Student name, Data Mining (H6016), Assignment Paper 2. Dec 2010.

Agglomerative Hierarchical Clustering

1. Abstract

In this paper agglomerative hierarchical clustering (AHC) is described. The algorithms and distance functions which are frequently used in AHC are reviewed in terms of computational efficiency, sensitivity to noise and the types of clusters created. Techniques used in evaluating the resulting cluster set are described and dendrograms are explained and their usefulness in determining an optimal final cluster number is explored. The application of AHC to document collection searches is reviewed including a comparison of the performance of the AHC algorithms.

2. Introduction

In data mining, classification is a form of supervised learning where a model is trained on known class labels in the training dataset. If class labels are not available clustering can be used where the desired outcome is that instances within a cluster are more similar to each other than to instances in other clusters. As the model is not trained on class labels clustering is a form of unsupervised learning. Common applications of clustering include image grouping, genetic information comparison and information retrieval.

Clustering types include partitional clustering which divides the dataset into a preselected number of clusters, instance density based clustering approaches and hierarchical clustering which is described in this paper. In divisive hierarchical clustering (DHC) the dataset is initially assigned to a single cluster which is then divided until all clusters contain a single instance. The opposite approach is called agglomerative hierarchical clustering (AHC) where each instance is initially assigned to a separate cluster and the closest clusters are then iteratively joined or agglomerated until all instances are contained in a single cluster (Figure 1).

This paper will focus on AHC as it is more frequently used than DHC (Vesanto et al, 2000) and the information contained in this paper is also generally applicable to DHC. One advantage of AHC is that preselection of the final cluster number is not required allowing a domain expert to analyze the resulting cluster hierarchy in order to determine the optimal cluster number. AHC can be applied successfully to both regularly and irregularly shaped clusters if the appropriate algorithm is selected as described in section three. One of the disadvantages of AHC is that the decision to join clusters is localised to the two clusters being joined which can produce poor clustering decisions, and once joined clusters in AHC cannot be separated. AHC has significant computational overhead on large datasets as it requires the creation of a complete distance matrix (where all instance distances are calculated). As the various clustering algorithms and distance functions discussed in this paper produce different cluster hierarchies (Figure 2) it may be necessary to execute the clustering process multiple times in order to determine the

optimal algorithm and distance function combination. (Myatt et al, 2009) (Steinbach et al, 2008).

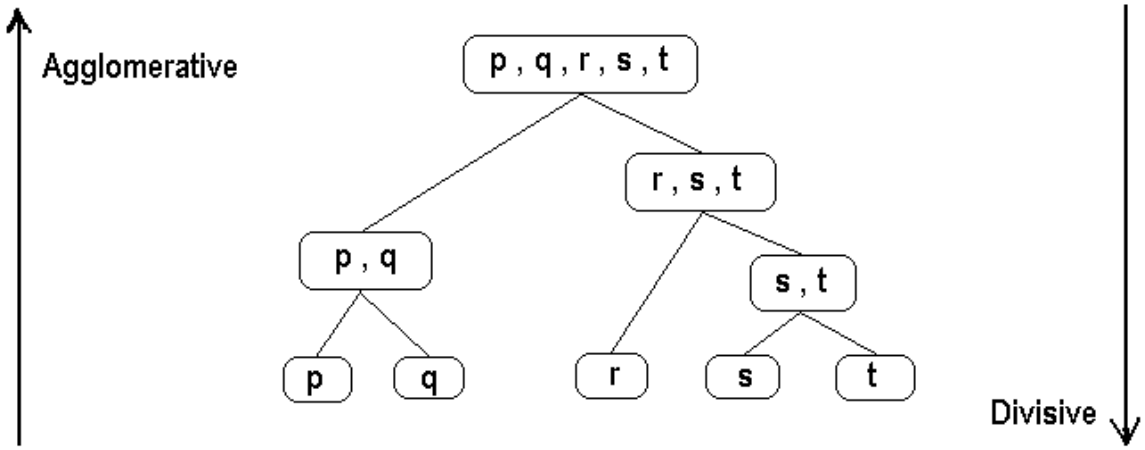

Figure 1. Agglomerative and divisive hierarchical clustering. [www.resample.com/xlminer]

