Cluster Analysis Tutorial

Nick Coleman

22 March 2002

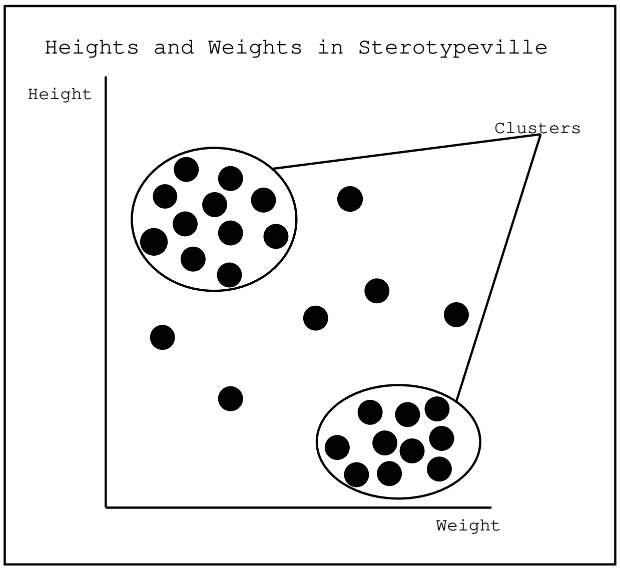

Figure 1: Clusters in Sterotypeville

1 Introduction

Clustering, also known as Cluster Analysis or Statistical Classification, is the grouping together of items from a population depending the measured similarity between them. We choose important features of the items in our population and decide how similar the items are based on the features they share. For example if we choose too compare height and weight of a sample of individuals from the town of Sterotypeville (figure 1) we can clearly there are two clusters in Sterotypeville, short fat people and tall skinny people.

Originally clustering methods were developed by biologists to classify animals and plants, these methods are applicable to any data. There are many areas in which clustering is applied, including Computer Vision and Data Mining.

This tutorial will give an overview of the main types of clustering algorithm: Hierarchical, agglomerative and decisive, and non hierarchical methods like iterative partitioning, and density search techniques. Also explain in more detail Centroid Cluster analysis and the K-means algorithm.

Explanations of the statistical terms used in this tutorial can be found in Lindsay Smiths principle components analysis tutorial for cosc452 from last year (2001).

2 Measures of Similarity

Once you have decided the features that are important in the classification of your items you must decide how you are going to measure how similar each item is. The simplest way is to compare Euclidean distance. If you imagine each of the n measured features as an orthogonal axis, by orthogonal we mean they are perpendicular for example height and weight are two separate things and should have their own axis each. Each item can be represented as a point in n-dimensional space and the the similarity between items is the distance between them. For example the two axes used in Figure 1 are height and weight. Euclidean distance between

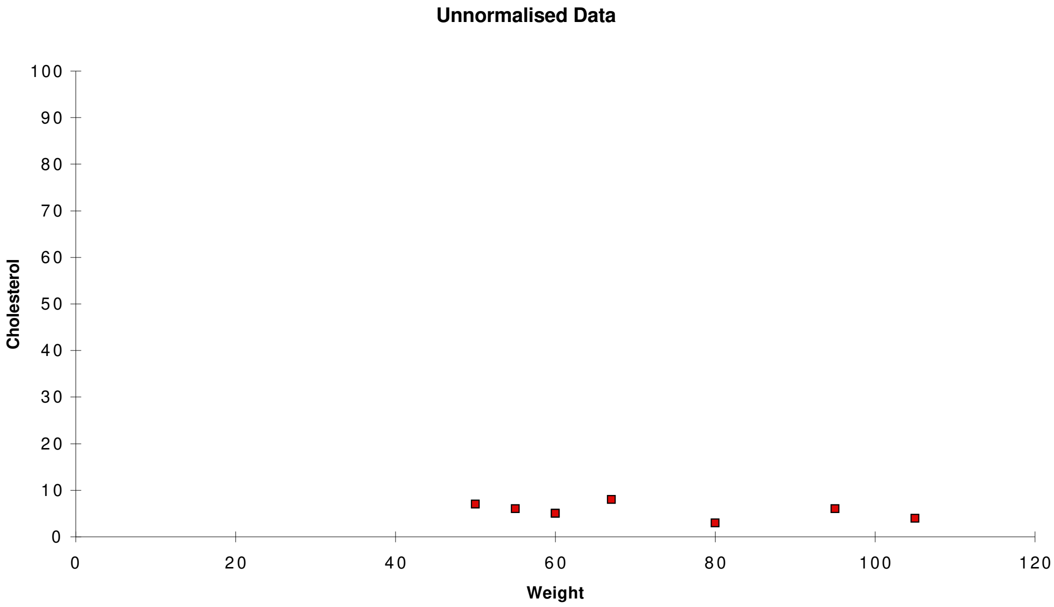

Figure 2: The Cholesterol becomes Insignificant

X and Y is calculated using Pythagoras on the line between them as follows:

$$D = \sqrt{(X_1 - Y_1)^2 + (X_2 - Y_2)^2 + \ldots + (X_n - Y_n)^2} \tag{1}$$

Where X_1 to X_n are the values of each measure for one item and Y_1 to Y_n are the values for the other one.

However when using distance as a measure of similarity it is important to normalise the data. By normalising data we scale it so all measures have some standard mean and some standard deviation. If different measures have different scales then they cannot be compared. For example (see figure 2 if you were classifying a population according to their weight and total cholesterol level you would have two axis, cholesterol measured in mmol/L and weight measured in Kg, the cholesterol measure may vary between 3 and 10 mmol/L while the weight may vary between 3 and 150Kg. These measures would have to be scaled so they covered a similar range, otherwise the weight measure would make the cholesterol one insignificant.

To standardise a value x_i from a population with standard deviation σ and mean $\bar{x}$ use the equation:

$$x_i' = \frac{x_i - \bar{x}}{\sigma} \tag{2}$$

This will scale each data point so the data has a new mean of 0 and a new standard deviation of 1.

If the data is not normally distributed, but rather spread evenly over a range r, you can normalise the data so it all lies between 0 and 1 by dividing by the range. This will give you a range of between 0 and 1, if you would rather have a range between -0.5 and 0.5, you can subtract the half the range r (which if the data is evenly distributed should equal to the mean $\bar{x}$) before dividing by the range r which if the data is evenly distributed should .

$$x_i' = \frac{x_i - \frac{r}{2}}{r} \tag{3}$$

There are other measures of similarity you can use. These include correlation coefficients for continuous data and other coefficients for binary (eg: blue or brown eyes: yes or no) data.

For a more information on similarity measures check [2, 5]

3 Have we got a Cluster?

The items in a cluster not only have to be similar, but the actual clusters have to be different from each other, otherwise they would be similar and therefore be part of the same cluster. So we are looking to see if we have found some clusters we need not only to verify if the items inside each cluster are similar but also that the clusters are sufficiently different to be considered separate clusters.

To measure how similar the items in a cluster are we could measure their average distance to the centroid of the cluster. This would tell us how tightly packed our clusters are. Tightly packed is a good thing because the 'closer' the items the more similar they are. Since square rooting the distance squared doesn't change which distances are larger than the others we wont bother doing it. This mean distance to the centroid idea is similar to something statisticians call variance. Variance is defined as the mean of the square of the difference between each item and the mean of all of the items or:

$$\sigma^2 = \sum_{i=1}^{n} \frac{(x_i - \bar{x})^2}{n} \tag{4}$$

Where σ^2 is the variance (standard deviation σ is the square root of variance), n is the number of items, x_i is the item and $\bar{x}$ is the mean. Since we are not statistician's we can just think of variance as the average distance to the centroid. So we replace the $x_i - \bar{x}$ term with the distance to the centroid.

You might have already guessed that to find the difference between two clusters we just use the distance between them. You are about to read how when we calculate the difference between clusters in different ways for hierarchical agglomerative methods, the same methods can be used here.

So in summery we want items in our clusters to have a small average distance to their centroids, and our clusters to have large distances between them.

4 Hierarchical Methods

Hierarchical methods group the closest clusters, so each cluster is a node of a tree and the closest two nodes are joined. This is repeated until all the clusters are joined together.

4.1 Agglomerative

Agglomerative algorithms are a subset of Hierarchical methods. These methods start with as many clusters as items and amalgamate clusters until you are left with a single cluster. The general algorithm is as follows:

1. Each item begins as its own cluster.

2. The two most similar clusters are joined.

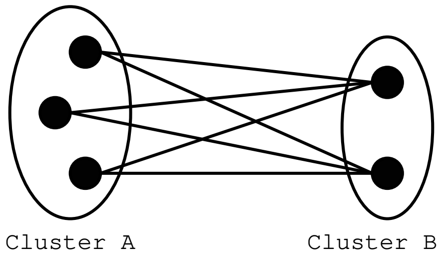

Figure 3: Calculating the Group Average

3. The new cluster is recorded

4. If more than the desired number of cluster remains then repeat from 2.

4.1.1 Methods

Agglomerative methods differ mainly on how you decide the similarity or distance between clusters.

