Data Clustering and Classification Methods

James O'Reilly and Adam Pluck

1 Introduction

Given a set of unlabelled training and test data, we were tasked with selecting the appropriate features and identifying the classes before training and applying a classifier. The aim of this report is to illustrate in detail the approach we took to each of these tasks, from feature selection to eventually applying a classifier to the test data. Furthermore, it attempts explain to the reader the underlying theory and mathematics behind the clustering and classification methods used while expanding upon the topics we thought to be both interesting and relevant to the coursework.

2 Feature Selection

Feature selection, also known as variable selection or attribute selection, is the selection of a subset of features in the data set which are most relevant for the construction of a predictive model. Feature selection removes redundant features that do not aid with the accuracy of the model. Feature selection is performed in order to improve the performance of the predictors, provide faster predictors, and to provide a simpler model that is easier to understand. [2]

Our data set consists of a set of instances which are described by the five feature vectors. The aim was to choose the feature vectors which best divide the set of instances into subsets, such that each subset is associated with a class. The attribute values for each pair of features was plotted in a scatter-plot matrix in order to discern which pair of features best separated the classes. Originally, it was unclear to us which features best separated the classes. By visual inspection there were clearly three pairs of features which separated the data into two classes while another pair of features separated the data into three classes.

(a) Plot A (b) Plot B (c) Plot C (d) Plot D

Ultimately, the pair of features that separated the data into three classes was chosen for a number of reasons:
1. Splitting the data into three classes is better than two classes as it classifies the data more specifically.
2. By visual inspection we could clearly see this pair of features yielded the most compact clusters.
3. The clusters were nucleated, while the clusters produced by the other pairs of features varied substantially along an axis.
4. We observed few outliers in Plot A relative to the other plots. Furthermore, it is important to note that K-means clustering uses Euclidean distance as a metric, which tends to be sensitive to outliers. Knowing that K-means would be implemented later on, it was decided it was best to pick the data with the least number of obvious outliers.
5. Plot A has a similar number of observations in each cluster, while the clusters in the other plots have varying cardinalities. It was clear that having a uniform prior probability for the classes would be very useful going forward, this will be expanded upon when we discuss Maximum-Likelihood classification.

As each instance in the data set was not yet classified, feature selection was performed by visually inspection. It is important to note that had each instance already been classified, feature selection algorithms such as Recursive Feature Elimination (RFE)[3] could have been used to determine which features were most valuable to the predictive model and which features were effectively redundant.

3 Identifying the classes

The data used to train our model did not contain labelled instances. As a result of this, unsupervised learning must be used to draw inferences from the data. One frequently-used unsupervised learning method is cluster analysis, which highlights patterns or groupings in the data. The clusters are modelled using some measure of similarity, defined by metrics such as Euclidean distance. A number of different clustering algorithms could have been used to identify the classes. The K-means clustering algorithm was chosen for this coursework.

The class labels for the training data were derived automatically using the K-means algorithm with the number of clusters set to three, as it had previously been established there were three classes. The K-means algorithm clusters the data as follows: Given an initial set of k centroids (means), the algorithm iterates through two steps:
Step 1: Assignment step: Calculate the least squared Euclidean Distance from an observation to each centroid and assign that observation to the cluster whose centroid is nearest. Repeat this for each observation.
Step 2: Calculate means: Set the new centroids to be the mean of the observations in the new cluster.
K-means terminates when the assignments no longer change.[5] Upon termination, K-means returns the class label for each observation, the centroids (means) of the clusters and the inertia: the sum of squared distances of samples to their closest cluster centre. Observations that share the same class label are then placed into separate arrays which are the classes. The centroids returned by K-means will be used as a nearest neighbour classifier, which we will discuss later. It is important to note that each class produced by the k-means classification contained the exact same number of observations and so has uniform prior probability. This will be relevant later when we discuss Maximum Likelihood Estimation.

Although K-means classified our data correctly, it is important that one considers why K-means was appropriate in this case, and the possible alternative clustering algorithms that could have been used to classify the data. Strictly speaking, the K-means clustering algorithm doesn't operate under any assumptions about the nature of the data, however it is well suited for some data sets and rather poorly suited to others. K-means tends to classify the data incorrectly if the clusters vary greatly in cardinality or if the data is in certain non-spherical arrangements.[1] K-means is an appropriate clustering algorithm for our training data as the data consists of equally-sized clusters which are roughly spherical in shape.

