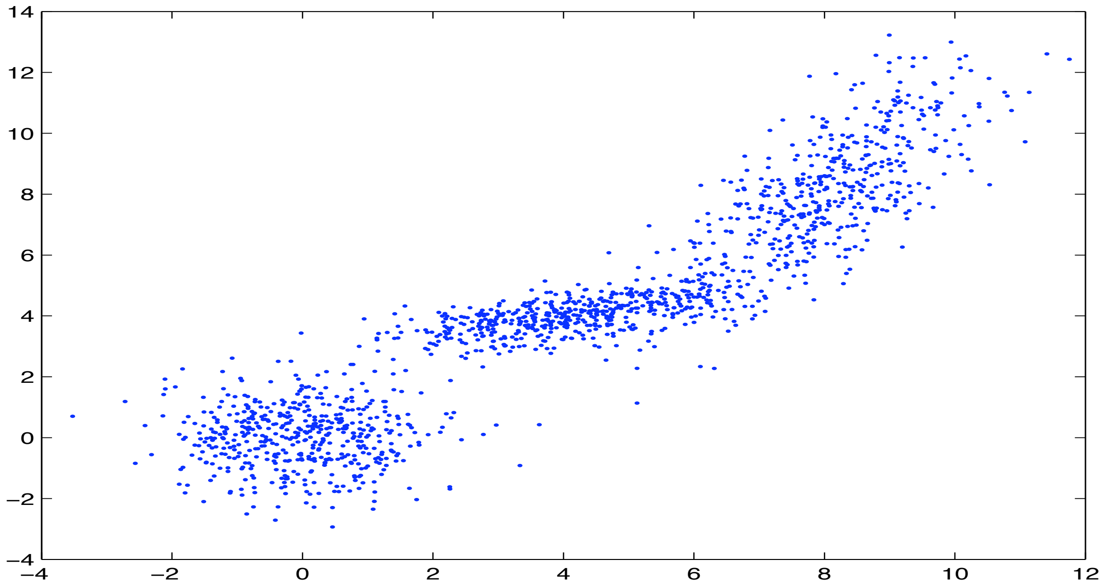
Clustering

Scotland C. Leman
Department of Statistics
Virginia Tech
leman@vt.edu

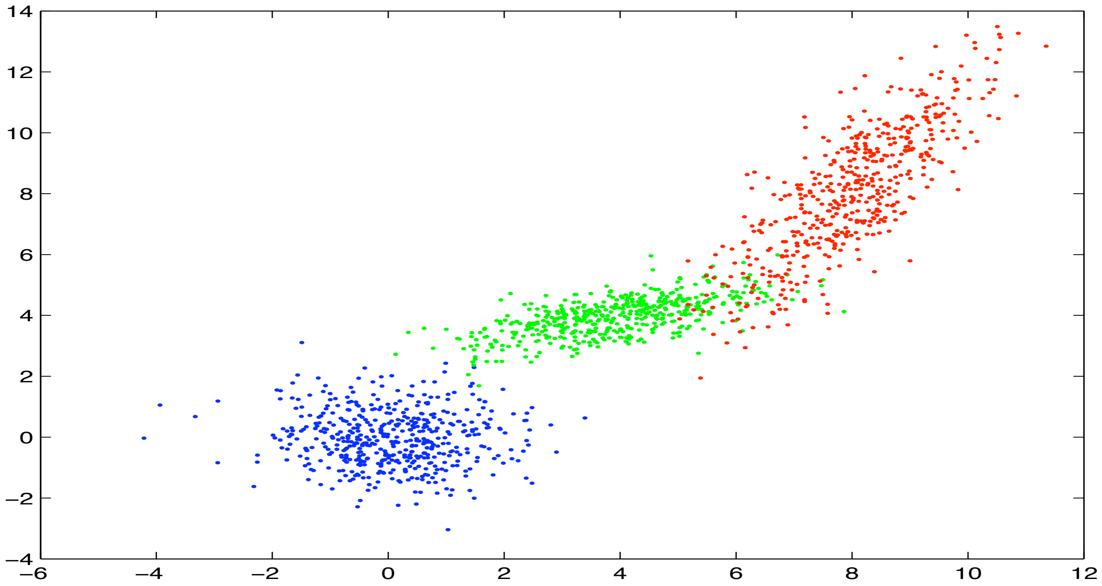
Oct. 13, 2015

Brief Introduction

A cluster analysis seeks to classify points in some space. Consider the example



Mixture of Gaussians



This was generated under the 3 normal distributions

$$N(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}), N(\begin{pmatrix} 4 \\ 4 \end{pmatrix}, \begin{pmatrix} .5 & .6 \\ .6 & 5 \end{pmatrix}), N(\begin{pmatrix} 8 \\ 8 \end{pmatrix}, \begin{pmatrix} .1 & .9 \\ .9 & 10 \end{pmatrix})$$