One measure of the distance between two clusters is average distance between each set of two items containing a member from each cluster (see figure 3). This is known as the Group Average Method. It has the disadvantage of having to calculate the difference between every node with every other node that is not a member of the same cluster.

The single link method defines the difference of two clusters as the difference between their closest neighbors, conversely the complete link method uses the difference between the furtherest neighbors. This has the same overhead as the group average method of having to calculate the difference between every node that is not in the same cluster as the current node to find which two are closest.

The agglomerative method I have implemented is Centroid Cluster Analysis. As the name suggests this method uses the difference between each cluster as the difference between each clusters centroid. To calculate the component of a clusters centroid C along some axis X we use the following equation; where $x_1 \ldots X_n$ are each of the n clusters component along the X axis.

$$C_X = \frac{x_1 + \ldots + x_n}{n} \tag{5}$$

The Centroid Cluster Analysis method has the advantage of only having to calculate the difference between each cluster's centroid, which can be updated at each merger by taking the weighted average of the two clusters centroids.

4.1.2 Example

The following is as simple example of Centroid Cluster Analysis using only 6 data points $\{(0.9, 0.4), (0.7, 0.4), (0.2, 0.6), (0.2, 0.5), (0.8, 0.3), (0.1, 0.5)\}$

As you can see from figure 4, this trivial example has two obvious clusters. We would expect these two clusters to be the last remaining before finally being combined to form the set containing all items.

My simple implementation (Appendix A) used Euclidean distance (1) to calculate the similarity between centroids. It produced the following results:

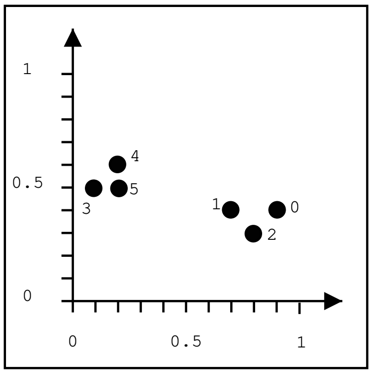

Figure 4: Step1

Beginning with the the following clusters it did the following:

- Cluster: 0;

  Centroid: (0.90, 0.40);

  Members: {(0.90,0.40)}

- Cluster: 1;

  Centroid: (0.70, 0.40);

  Members: {(0.70,0.40)}

- Cluster: 2;

  Centroid: (0.80, 0.30);

  Members: {(0.80,0.30)}

- Cluster: 3;

  Centroid: (0.10, 0.50);

  Members: {(0.10,0.50)}

- Cluster: 4;

  Centroid: (0.20, 0.60);

  Members: {(0.20,0.60)}

- Cluster: 5;

  Centroid: (0.20, 0.50);

  Members: {(0.20,0.50)}

As seen in figure 4 we start of with each data point being the sole member of its own cluster. Clusters 4 and 5 are the closest, with the distance between them being $\sqrt{0.01}$ so we join these to form a new cluster 6 as seen in figure 5.

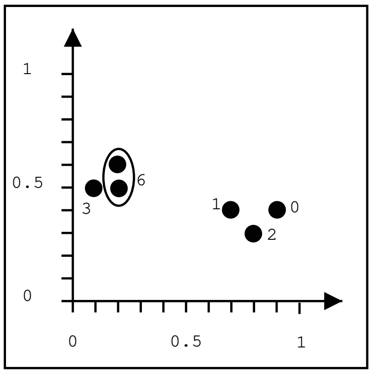

Figure 5: Step2

- Cluster: 0;

  Centroid: (0.90, 0.40);

  Members: {(0.90, 0.40)}

- Cluster: 1;

  Centroid: (0.70, 0.40);

  Members: {(0.70, 0.40)}

- Cluster: 2;

  Centroid: (0.80, 0.30);

  Members: {(0.80, 0.30)}

- Cluster: 3;

  Centroid: (0.10, 0.50);

  Members: {(0.10, 0.50)}

- Cluster: 6;

  Centroid: (0.20, 0.55);

  Members: {(0.20, 0.60), (0.20, 0.50)}

Now we can see from figure 5 clusters 3 and 6 are now the closest together with a distance of $\sqrt{0.0125}$ these are amalgamated to form cluster 7 (figure 6).