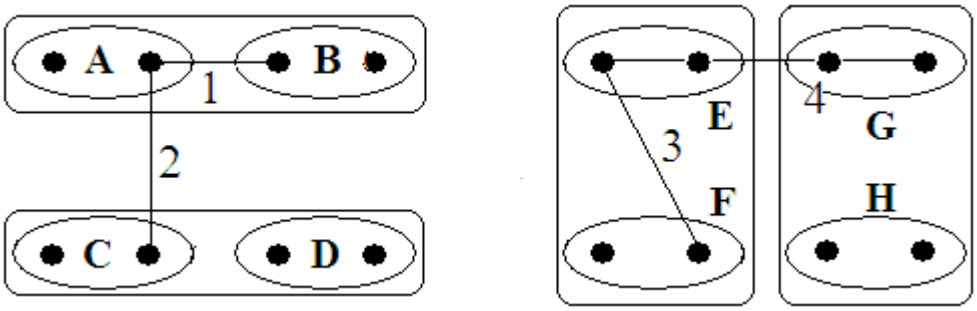

Figure 2. Single and complete linkage cluster sequence. Single linkage (see section three) on left joins the clusters horizontally as the smallest minimum distance (1) is between clusters A and B. Complete linkage joins the clusters vertically as smallest maximum distance (3) is between clusters E and F.

3. Cluster distance algorithms

This section reviews the most commonly used cluster distance algorithms in AHC which are single linkage (or nearest neighbour), complete linkage (or farthest neighbour), average linkage, centroid linkage and Ward's method.

In single linkage (Dunn, 1982) the cluster distance is the minimum instance distance between the two clusters (Figure 3A), so that for clusters x and y the distance is given as:

$$\mathbf{D(x,y)} = \sum_{i=1}^{Sx} \sum_{j=1}^{Sy} \text{Min}\{d(x_i, y_j)\}$$

(1)

where Sx and Sy are the cluster sizes.

As this distance measurement is localised to the two nearest neighbours the overall cluster shapes are not taken into account which can result in chaining (Figure 4) producing stretched or irregularly shaped clusters containing dissimilar instances. Single linkage is computationally efficient as the use of persistent best merges (Figure 4) means that it is not necessary to recompute the similarity of nearest neighbour clusters when one is joined to another cluster (Cimiano, 2006).

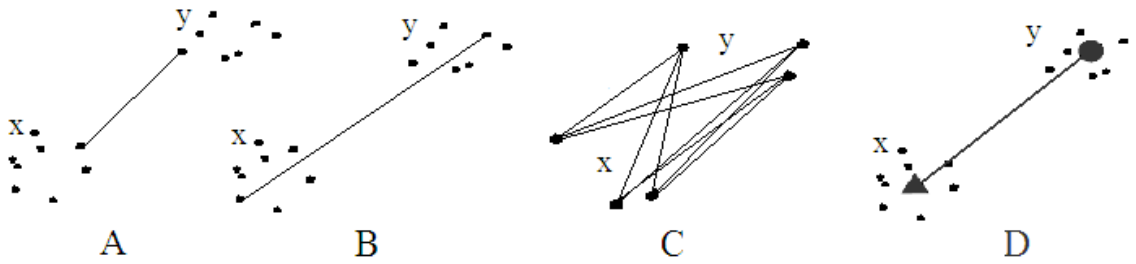

Figure 3. Single, complete and average and centroid linkage

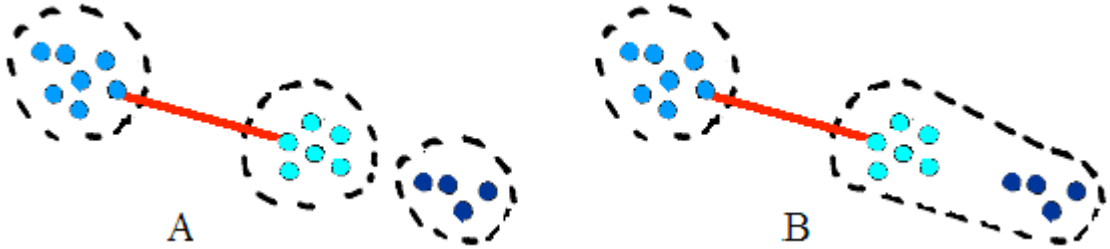

Figure 4. Single linkage persistent best merge. The nearest neighbour instances in A are also the nearest neighbours after the lower cluster has been joined to another cluster in B. Note: The lower cluster in B is exhibiting signs of chaining where the overall shape is stretching away from the merge point.