Another clustering algorithm which could have potentially been used to classify the training data is k-medoids clustering.[4] K-medoids clustering is a variant of the k-means algorithm wherein the centroids are restricted to members of the data set. Unlike the standard K-means algorithm, K-medoids clustering is not restricted to Euclidean distance (which is sensitive to outliers) and instead uses Manhattan distance. Thus, K-medoids could be appropriate if the data had many outliers.

4 Nearest-centroid classification

Once the training data has been classified using K-means, the centroids found by K-means can now be used as a nearest-centroid classifier. This classifier is then applied to the test data. Firstly, the Euclidean distance from each observation to each centroid is must be calculated. This is done using the "cdist" function with "p = 2". Setting "p = 2" ensures that Euclidean distance is used as a metric:

$$D(\boldsymbol{u}, \boldsymbol{v}) = \sqrt{(\boldsymbol{u} - \boldsymbol{v})^T (\boldsymbol{u} - \boldsymbol{v})}$$

Once the distance from an observation to each centroid has been calculated, the observation is assigned to whichever centroid it is nearest to. As such, each observation is assigned a class. A Voronoi diagram can can now be plotted which partitions the space into Voronoi cells. The Voronoi diagram is effectively a linear decision boundary where the line segments of the diagram are the points in the plane that are equidistant to the two nearest centroids.

Figure 1 (below) clearly shows that the Nearest Centroid Classifier classified the test data correctly, splitting the data properly into three classes. The Voronoi diagram shows the points which are equidistant to each centroid and partitions the plot into the three classes as it should.

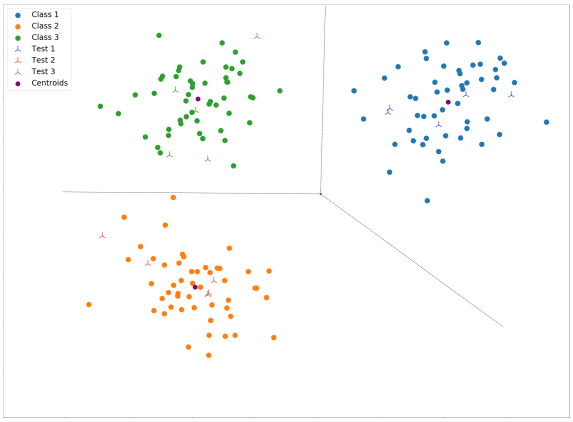

Figure 1: Locally Optimal Clustering

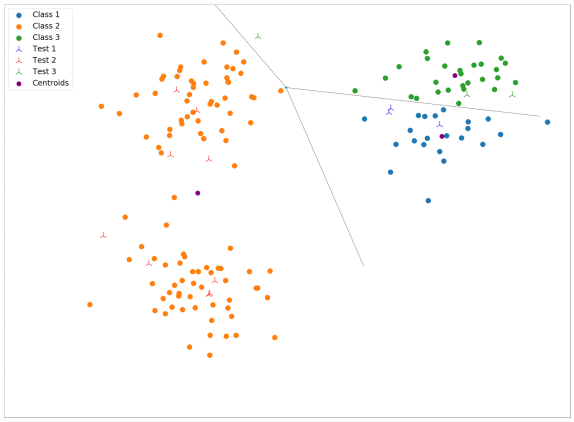

Figure 2: Sub-optimal Clustering

While the nearest Centroid Classifier seemingly classified our data correctly, it is important to note that K-means is a heuristic algorithm, and therefore there is no guarantee that it will converge to the global optimum. The result is dependant on the initialisation parameters such as the initial centroid seeds, the number of iterations the algorithm performs and the number of seeds. If one could carefully select the initialisation parameters, one could avoid the poor clusterings often generated by the standard K-means algorithm. The k-means++ algorithm addresses this issue by giving a procedure to select improved cluster centres before the standard K-means algorithm is run. Our implementation of K-means makes use of K-means++ algorithm.

A sub-optimal clustering of k-means is a clustering which clearly classifies the data incorrectly. In order to induce a sub-optimal clustering one must change the initialisation parameters of the k-means function. There are many ways of doing this. Originally, our approach was to run k-means with differently initialised seeds. The centroid seeds were chosen in such a way as to induce a sub-optimal clustering. While this approach did in fact generate a sub-optimal clustering, we decided that this approach was inferior as poor seeds could only be reliably chosen with some prior knowledge about the nature of the data. Another possible approach was to set the initialisation parameter to "random" so that the seeds were no longer generated by the K-means++ algorithm.