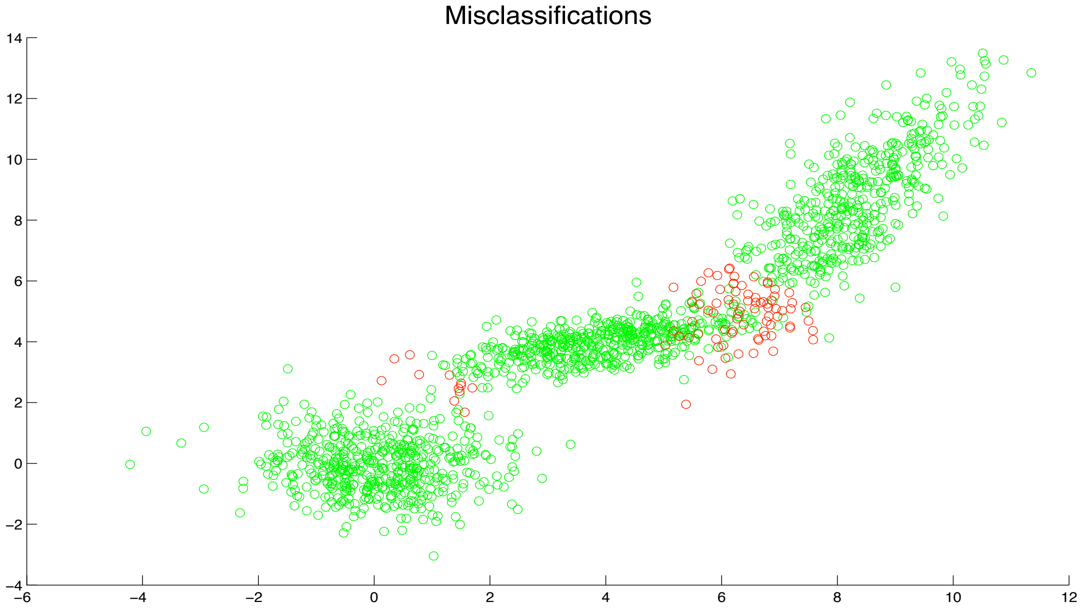
K Means

The K means clustering method is probably the simplest clustering method. Here is the algorithm

- Select the number of clusters (K).
- Initialize the centers for the K clusters.
- For each data point, identify the cluster center it is closest to and classify the point to that cluster.
- Compute the new means for each cluster.
- Repeat steps 3 and 4 until the centroids converge.

Classification Rate

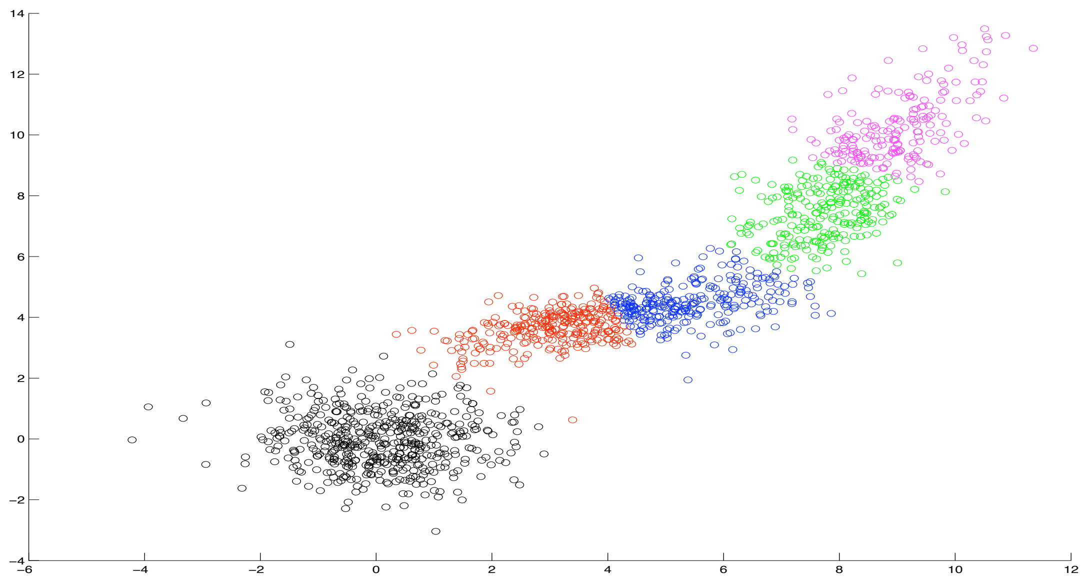
For our example of 3 Gaussian mixture samples, we ran the K-means algorithm with $K = 3$ (naturally). The correctly identified points (green) and misclassified points (red) are shown.



The misclassification rate is 7% for this example.

More Clusters?

The K-means framework doesn't really have a natural way of assessing how many clusters to use. This could be natural if you want to classify into a predetermined number of groups. However in general this is an open problem.



K Nearest Neighbors

The KNN scheme is a strict classification scheme based on memory from a training set. Take for example our original mixture of 3 distribution. Since we simulated it, we know everything about it (i.e. we know which class each point belongs to).

Now let's say, someone generates a new point x_0 but doesn't tell you the class. How do you decide which class it came from?

One possibility is to consider the closest (pick your favorite measure) point in the training sample to x_0 and classify it according to its nearest neighbor.