- Cluster: 0;

  Centroid: (0.90, 0.40);

  Members: {(0.90, 0.40)}

- Cluster: 1;

  Centroid: (0.70, 0.40);

  Members: {(0.70, 0.40)}

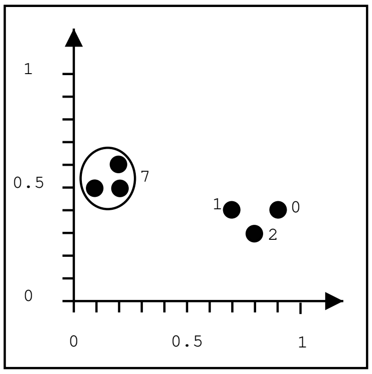

Figure 6: Step3

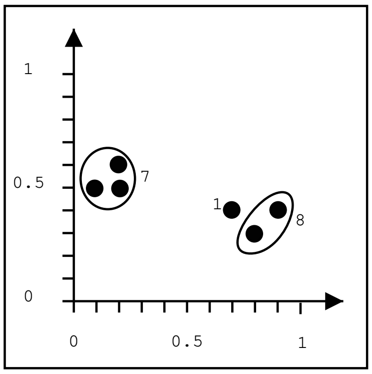

Figure 7: Step4

- Cluster: 2;

  Centroid: (0.80, 0.30);

  Members: {(0.80, 0.30)}

- Cluster: 7;

  Centroid: (0.17, 0.53);

  Members: {(0.20, 0.60), (0.20, 0.50), (0.10, 0.50)}

Now the next two clusters that are closest together are 0 and 2 (figure 6), they are $sqrt 0.2$ apart. We join them.

- Cluster: 1;

  Centroid: (0.70, 0.40);

  Members: {(0.70, 0.40)}

- Cluster: 7;

  Centroid: (0.17, 0.53);

  Members: {(0.20, 0.60), (0.20, 0.50), (0.10, 0.50)}

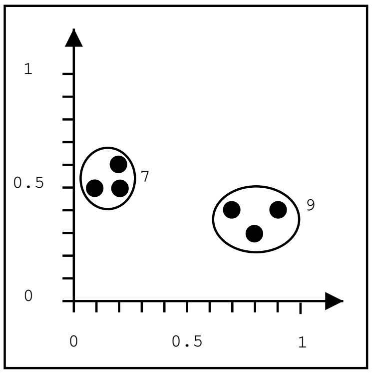

Figure 8: Step5

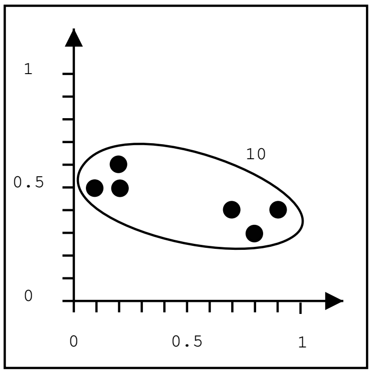

Figure 9: Step6

- Cluster: 8;

  Centroid: (0.85, 0.35);

  Members: {(0.90, 0.40), (0.80, 0.30)}

Now looking at figure 7 we can see that 1 and 8 are closest together, $\sqrt{0.02}$. These amalgamate to for 9 (figure 8).

- Cluster: 7;

  Centroid: (0.17, 0.53);

  Members: {(0.20, 0.60), (0.20, 0.50), (0.10, 0.50)}

- Cluster: 9;

  Centroid: (0.80, 0.37);

  Members: {(0.90, 0.40), (0.80, 0.30), (0.70, 0.40)}

Now if you look really closely you can see the distance between 9 and 10 is less than the distance between any other two clusters (figure 8). We amalgamate these two too form 10 (figure 9) and we stop here because we have run out of clusters to merge with.