In complete linkage (Dunn, 1982) the cluster distance is the maximum instance distance between the two clusters (Figure 3B) and is given as:

$$\mathbf{D(x,y)} = \sum_{i=1}^{Sx} \sum_{j=1}^{Sy} \mathrm{Max}\{d(x_i, y_j)\}$$

(2)

where Sx and Sy are the cluster sizes.

This approach is non-local in that all cluster instances influence the distance calculation. Noise or outlier instances in irregularly shaped clusters can skew the distance calculation making this method more suitable for the compact regular clusters shown below (Figure

5A). Single linkage would be more appropriate for the irregularly shaped clusters (Figure 5B) as the highlighted outlier instance would skew the distance calculation in complete linkage (Tryfos, 1997). The concept of a persistent best merge does not apply to complete linkage as the merge point may change during a cluster join.

Figure 5. Comparison of single (SL) and complete (CL) linkage clustering on regularly and irregularly shaped clusters.

In average linkage (Anderberg, 1973) the cluster distance is calculated as the average distance between all instances in the two clusters and is given as:

$$\mathbf{D(x,y)} = \frac{1}{Sx\ Sy} \sum_{i=1}^{Sx} \sum_{j=1}^{Sy} d(x_i, y_j) \tag{3}$$

where Sx and Sy are the cluster sizes.

Average linkage (Figure 3C) is a compromise between single and complete linkage and attempts to maximise the coherency of the joined clusters (Berrar et al, 2003). As all instance distances influence the distance calculation this algorithm may be less sensitive to outlier instance values than complete linkage. Average linkage is computationally more expensive than single linkage as the concept of a persistent best merge does not apply.

In centroid linkage (Figure 3D) the cluster distance is calculated between the cluster centroids (Tsiptsis et al, 2010). The cluster centroid is located at the mean attribute values in the cluster. For a cluster with two instances (X1,X2,X3) and (Y1,Y2,Y3) the centroid (C1,C2,C3) is given as:

$$C1 = \frac{X1 + Y1}{2} \quad C2 = \frac{X2 + Y2}{2} \quad C3 = \frac{X3 + Y3}{2} \tag{4}$$

Centroid linkage has a lower computational overhead than average linkage as only the centroid distances are calculated and stored (Kimmel et al, 2006). Clustering is referred to as monotonic when lower level joined clusters are more similar to each other than higher level joined clusters. In centroid linkage it is possible for non-monotonic

clustering to occur (see section four) where the distance between two joined clusters is greater than the distance to their parent cluster (Hughes, 1994).

The Sum of Squared Error (SSE) is a measure of the distance between a cluster's instance attribute values and mean attribute values and is given as:

$$SSE = \sum_{i}^{Sx} \sum_{j}^{Sy} |Xij - Yi|^2 \tag{5}$$

where Sx is the number of instances
Sy in the number of attributes
Xij is the value of attribute j in instance i
Yi is the mean value of attribute j

A small SSE value indicates that all instances in the cluster are close to the cluster mean and therefore have a high degree of similarity. Ward's method joins the two clusters which minimally increase the value of SSE:

$$D(x,y) = SSE(x\ y) - (SSE(x) + SSE(y)) \tag{6}$$

where D(x,y) is the SSE change after joining clusters x and y
SSE(x,y) is the SSE of joined clusters x and y
SSE(x) is the SSE of cluster x
SSE(y) is the SSE of cluster y

When the total number of clusters is plotted against the average SSE value for a given dataset (Figure 6) an initial increase in the cluster number decreases SSE quite rapidly as the clusters become more similar, but at a turning point the reduction in SSE slows as the dataset becomes well partitioned and the clusters more cohesive (Guan, 2003). A turning point (if it can be identified) may be used as an estimate of the natural number of clusters in a dataset.