K Nearest Neighbors

The KNN scheme is a strict classification scheme based on memory from a training set. Take for example our original mixture of 3 distribution. Since we simulated it, we know everything about it (i.e. we know which class each point belongs to).

Now let's say, someone generates a new point x_0 but doesn't tell you the class. How do you decide which class it came from?

One possibility is to consider the closest (pick your favorite measure) point in the training sample to x_0 and classify it according to its nearest neighbor.

One might also considering its K Nearest Neighbors, and classify it according to a majority wins technique. This is the KNN scheme.

Hierarchical Clustering

With K means clustering, the user was required to specify the number of clusters and perhaps starting centers. In contrast, hierarchical clustering methods do not require such specifications. Instead, the user must assign a measure of dissimilarity between groups of observations based on pairwise consideration of the groups.

As the name suggests, clusters have a hierarchical representation. Clusters at each level of the hierarchy are produced by merging clusters at the next level down. At the highest level, there is exactly 1 cluster, and at the bottom level there are N clusters (one for each data point)

Hierarchical Clustering

There are two basic strategies for constructing a hierarchical set of clusters.

- Agglomerative (bottom up).
- Divisive (top down).

Agglomerative methods start with every point in its own cluster and merges two clusters to form one cluster based on closeness (least dissimilar). Most of our attention will be on this case.

Divisive methods start with one large cluster, then recursively split the clusters based on a rule. (Think CART).

Merging Rules

In agglomerative clustering, three very common techniques exist for merging clusters. They are:

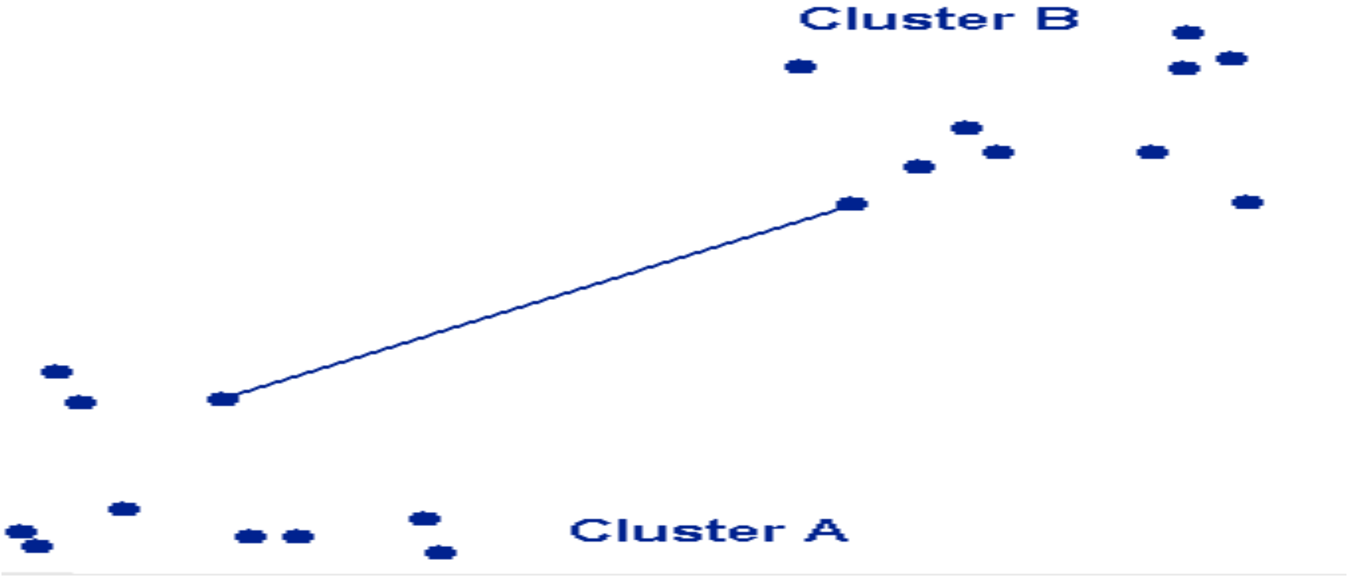
- Single Linkage
- Complete Linkage
- Group Average

Single Linkage

Takes the inter-group dissimilarity to be that of the closest pair:

$$d_{SL}(G,H) = \min_{i \in G; j \in H} d_{ij}.$$

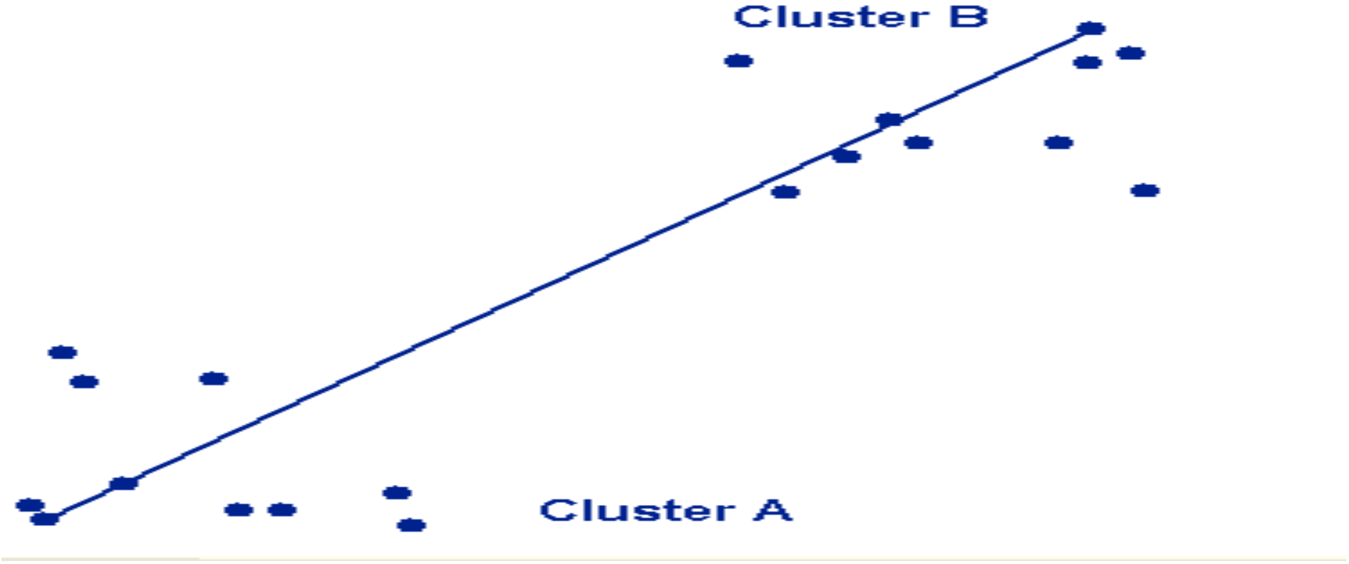
This is often called the nearest neighbor technique.



Complete Linkage

Takes the inter-group dissimilarity to be that of the furthest pair:

$$d_{CL}(G,H) = \max_{i \in G; j \in H} d_{ij}.$$

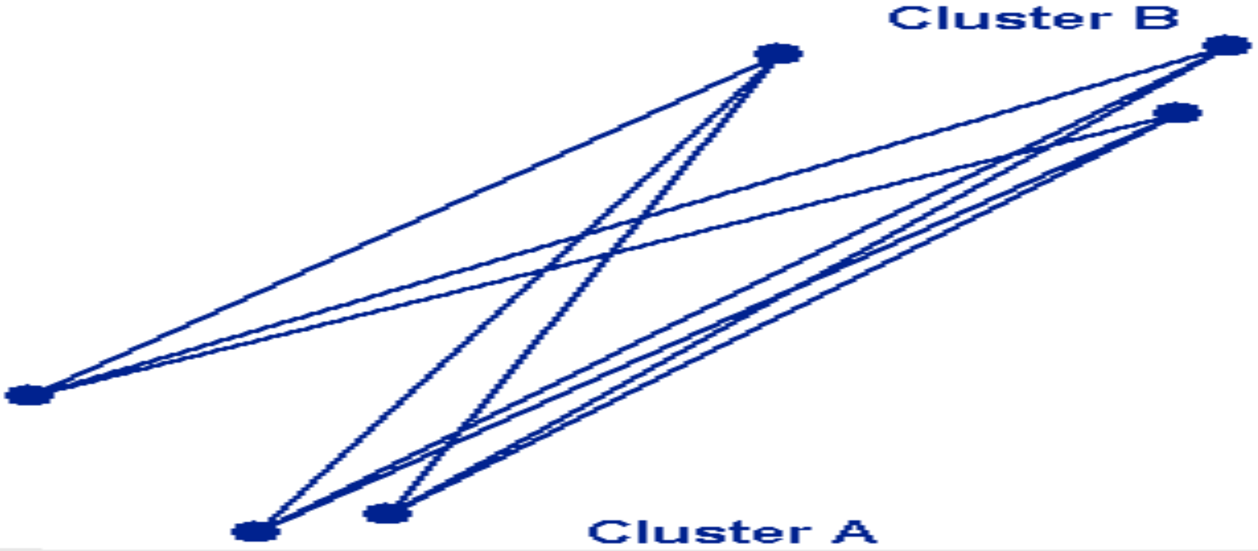


Group Average

Uses the average dissimilarity between the groups:

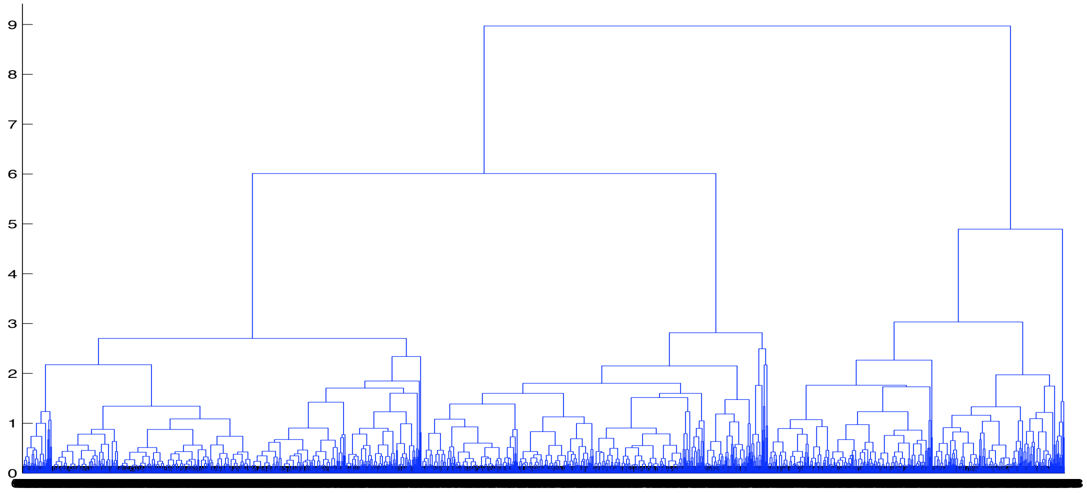
$$d_{GA}(G,H) = \frac{1}{N_G N_H} \sum_{i \in G} \sum_{j \in H} d_{ij}$$

where N_k is the number of points in group k.



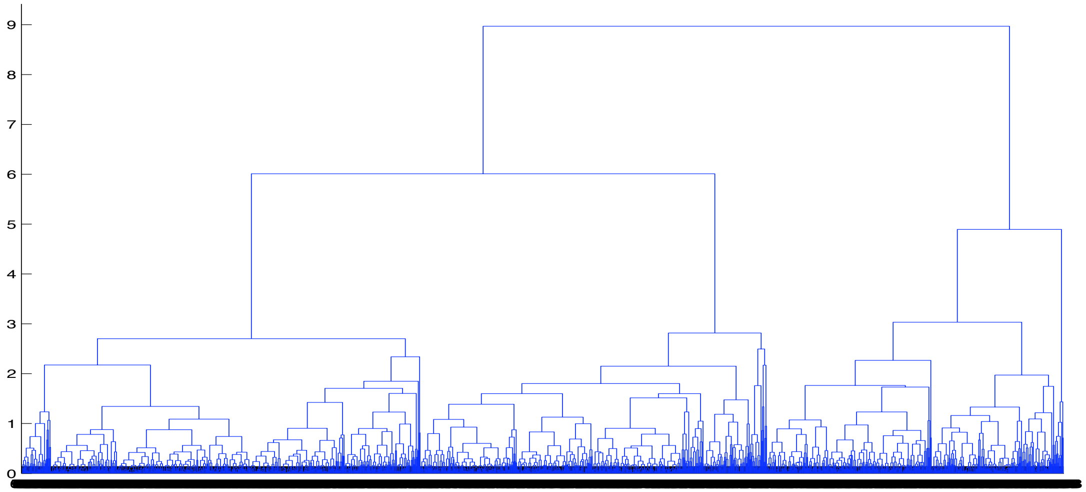
Dendrograms

Let us go back to our original data a run an agglomerative clustering scheme with Euclidean distances and Group Average linkage. From this we produce a representation of the clusters merging and the distances at which they are merged. This is called a dendrogram.



Dendrograms

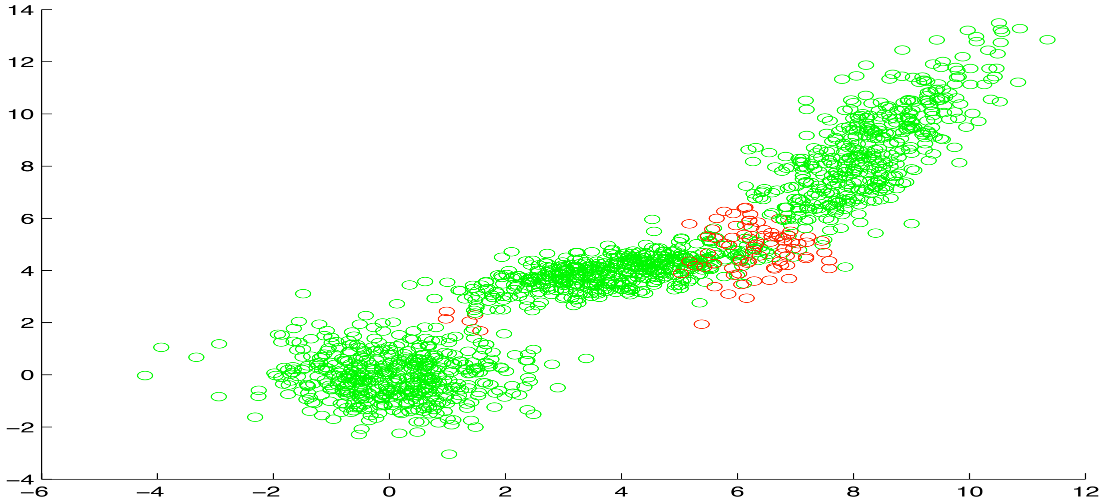
Let us go back to our original data a run an agglomerative clustering scheme with Euclidean distances and Group Average linkage. From this we produce a representation of the clusters merging and the distances at which they are merged. This is called a dendrogram.



It appears that there are 3 main clusters.