We decided instead to limit the number of times the k-means algorithm will be run with different centroid seeds. The algorithm was ran 10000 times without K-means++ and with a limited number of seeds. The largest inertia generated by these 10000 runs was then selected and the labels corresponding to the largest inertia were then used to classify the data. It is clear in Figure 2 (above) that this resulted in a sub-optimal classification.

5 Maximum-likelihood classification

Having stated previously, the classes have uniform prior probability. This uniformity gives rise to a special case when attempting maximum a posteriori estimation, known as maximum-likelihood estimation. This is the "procedure of finding the value of one or more parameters for a given statistic which makes the known likelihood distribution a maximum"[6]. Essentially, one needs to find a set of values for which the distribution best fits the given data set.
It is fair to assume that the "known likelihood distribution" for each of the classes is the Bivariate Normal distribution. This means that there are two parameters for the distribution: mean and variance. In solving the maximising problem via the manipulation of the log-likelihood, one will arrive at the following formulae for the maximum-likelihood estimators:

$$\hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i \qquad\qquad \hat{\boldsymbol{\sigma}}^2 = \frac{1}{n}\sum_{i=1}^{n} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})^T$$

This is trivially just the sample mean and the covariance matrix respectively. So for each of the classes we can easily calculate the maximum-likelihood estimators. In the case of our data there are three different

classes, and therefore three different estimated distributions with 6 estimators in total.

What was the motivation for finding the estimated distribution? By finding an estimated distribution for each of the different classes, we now have the ability to estimate probabilities for the test data. This is particularly important as Bayes' theorem can now be used to derive a decision rule. Again, as the data is of uniform prior probability, it can be seen that:

$$P(\omega|x) = argmax_{\omega} P(x|\omega)$$

Observing the above formula, the class that maximises the probability on the right-hand side is most likely which ever class that instance belongs to. This is our decision rule: choose the class that gives the largest probability value for that given instance. This decision rule gives flexibility as it can deal with the case of having more than 2 distinct classes. Pairwise likelihood ratios are still very useful as a decision rule, but only handle a decision between two classes. It is understandable to want to have a level of confidence in the decision rule. By implementing a 95% confidence interval, it can be said that a given instance has a probability of 0.95 of belonging to that particular class. This interval will of course again be an ellipse.

Achieving this 95% confidence interval can be done in many different ways, but breaking this problem down into two parts helps: first, plot a rotated ellipse and then find the appropriate principle axes lengths for the required confidence interval. The estimators for the distributions are the sample mean and covariance matrix of each individual class. As a result of this, the required ellipses' centres (centroid of the data) will be the sample means of the classes. The principle axes and orientation of the ellipse (moment of inertia) can also now be derived from the covariance matrices. The principle axes are two orthogonal vectors intersecting at the centroid with the larger going to the furthest points of the ellipse. These orthogonal vectors are the eigenvalues of the covariance matrix. The orientation of the ellipse is the calculated angle between the horizontal and the eigenvector with the largest associated eigenvalue. These results directly follow from a manipulation of the covariance matrix. Having calculated the required vectors and values, an ellipse at a general confidence interval with the correct orientation can be drawn.

Now the required lengths of the principle axes are needed to show the correct confidence interval. This can be achieved through analysing the equation of the required confidence ellipse. The equation of the required ellipse is:

$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 = z$$

Both x and y are normally distributed and independent, thus using the fact that the definition of the Chi-squared distribution is the sum of squared normally distributed variables, we have that $z \sim \chi^2$. The summation is only over two random variables and so we have 2 degrees of freedom. As we are picking a 95% confidence interval, we are concerned with finding a value of z for which 95% of the data is contained in the ellipse. By reducing this to a chi-squared distribution with 2 degrees of freedom, the problem in essence boils down to finding a z for which the probability of choosing a value less than z is 0.95 (0.95 from the confidence interval).

$$P(x < z) = 0.95\ , \qquad where \qquad z \sim \chi^2_2$$

This means that z is equal to 5.99. We therefore have that the required principle axes lengths are $\sqrt{5.99\lambda_1}$, $\sqrt{5.99\lambda_2}$ where $\lambda_{1,2}$ are the eigenvalues. This is then easily plot-able via parametric coordinates in the form $\sqrt{5.99\lambda_1}\cos(\theta)$ + x-mean , $\sqrt{5.99\lambda_2}\sin(\theta)$ + y-mean. First, plot the ellipse with the required principle axes lengths for a 95% confidence region. Then rotate the ellipse via matrix multiplication with the rotation matrix and angle previously calculated. Finally, translate the ellipse by the sample mean so that the ellipse is centred about the correct centroid.