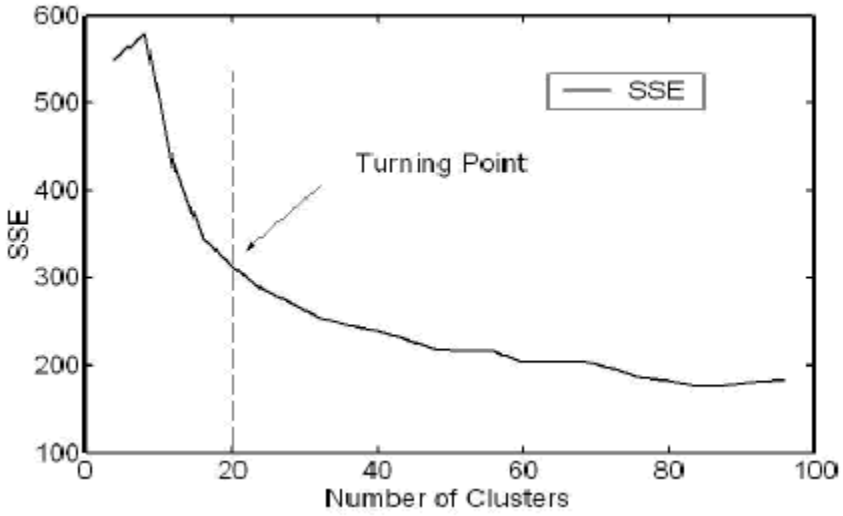

Figure 6. Cluster number vs average SSE value for KDD-99 network intrusion dataset

4. Dendrograms

Dendrograms are used to display the cluster hierarchy and the distances at which the clusters were joined (Figure 7A) which can be useful when selecting an appropriate number of clusters for the dataset (Figure 7B). Another approach in selecting a cluster number is to cut the dendrogram where there is a significant jump in the distance of the cluster joins (Figure 7B) which is equivalent to selecting the knee point in a k-Means curve.

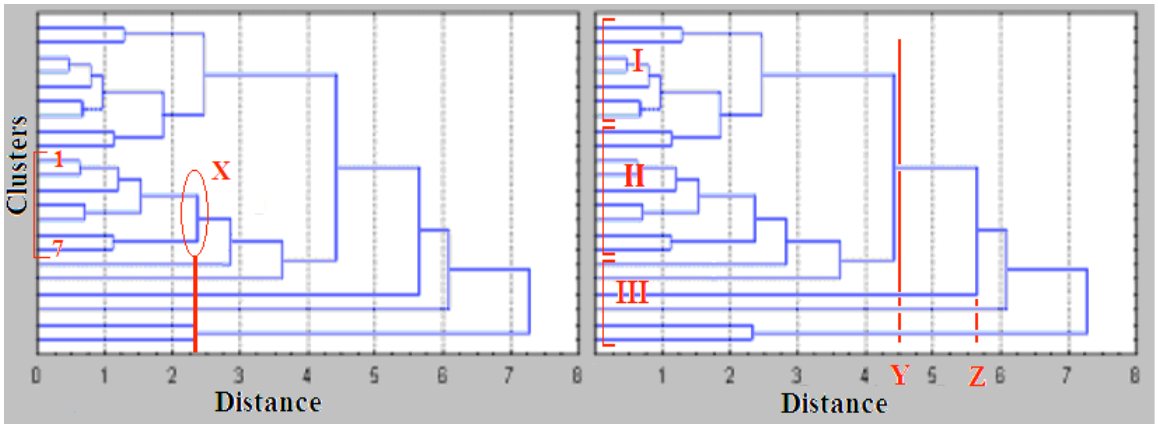

Figure 7. A. Cluster dendrogram with join X at a distance of 2.28 containing seven single instance clusters. B. Cutting dendrogram at distance of 4.5 (Y) produces two well partitioned clusters I and II and removes the outlier chained clusters at III. Dendrograms may also be cut at a jump in the distance values such as between Y and Z above.

Clustering is monotonic when lower level joined clusters are more similar than higher level joined clusters, but in centroid linkage it is possible to get non-monotonic clustering where the joined pair are more similar to the parent cluster than to each other, which

causes inversions or downward steps in the dendrogram (Figure 8). Inversions can impact the appropriate distance level at which to cut a dendrogram.