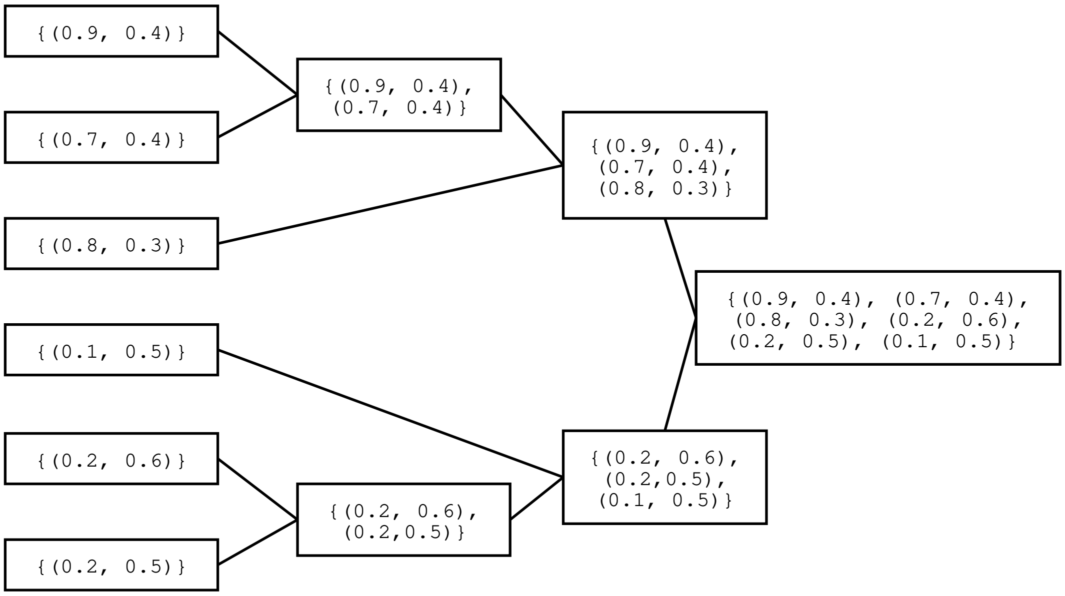

Figure 10: Hierarchal relationship

If you follow through the above you will see at each step the closest two clusters are merged. This forms hierarchical relationships between the items, which can be represented as a tree (figure 10). As you can see the two obvious separate clusters were the last to be merged together.

4.2 Divisive

Divisive methods as the name suggests start with the whole group as one big cluster and dividing it to get more and smaller clusters. These methods generally fit into two categories, Monothetic considers one variable at each split and Polythetic considers all variables at each split.

4.2.1 Monothetic

Monothetic use one (normally binary) variable to split the population at each level , the best way to explain them is with an example.

Say you have a population of people, you could break them down into clusters by dividing the population in to male and female, then into blue and brown eyes. This gives you four clusters: blue eyed males; brown eyed males; blue eyed females; brown eyed females.

4.2.2 Polythetic

An example of a polythetic method is to move items from the big cluster to a cluster that they are more similar too. So to start with you could take the item that is furtherest away from the centroid and make it is own cluster, then move items into the group that they are closest too. Move the clusters that are furthest away from the centroid. If an item is furthest away from its centroid but is not closer to another cluster start a new cluster with it.

4.3 Discussion

Hierarchical methods are good if you know the data you are examining has hierarchical relationships. Hierarchical methods can be altered to stop before all the clusters are merged. The Taxmap method does this, it uses the single link method and stops joining clusters when the next cluster is 'much further' away than the last pair were. Taxmap is a density search method. Another advantage of hierarchical clustering is you don't need to know the number of clusters before you start. Hierarchical methods can also suffer from an effect known as chaining, this is caused by chains of small numbers of points spanning the distance between clusters and joining big clusters together.

5 Iterative Partitioning

Iterative Partitioning methods work by estimating clusters and iteratively improving on this until it converges on a result.

5.1 K-Means algorithm

1. Choose initial centroids of K clusters.
2. Add each item to the cluster whose centroid they are closest too.
3. Recalculate each centroid.
4. If the new centroid is different repeat from 2.

The methods for choosing the first K elements vary. One was is to choose the K items that are mutually furtherest apart, another is to choose K random points in you n-dimensional space. In my implementation I choose the first K items.

5.1.1 Example

Using the same data as before, $\{(0.9, 0.4), (0.7, 0.4), (0.2, 0.6), (0.2, 0.5), (0.8, 0.3), (0.1, 0.5)\}$ (see figure 4), we can test the K-Means algorithm. Again my code is listed in Appendix A.