Truncating the clusters

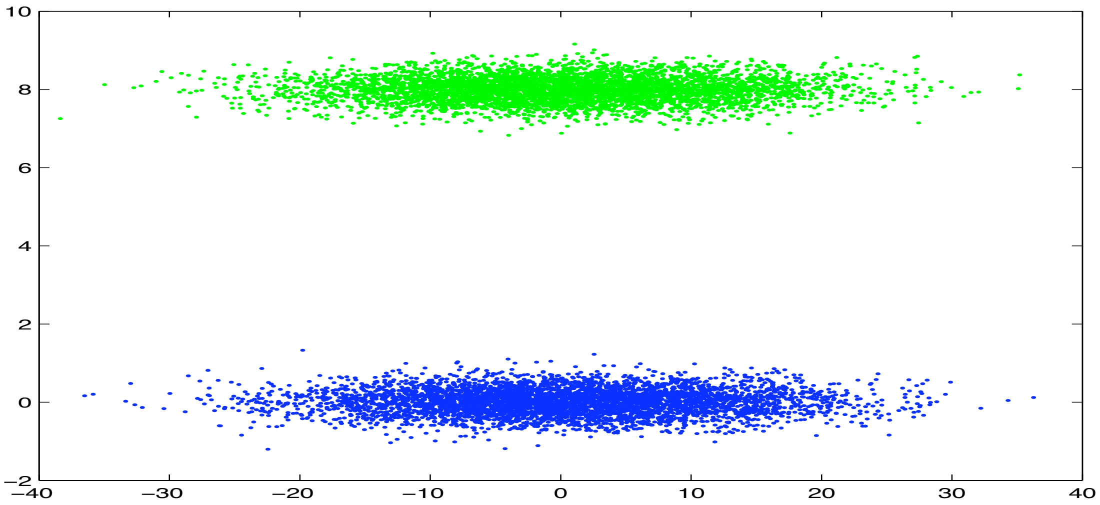
From the dendrogram, we observe 3 main clusters, so we should only allow for three clusters and rerun the scheme. After doing this and observing the correctly classified and misclassified points as



The misclassification rate is 5% which is a bit better than the K-Means case.

Problems With Complete Linkage

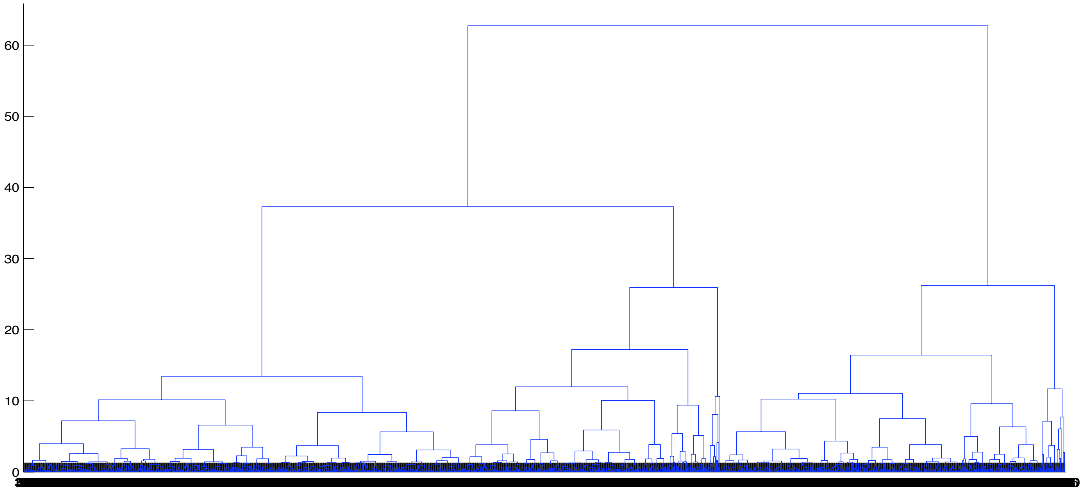
Consider the data set generated from 2 normal distributions



Can anyone see why complete linkage is a really bad idea here?

Problems With Complete Linkage

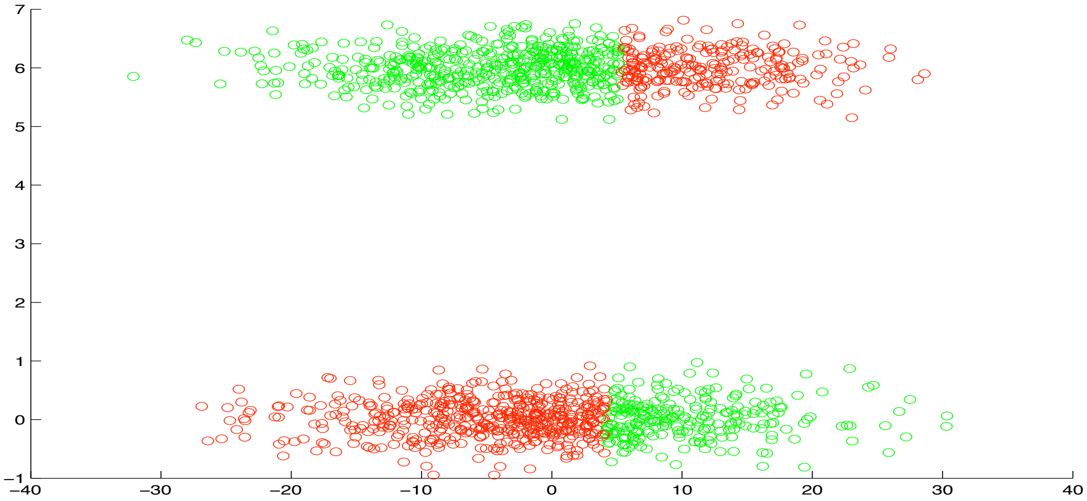
The dendrogram for the complete linkage case follows as:



Let us proceed by assuming 2 clusters.