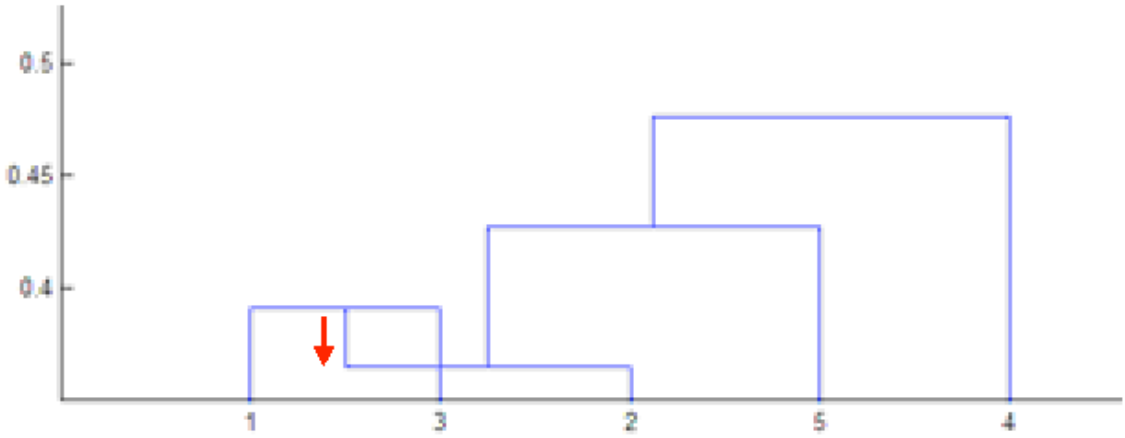

Figure 8. Non-monotonic dendrogram [www.mathworks.com/help/toolbox/stats/linkage.html]

5. Distance functions

The cluster distances in the algorithms described in section three can be calculated using a number of distance functions. The optimal distance function produces the most cohesive well separated clusters. These distance measurements may be skewed by attributes having larger than average values and therefore the data should be normalised before distances are calculated.

The euclidean distance between two instances with j attributes is given by:

$$d_{\mathbf{x},\mathbf{y}} = \sqrt{\sum_{j=1}^{J} (x_j - y_j)^2} \tag{7}$$

The squared euclidean distance calculation drops the square root term and is therefore slightly faster than the calculation of the euclidean distance. This function is useful for calculating relative as opposed to absolute instance distances:

$$d_{\mathbf{x},\mathbf{y}} = \sum_{j=1}^{J} (x_j - y_j)^2 \tag{8}$$

The manhattan distance is calculated by determining the distance required to move on a grid between two instances with j attributes:

$$d_{\mathbf{x},\mathbf{y}} = \sum_{j=1}^{J} |x_j - y_j|$$

(9)

The manhatten calculation is more sensitive to changes in the instance attribute values than euclidean distance (Figure 9).

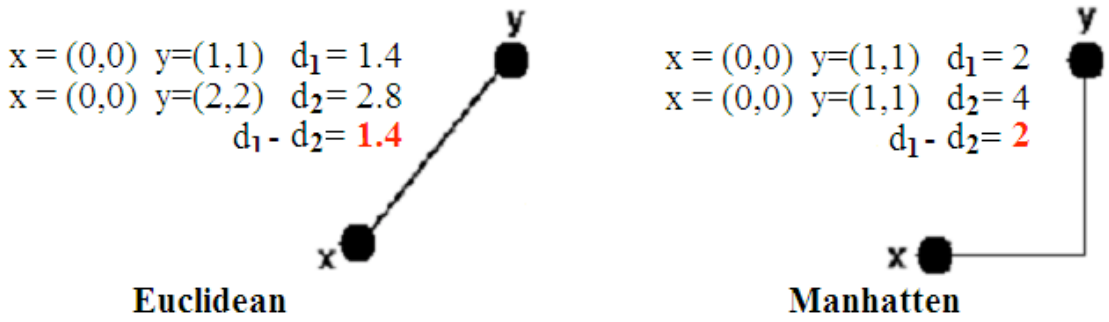

Figure 9. Euclidean and manhattan instance distance calculations.
A change of one unit in each attribute value (from one to two) increases the euclidean distance by 1.4 and the manhatten distance by 2.

The Chebychev distance between two instances is given by the maximum attribute distance between the instances:

$$d_{\mathbf{x},\mathbf{y}} = \sum_{j=1}^{J} \max\left(|x_j - y_j|\right)$$

(10)

See Appendix A for an example of the calculation of a chebychev distance.