In this example (figure 11), the first two points, 0 at (0.90, 0.40) and 1 at (0.80, 0.30) are chosen as the centroids of the initial clusters. The points are now added to the cluster that they are closest too. After all points are assigned to a cluster the centroids are recalculated to give (0.85, 0.35) and (0.30, 0.50). The points are now moved to the cluster whose centroid they are closest too. The centroids are recalculated to give (0.80, 0.37) and (0.17, 0.53). Once more the points are moved to their closest cluster and the cluster centroids recalculated, this time the give (0.80, 0.37) and (0.17, 0.53) again, since this is the same as last time we know nothing has moved and we can call it a day with the clusters we have.

5.2 Discussion

Iterative Partitioning methods such as the K-means algorithm have the obvious disadvantage of having be told the number of clusters in the beginning. This can be overcome by varying K to optimise clusters so that the distance between clusters is maximised and the distance

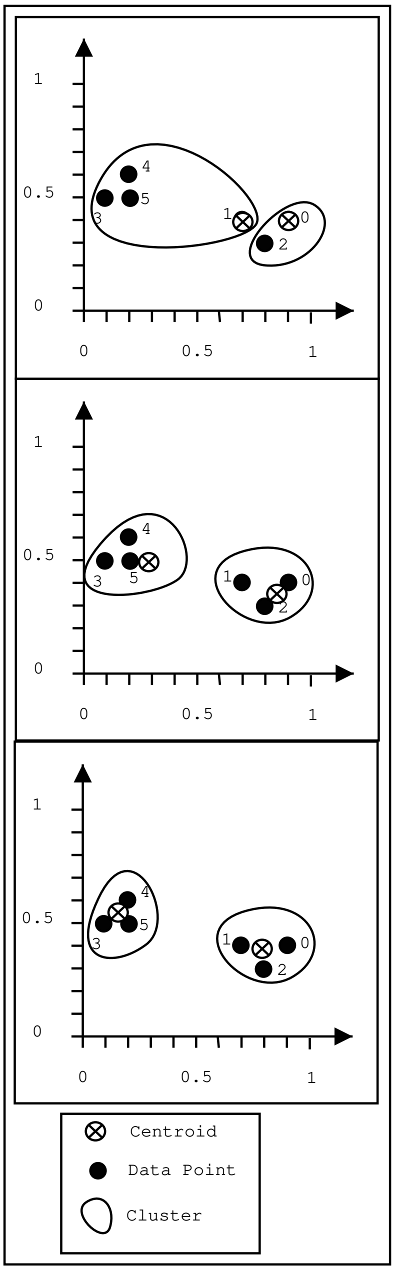

Figure 11: K-Means Example

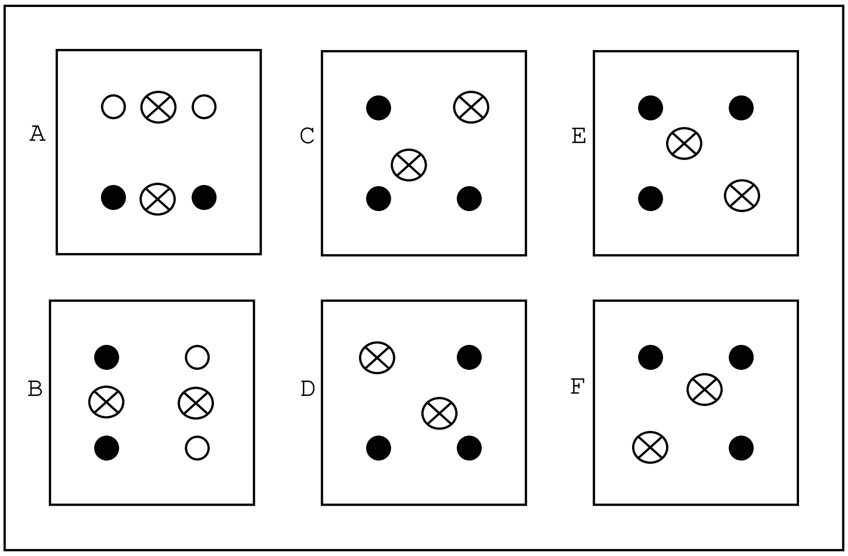

Figure 12: Stable Suboptimal Solutions

between points inside clusters is minimised. An advantage of these methods is that if an item is classified in an unsuitable cluster it can be moved to another one at a later stage.

