command. So for a matrix of data X, the covariance will be:

$$\text{Covariance matrix} = \frac{1}{N} X^T X$$

where N is the number of data entries. This is used because the MATLAB function centers the covariance matrix as you are subtracting the means of all the columns during the usual process. The covariance matrix can then be used to identify the features with the largest variance. These features can be easily found by analysing the eigenvalues and vectors of the covariance matrix. The eigenvectors with the largest eigenvalues will give the principal components. In this case the two largest ones are used. Now arranging the two corresponding eigenvectors as a 4x2 matrix, gives the projection matrix. By multiplying the transpose of this matrix by the transpose of the data set, we get the data in terms of the two principal features. The same can be done to the cluster centres. Now plotting the new data points after PCA with their cluster assignment along with the cluster centres the following graphs can be found:

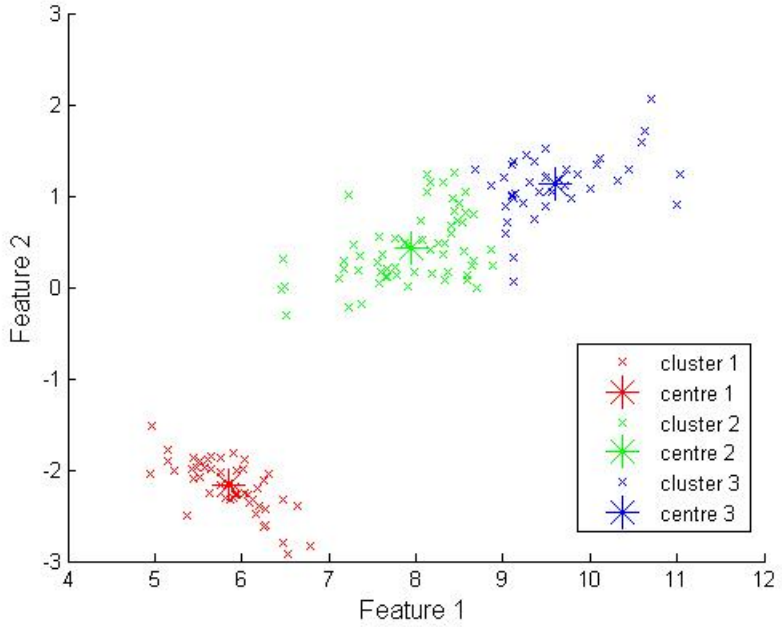

Figure 1: Clustering with PCA, without centering graph 1

Figure 1 is the most common result produced by the program, with one little discrepancy between a single point that in this case switches between cluster 1 and 2, however the overall result remains mostly identical.

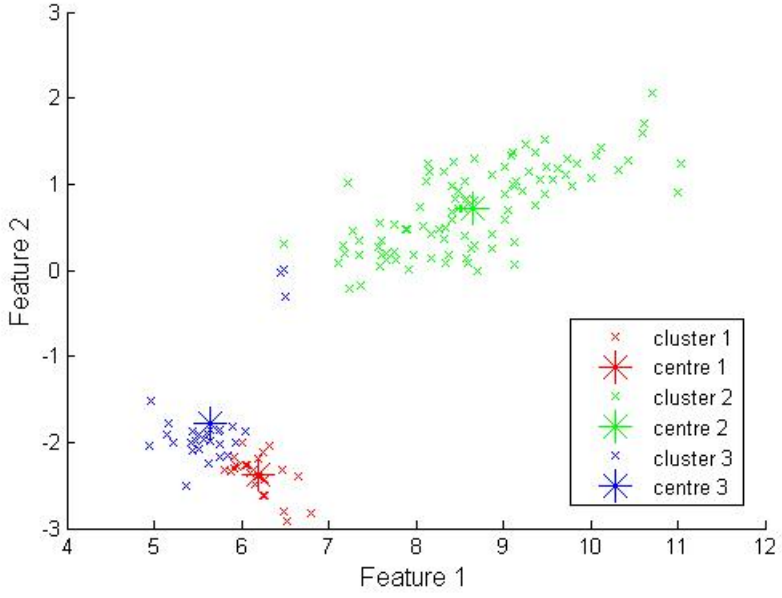

Figure 2: Clustering with PCA, without centering graph 2

Figure 2 shows an alternative result that is sometimes found by the program. This is an example of how the initial guesses for the cluster centres can influence the final result. It most likely occurs if, in this case, centre 1 and 3 are initialised close to each other in comparison to centre 2, especially if they share similar initial guesses for the principal feature values.