The cosine similarity function calculates the cosine of the angle between two instance distances with j attributes:

$$d_{\mathbf{x},\mathbf{y}} = \frac{\sum_{j=1}^{J} x_j . y_j}{\sqrt{\sum_{j=1}^{J} x_j^2 . \sum_{j=1}^{J} y_j^2}}$$

(11)

This function is useful when calculating the similarity between documents where the keywords are assigned to attribute values in the two instances.

6. Data cleansing

One approach to data cleansing with centroid linkage (Tsiptsis et al, 2010) is to remove instances which contain attribute values which are more than 'n' times the centroid attribute value (typically n has a value of three to five) which helps to improve cluster cohesion and separation. Another approach (Liang 2007) is to move clusters with sizes below a specified size at each cluster level to a null cluster, where they can be later discarded or inserted into a cluster as appropriate. A similar approach (Almeidaa et al, 2007) initially removes outliers (Figure 10B), then applies AHC to the dataset (Figure 10C) and finally uses kNN classification to assign the outlier values to an appropriate cluster (Figure 10D).

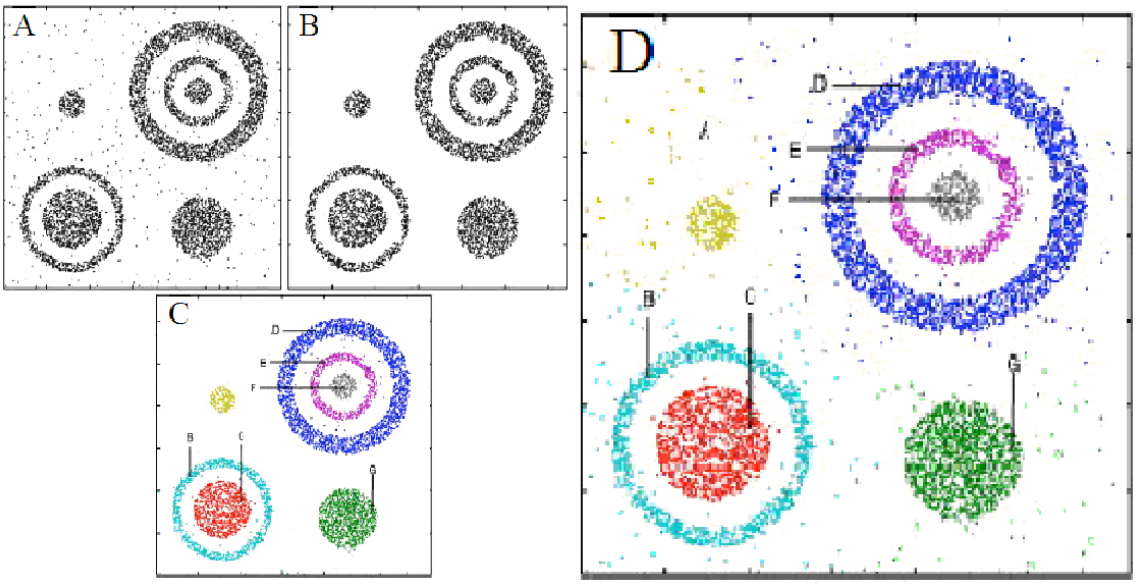

Figure 10. Data cleansing using AHC and kNN. A. Original dataset. B. Outliers removed. C. Remaining instances clustered. D. Outlier included in appropriate cluster using kNN classifier.

7. Evaluating the cluster hierarchy

A clustering algorithm can be evaluated by analysing it's ease of use and efficiency which includes execution times and storage requirements and it's accuracy or usefulness (Rand, 1971). One approach in assessing accuracy is to use a training dataset to compare a set of calculated clusters (X) with a set of known good clusters (Y). The rand index can be used to quantify the level of similarity between X and Y and is given as:

$$R = \frac{a + b}{a + b + c + d} \tag{12}$$

where:

a = instance quantity in same clusters in X and Y
b = instance quantity in different clusters in X and Y
c = instance quantity in same clusters in X and different clusters in Y

d = instance quantity in different clusters in X and same clusters in Y

A related cluster quality measure called the F-Score uses precision P and recall Rc where:

$$P = \frac{a}{a+c} \qquad R_c = \frac{a}{a+d} \tag{13}$$

The F-Score is given as:

$$F = \frac{2(P.Rc)}{P+Rc} \tag{14}$$

The highest average F-Score across all clusters indicates the optimal cluster set.