A disadvantage is that a stable sub-optimal clustering could be found; for example if there are 4 points arranged in a square, there are 6 different ways the K-means algorithm could cluster them. It is obvious from figure 12 that when the centroids are arranged as they are in the middle and right hand clusterings we can see that every point is in the cluster of it's nearest centroid and are therefore stuck in this stable, though suboptimal state. It is easy to show this is a sub optimal.

Say the data points are at (0,0), (0,1), (1,0), and (1,1). It is easy to see that C, D, E and F all have the same distance between clusters and the same average distance from each member of a cluster to the clusters centroid; similarly so do A and B. So lets figure out the average distance squared between points and their clusters centroids (we don't need the actual distance, square roots just add confusion), and the distance between the two centroids for 2 different possibilities A and F. It is easy to see that the average distance of points to their centroids clusters is $\frac{1}{2}$ so the average distance squared is $\frac{1}{4}$, the distance between clusters is 1 and the distance squared is also 1. Now for F; it's easy to see the centroid of the black group is at $(\frac{2}{3}, \frac{2}{3})$ and the white group is at $(0, 0)$. Now the distance squared between centroids is $(\frac{2}{3})^2$ which is $\frac{4}{9}$. $\frac{4}{9} < 1$ so the clusters in A and B are further away from each other than the clusters in C, D, E and F. The mean distance to each centroid in A is $\frac{1}{2}$ and distance squared is $\frac{1}{4}$. For it is easy to calculate the distance squared from the upper right point to the black centroid is $\frac{4}{9}$ and the other two is $\frac{5}{9}$ and the distance from the white point to its clusters centroid is 0 so the average distance to centroid for all the clusters in C to F is $\frac{0+\frac{4}{9}+\frac{5}{9}+\frac{5}{9}}{4} = \frac{7}{18}$, $\frac{1}{4} < \frac{7}{18}$ so solutions C to F are suboptimal.

The moral of the story is that the way you decide to choose your beginning centroids can have an effect on the final outcome.

Since Iterative Partitioning techniques have to iterate over the data a number of times they can take a long time to run on large sets of data.

6 Other Methods

Other techniques for clustering include Density Search techniques, Clumping and Neural Networks.

Density Search techniques generally try to find areas with high density of points surrounded by a low density area.

The Taxmap method was mentioned in the discussion of Hierarchical methods, it uses single linkage and stops adding more items when the next item to add is 'much further' away than the last it added.

The Cartet count technique works by dividing the n-dimensional space up into 'cartets' or hypercubes and count the number of items in each one, when adjacent cartets have a relatively high number of items are joined together to form clusters.

DBSCAN works buy making each item in a cluster have a certain number of other items within a certain radius around it. Since membership to a cluster only depends on the items within a certain radius, new data can be added to existing clusters using this method very efficiently.

Clumping techniques are techniques that allow data to overlap.

Neural Networks are also being used for clustering, as this is the sort of noisy task that they supposedly excel at.

7 Validation Techniques

We want to know if we have found actual clusters in our data or have we just got groups of points that are not related.

There are three main groups of methods for deciding this:

7.1 External Criteria

These methods involve using your clustering technique on artificial data, and comparing the results you got from the real data. You have doctored to have what you think the general characteristics of the real data are. This requires you have some idea of the general characteristics of your data.

Also random data can be used to see if the cluster you found were probable on random data containing no clusters. Techniques involving the generation of random data are called Monte Carlo procedures. See Jonathan Pengelly's cosc 453 tutorial from last year (2001) on Monte Carlo methods.

Another external criteria is significance tests on external variables. What this means is to compare features of the items that were not considered when creating the clusters. And see if these features are statistically significantly similar for items in the same cluster. This tells you that you have classified the items so items in a certain cluster share features that the cluster was based on. For example if you were crusting by if a creatures were warm blooded and lay ed eggs, and then found that all of the creatures in one cluster had feathers while the others didn't you could be reasonably confident that your clustering is correct.