Problems With Complete Linkage

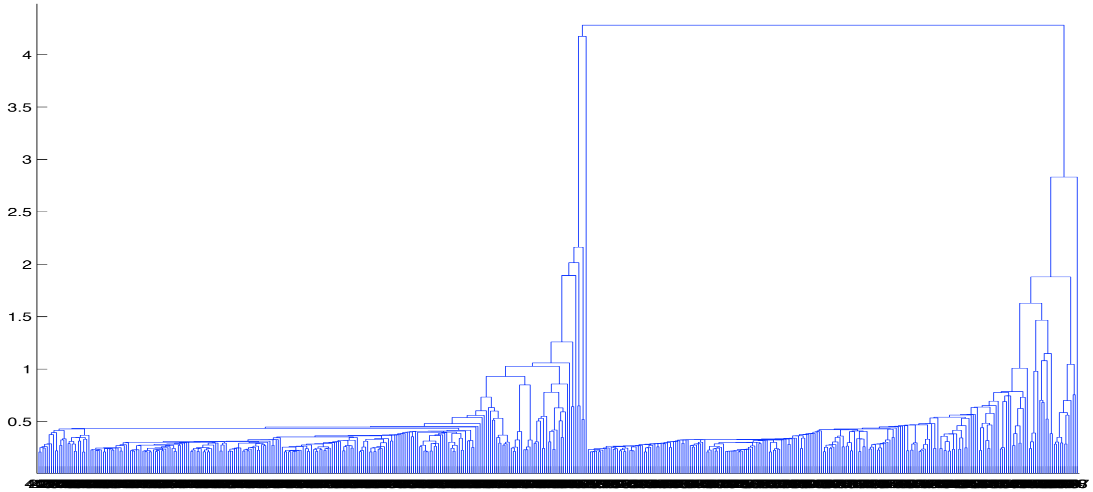
The misclassification plot looks like:



The misclassification rate is 50%!!! Could you have predicted that?

Problems With Complete Linkage

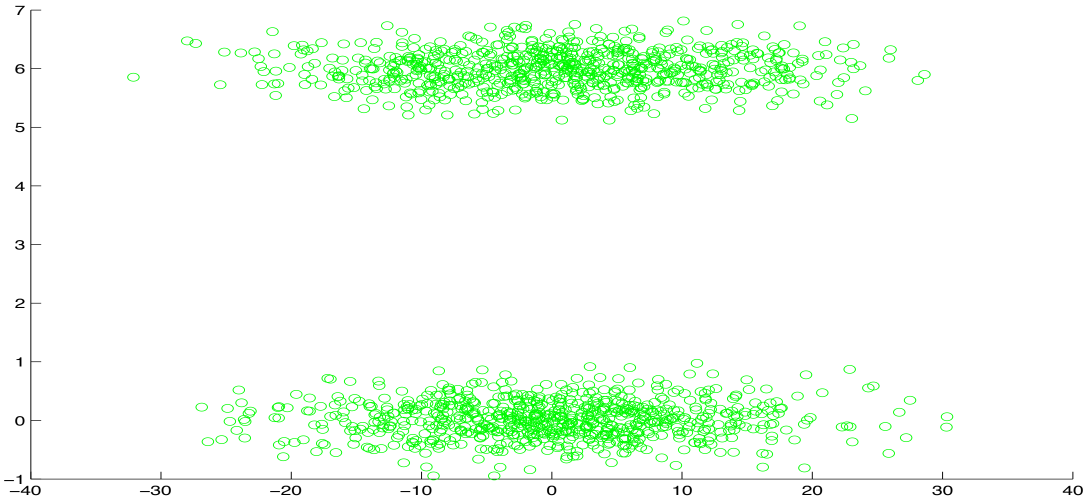
Let us examine single linkage in the same problem. The dendrogram is



This looks promising.