The Dunn Index (Dunn, 1974) can also be used to evaluate the cohesiveness and degree of separation of clusters. The optimal cluster set will have the highest Dunn index:

$$D = \frac{D_a}{D_b} \tag{15}$$

where Da is the smallest cluster distance and Db is the largest instance distance within a cluster in the cluster set. This measure is sensitive to noise as it will be skewed by large Db values (Yang, 2009).

A similar approach uses the cluster silhouette width (Rousseeuw, 1987) which is a reflection of it's cohesiveness and separation from other clusters:

$$S(i) = \frac{(Bi - Ai)}{\max(Ai, Bi)} \tag{16}$$

where Ai is the average distance of instance i to all other instances in it's cluster and Bi is the minimum average distance from instance i to all points in another cluster. S(i) can vary from a value of one which is optimal to minus one.

8. An application of AHC to document discovery

This section reviews research on using single, complete and average linkage and Ward's method to find relevant documents for a search string in a document collection (El-Hamdouchi et al, 1989). This research used seven document collections ranging from an aerodynamics collection containing fourteen hundred documents to a collection on

chemical characteristics containing twenty seven thousand documents (see collections in Table 2). The collections were initially cleaned of common and duplicate words and then clustered using the cosine similarity function. The goal of the clustering was to improve the efficiency of document searches whilst retaining a high level of relevancy in the returned documents. The lowest level clusters were searched first and the parent cluster of any cluster that returned relevant documents was then searched. This process continued until the required number of relevant documents had been found. These documents were compared to the most relevant documents found during a complete search of the document collection in order to evaluate the quality of the clustering.

The single linkage SLINK approach (Sibson, 1973) calculated the cluster distances based on the most similar documents in the two clusters which resulted in clusters containing large numbers of documents which were significantly dissimilar (Table 1). These clusters generally returned documents which were not relevant to the search string and were therefore removed by using a maximum cluster size threshold value. In this research the threshold value was set to forty which eliminated almost fifty percent of the documents in the single linkage clusters and reduced the usefulness of this algorithm. The best search results were obtained by using the smaller more cohesive clusters created by the other algorithms and it was noted that a high percentage of these clusters actually contained just a single document and it's nearest neighbour (Table 1). The complete linkage CLINK approach (Defays, 1977) calculated the distance based on the least similar documents in the clusters creating large quantities of cohesive clusters containing very similar documents and average linkage produced results which were quite similar (Table 1). Ward's method produced very tightly bound clusters which did not always reflect the underlying collection structure but were found to return very relevant documents in practise. A proposed outcome of this study was to combine (based on the search details) full collection searches with cluster searches.

Table 1. Cluster size by cluster type (Keen document collection).
In single linkage almost 50% of the clusters exceeded the maximum permitted cluster size and were removed from the cluster set. In contrast the other algorithms produced clusters with only two instances 75% of the time and no clusters exceeded the threshold cluster size.

Method	Cluster Size						
	2	3	4	5-20	21-40	>40	Total
Single	234	74	30	59	8	395	800
Complete	598	141	34	25	0	2	800
Average	556	125	48	67	2	2	800
Ward	634	130	30	6	0	0	800

Each clustering algorithm were evaluated in two ways. The first approach used an effectiveness measure E (Van Rijsbergen, 1979) based on the precision (quantity of returned relevant documents / total documents returned) and recall (quantity of returned relevant documents / total number of relevant documents in collection). A β term was

used to assign a relative level of importance to recall and precision where β = 0.5 indicated that precision had twice as much importance as recall. The effectiveness measure is given as:

$$E = 1 - \frac{(1 + \beta^2)\, PR}{(\beta^2 P + R)} \qquad (17)$$

where β is importance of recall and precision
P = precision
R = recall

For given values of β and the total number of relevant documents in collection and the total number of documents returned, a decrease in the effectiveness measure (Table 2) indicates that a higher level of relevant documents were being returned (see Appendix B example 2 for calculations).