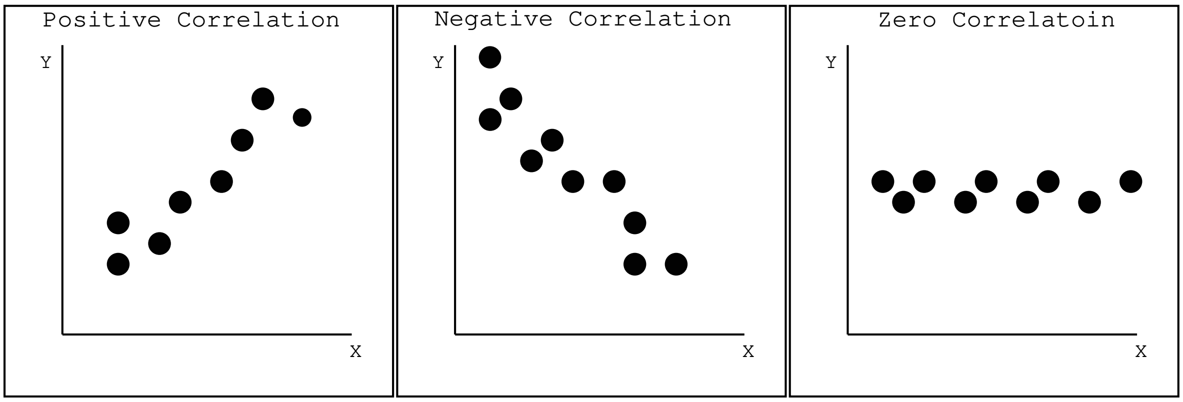

Figure 13: Correlation

7.2 Internal Criteria

Methods in this category rely totally on information from the data and clusters we have. Hierarchal clustering methods can also be validated by information from their hierarchical tree structure.

A Cophenetic correlation compares the similarity of two clusters merging together with the similarity of the original items in each of these clusters. There should be a positive correlation so, items that are far away from each other when are finally together by two clusters that were also far away from each other, and conversely if the items were close together should be made a set of the same Custer by a merge of two close clusters. So if you graph similarity of the original items vs the similarity of the two clusters that merged to make these two items part of the same cluster you should get an up wards sloping graph, a positive correlation. So say the distance between two of the original items was on the x axis in figure 13 and the difference between the two clusters who when joined were the first to encapsulate both of the original items is on the y axis in figure 13. So if we have a positive correlation y is increasing with x which is what we expect. If we have no correlation then our clustering was probably incorrect, and if we have a negative correlation we are either extremely unlucky or we probably have a ¡ around the wrong way somewhere in our code.

The measures described above in the *Have we got a Cluster?* section can be used for non hierarchical clustering data. The distance between clusters is a good way of telling if we have chosen to have too many clusters because some natural clusters may be split and these clusters will be very close together. Like wise if you don't have enough clusters there will be some clusters that have a very large mean distance to the centroid because our cluster is covering more than one natural cluster the mean is likely to be somewhere between them the mean difference is going to be large.

7.3 Relative Criteria

Relative criteria involves comparing the results you get for your clustering algorithm with results for the same data with same algorithm but with different parameters. The more algorithms agree on one clustering solution the more likely it is to be correct. You want to try to choose the best result, you can compare the results by again measuring the distance between clusters and the average distance from the centroid of the items in the cluster.

8 Applications

Applications of clustering exist in computer vision include edge detection and image segmentation.

An example of clustering being used for image segmentation would be if you have a picture were each pixels colour is represented by three numbers, one the the intensity of red, green and blue on that pixel. Suppose there is a picture of a red truck on a green background and we want to separate out the red truck from the green background. We would have a 5 dimensional space (2 for the 2 dimensional picture and 1 for each pixel) and using clustering techniques we could group together nearby pixels that had a similar colour, and clusters of pixels with similar colour and position normally represent the same object, like in this case they represent a red truck.

Cluster analysis is also used in many other fields. It was first used as a way of classifying different species. Another computer science field where clustering is used is information retrieval.

9 Conclusion

This tutorial has explained what clustering is. It has explained how to measure the similarity between two items. It has explained the main types of clustering algorithms and explained in depth the K-Means and Centroid Cluster Analysis algorithms. It has explained what validation of the results is and explained some techniques for doing so. It has given a brief idea of some of the applications.

[3, 2, 4, 1]

References

[1] The Animated Software Company. The internet glossary of statistical terms. http://www.animatedsoftware.com/statglos/statglos.htm.

[2] Qin He. A review of clustering algorithms as applied in ir. Master's thesis, University of Illinois, 1999.

[3] J.A.Hartigan. *Clustering Algorithms.* John Wiley and Sons, 1975.

[4] Michalis Vazirgiannis Maria Halkidi, Yannis Batistakis. On clustering validation techniques. Athens University of Economics & Business.

[5] Unkown. Simirlartity measures. http://prlab.ee.memphis.edu/frigui/ELEC7901/UNSUP2/SimObj.html.

10 Appendix A: Code