Although there are three classes, it is still very useful to analyse the pairwise ratio likelihoods between pairs of classes. The pairwise ratio is the ratio of probabilities that an instance belongs to a chosen class:

$$\frac{P(v|\omega_i)}{P(v|\omega_j)} = a$$

The points at which this ratio is equal to one (i.e. an instance is equally likely to belong to either of the two classes) form a curve. This curve is the decision boundary. A point contained in the decision boundary contour has a pairwise ratio of less than one, and thus more likely to belong to that class. A point outside of the contour has a ratio greater than one, hence is more likely to be part of the other class.

It can be seen that the decision boundaries are hyperbolic. This is due to the covariances of the distributions being unequal. A large deviation between the covariance matrices would cause the hyperbolic decision boundary to have a more severe arc.

Comparing this to the nearest centroid classifier, there is a clear difference in shape. The nearest centroid classifier has a linear decision boundary, while the Decision boundary for the maximum likelihood classifier is hyperbolic. Although there is a difference, the decision boundaries are somewhat similar. This is particularly apparent when looking at the training and test data. For both data sets there is total agreement as to which classes all the instances belong, irrespective of which classifier is used.

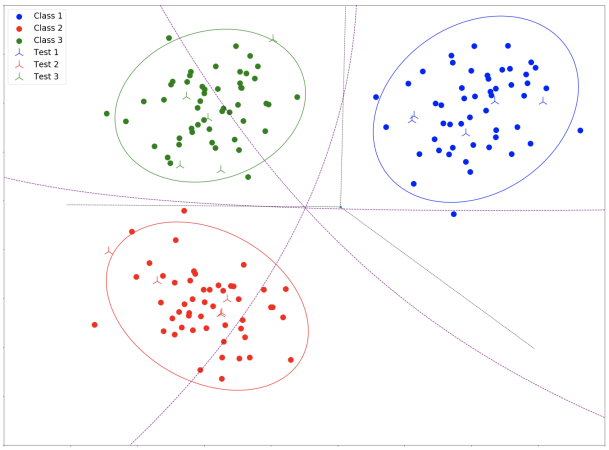

Figure 3: Data Set ap16894

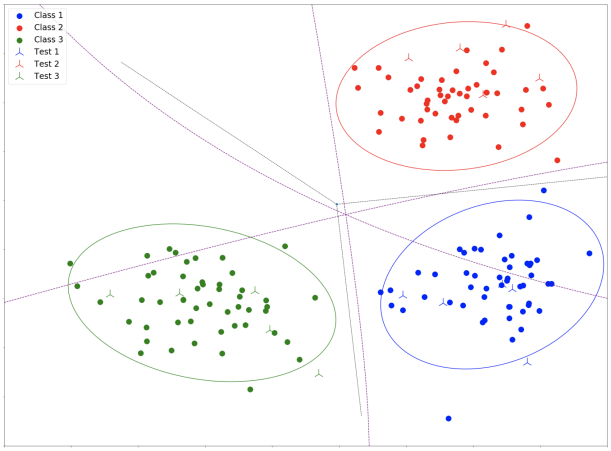

Figure 4: Data Set jo16827

In order to align the maximum likelihood decision boundary with nearest centroid decision boundary, the Mahalanobis distance (distance from a point to the mean of the distribution) must equal the Euclidean distance for each class. Although we are concerned about the Mahalanobis distance for points where the ratio is equal to 1, we can solve the problem more generally. As both the nearest centroid classifier and maximum likelihood classifier share the same centroids, namely the sample mean of each class, we need only find the parameters for which the equation below holds. i.e, where the LHS is the formula for Mahalanobis distance and RHS is the Euclidean distance:

$$D_m(u, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \sqrt{(u-\boldsymbol{\mu})^T \boldsymbol{\sigma}^{-1} (u-\boldsymbol{\mu})} = \sqrt{(u-\boldsymbol{\mu})^T (u-\boldsymbol{\mu})} = D_e(u, \boldsymbol{\mu})$$

It is clear that, in order to make the two distances equal one must find a covariance matrix ($\boldsymbol{\sigma}^2$) whose inverse is the identity matrix. This is trivially the identity matrix. And so setting the covariance matrix to the identity matrix causes the decision boundaries of the Maximum Likelihood classifier to be the same as the decision boundaries for nearest centroid. This can be seen clearly in Figure 5.