3. **Centering:**

   The centering of the data simply is the process of subtracting the mean of each feature for the entire data from the corresponding feature of each data point. This will give a data set that is centred around zero, and is performed at the beginning of the program.

   Now plotting the new data points after PCA with their cluster assignment and having been centred along with the cluster centres the following graphs can be found:

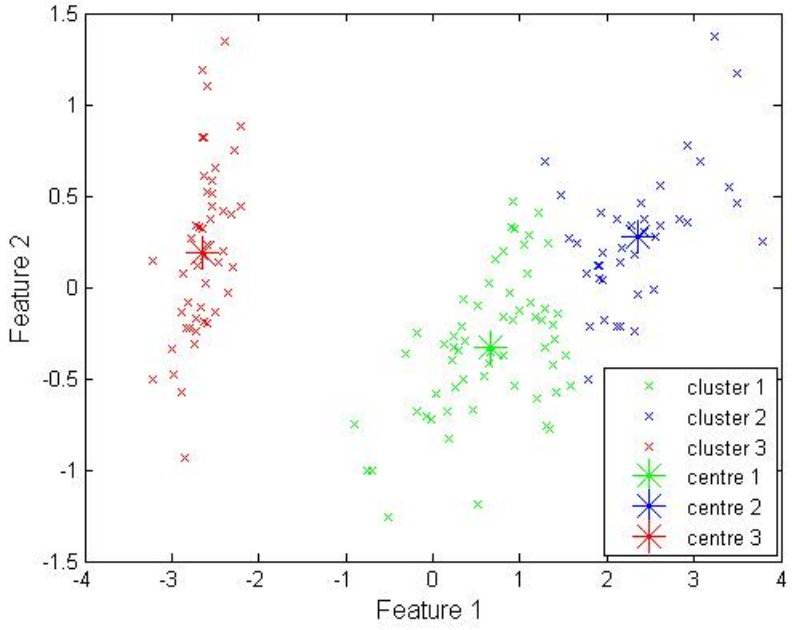

Figure 3: Clustering with PCA, with centering

There is a clear difference in the qualities of this graph compared to without centering. This is because the centering

4. **Extension:**

   As an extension to the coursework description, the program was adapted to cluster variations of data and with different numbers of clusters. Taking data on wine as an example, found at *http://archive.ics.uci.edu/ml/datasets/Wine*. The data deals with data containing 13 features and 3 classes of wine. Using the exact same process as before including the normalisation, the following plot can be obtained:

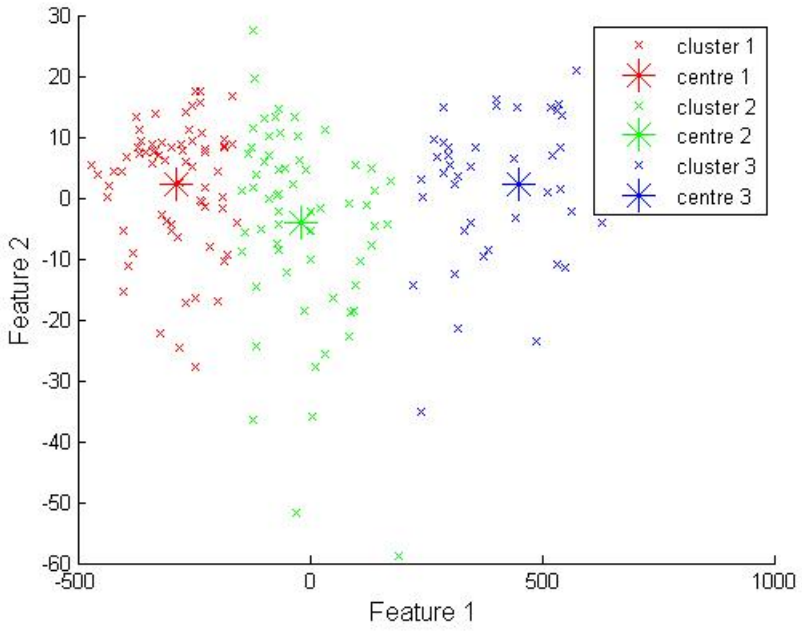

Figure 4: Clustering with PCA, with centering for data set on wine

The score achieved for the clustering in this particular case was 0.7022, which isn't quite as good as for the previous set of data. However this was the best case in terms of score. This is most likely due to the fact that the clustering is dealing with more features, and possibly as the differences between features for the types of wine aren't that great. As a slightly alternative approach, the method was swapped around slightly to see if it made a difference. Specifically, the PCA part was instead carried out first, and then perform k-means on the principal components of the data. However this yielded very similar results to the original case.