Problems With Complete Linkage

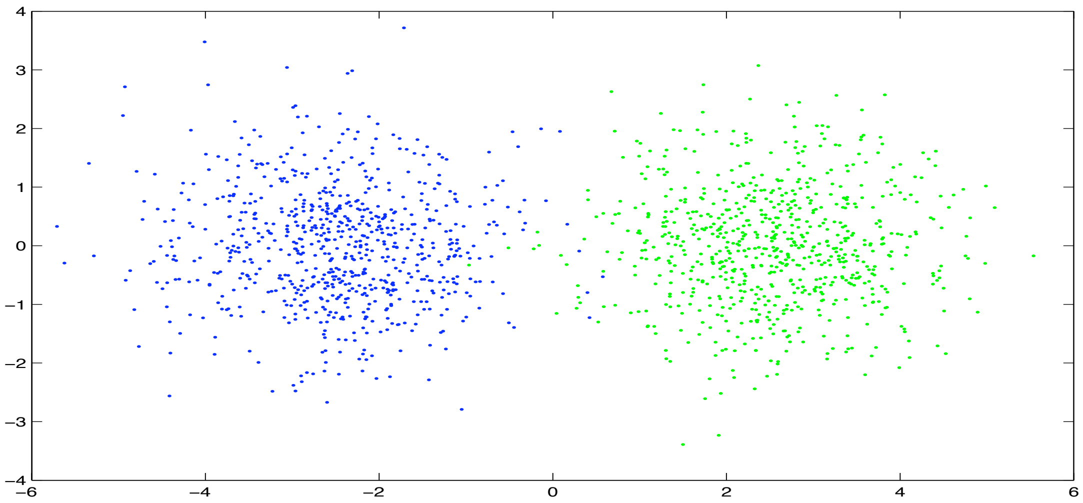
And the classification looks like:



The misclassification rate is 0%!!! Group Average linkage will fail here too. Can you see why?

Problems With Single Linkage

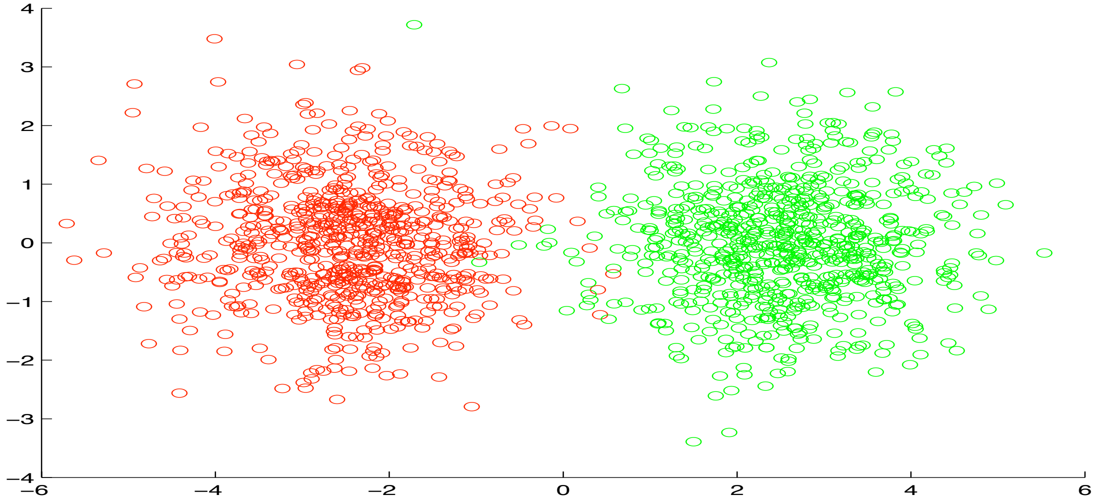
Now consider the example



Can you see that single linkage is going to not work here.

Problems With Single Linkage

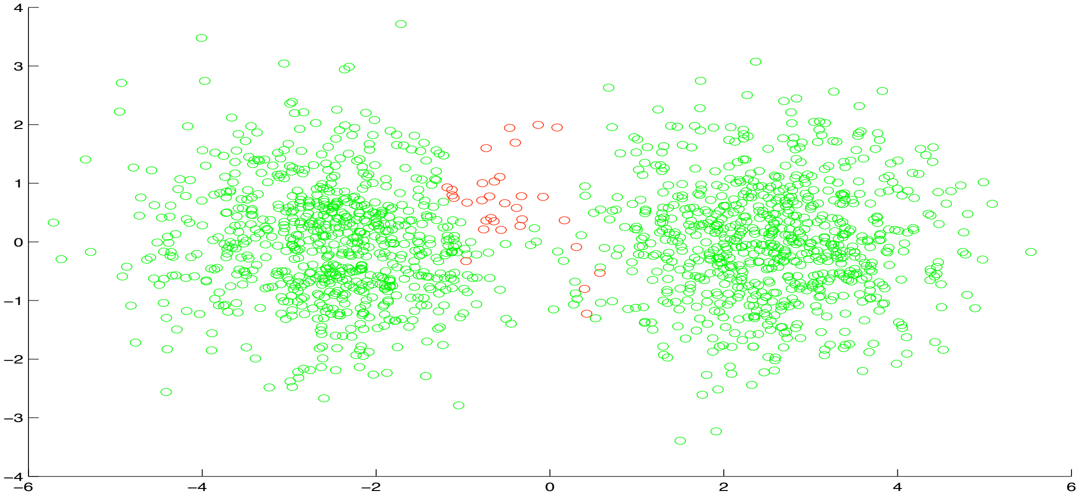
Here's what happens in the classification.



The misclassification rate here is 50%.

Problems With Single Linkage

However with complete linkage we get:



So complete linkage works here. Average linkage will also do the job here. Basically there are no simple answers when it comes to clustering since there is no real model behind it.