If one class was twice as likely as another, it is should in theory shift its decision bounds towards the favoured class (Figure 6). This can be done by multiplying the probability of an instance being part of that class by two.

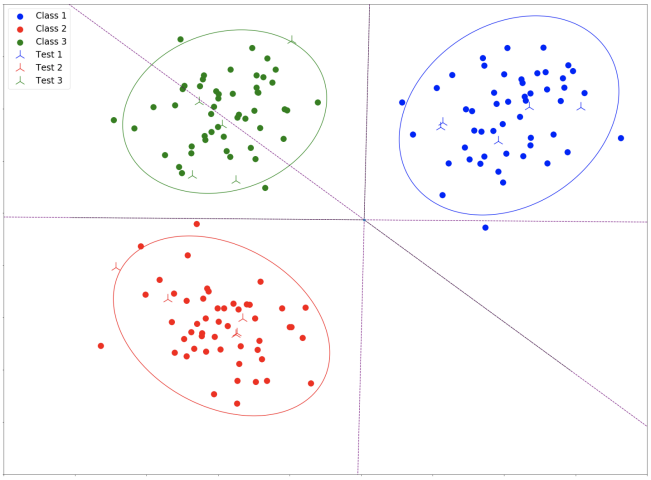

Figure 5: Matching decision boundaries

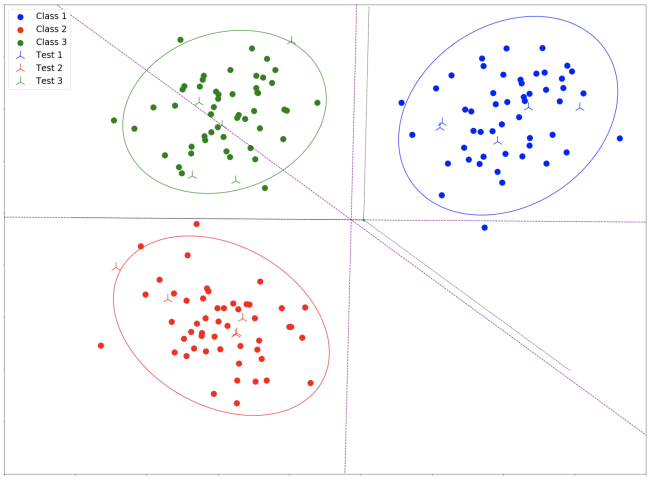

Figure 6: Class 1 twice as likely

6 Evaluation of results and classification methods

Both the maximum likelihood and nearest centroid classifiers produce flawless results for both data sets. When observing the test observations for data ap16894 and data jo16827, we can see that not only do they assign the same class to each of the observations, the assigned class is correct for every single observation. There were no false classifications and 150 correct classifications leading to an 100% accuracy for both data sets. One would of course like to test the classifiers with a larger test set, however we can still say with confidence that both classifiers performed exceptionally.

Looking at the 95% confidence interval ellipses, we can see that 94% of ap16894 training data falls within the ellipse. For jo16827, we have that 94.6% of the data falls within the ellipse. The amount of data of course should be 95%, so although we do have highly accurate estimating distributions for both data sets, we have that ap16894 training data is not perfectly estimated.

There are many ways to better analyse the effectiveness of the two classifiers. As we do not have the use case for these classifiers and what data we would ultimately be classifying, we cannot fully assess the cost of an incorrect classification. If we did, however, understand the context of the classification, our judgement on the success of a classifier could change. If, for example, we were classifying cancer test results as positive or negative, one would prefer a large amount of false positives to a large amount of false negatives.

References

[1] K-means clustering is not a free lunch. https://www.r-bloggers.com/k-means-clustering-is-not-a-free-lunch/. Accessed: 2018-18-3.

[2] Guyon, I., and Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res. 3* (Mar. 2003), 1157–1182.

[3] Jain, A. K., Murty, M. N., and Flynn, P. J. Data clustering: A review. *ACM Comput. Surv. 31*, 3 (Sept. 1999), 264–323.

[4] Kaufmann, L., and Rousseeuw, P. Clustering by means of medoids. 405–416.

[5] MacQueen, J. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics* (Berkeley, Calif., 1967), University of California Press, pp. 281–297.

[6] Weisstein, E. W. "Maximum Likelihood." From MathWorld–A Wolfram Web Resource. Visited on 18/03/118.