Table 2. Effectiveness measure values by cluster algorithm

Collection	Single		Complete		Average		Ward	
β	0.5	2.0	0.5	2.0	0.5	2.0	0.5	2.0
Keen	0.89	0.87	0.88	0.86	0.85	0.84	0.84	0.83
Cranfield	0.87	0.83	0.93	0.92	0.80	0.74	0.83	0.79
Evans	0.91	0.94	0.94	0.96	0.91	0.94	0.91	0.94
Harding	0.93	0.95	0.95	0.97	0.90	0.94	0.92	0.95
LISA	0.95	0.94	0.96	0.95	0.93	0.93	0.94	0.93
INSPEC	0.94	0.96	0.96	0.97	0.90	0.93	0.92	0.95
UKCIS	0.96	0.97	0.96	0.97	0.94	0.96	0.94	0.96

Another approach used in evaluating the clusters was the total number of relevant documents returned from all document searches on a collection (T) and the number of searches on the collection which returned no relevant documents (Q). A increase in T and a decrease in Q (Table 3) indicates a higher level of relevant documents being returned.

Table 3. T and Q values by cluster type

Collection	Single		Complete		Average		Ward	
	T	Q	T	Q	T	Q	T	Q
Keen	80	25	83	22	105	21	111	22
Cranfield	279	109	148	136	444	89	364	96
Evans	46	19	30	20	44	19	45	18
Harding	63	38	40	38	88	37	69	34
LISA	17	21	16	21	26	21	25	19
INSPEC	68	37	50	38	123	27	94	33
UKCIS	137	114	124	111	177	104	185	107

9. Conclusion

We have seen in this paper how clustering is a form of unsupervised learning which can be used for data discovery when class labels are not available in the dataset. The goal is to create clusters which are cohesive and well separated in order to be useful in applications such as image grouping, genetic information comparison, information retrieval and document searches. Agglomerative hierarchical clustering (AHC) was found to produce clusters hierarchies which could then be reviewed to determine the optimal number of clusters to describe the dataset. This approach is possible as AHC does not require the preselection of a final number of clusters which is not the case in other clustering algorithms. One limitation in AHC was found to be that once clusters are joined they cannot be split so care must be taken when selecting the appropriate clustering algorithm and distance function. This can be a time consuming process. We have seen that AHC can be applied successfully to both irregularly shaped clusters (by using single linkage) and compact regular clusters (by using any of the other AHC algorithms). In single linkage the closest neighbours only influence the clustering decisions which may result in chained clusters containing dissimilar instances. Single linkage was found to be relatively computationally efficient as it takes advantage of persistent best merges and therefore this approach may be the optimal AHC algorithm for large datasets.

Complete and average linkage were found to produce compact and well separated clusters which were found to be particularly suitable for document searches. They do carry a higher computational overhead as the merge point changes on a cluster join and therefore the persistent best merge principle cannot be applied. Noise or outlier instances in the dataset can skew the distance calculation for complete linkage and it was found that this algorithm was most effective after data cleansing. Some effective approaches to data cleansing and outlier handling included removing instances which were remote from the centroid in centroid linkage, moving outliers to null clusters for later processing after the remaining instances have been clustered and using a combination of AHC and kNN classification to determine the appropriate cluster for the outlier instances. Average linkage was found to be slightly less sensitive to noise than complete linkage as all instance distances influence the calculation of the cluster distance. In the reviewed research centroid linkage produced similar results to average linkage and had a slightly lower computational overhead as only the calculation of the centroid distance is required when joining clusters. The concept of non-monotonic clustering was found to occur in centroid linkage which creates inversions or downward steps in the dendrograms and could impact the point at which the dendrogram is cut when selecting an optimal cluster number. Ward's method also produced compact well separated clusters and plotting the number of clusters against the Sum of Squared Error indicated a turning point which may suggest the natural number of clusters in the dataset.

Dendrograms were found to be useful in AHC in giving an overview of the cluster hierarchy and in selecting the final cluster number either by cutting the tree at a absolute cluster join distance or at a jump in the join distance values. In the research review on applying clustering to document searches it was found that single linkage produces large clusters which had poor precision and was rejected as a viable algorithm in this case. The

other algorithms produced reasonable results and it was concluded that a combination of full document collection searches combined with appropriate cluster searches was the best approach in supporting a wide range of document search queries.

10. Appendix A: Calculation of Chebychev distance

For two instances (0, 3, 4, 5) and (7, 6, 3,-1) the distance is:

$$d_{x,y} = \max(|0 - 7|, |3 - 6|, |4 - 3|, |5 + 1|)$$

$$= \max(7, 3, 1, 6)$$

$$= 7$$

11. Appendix B: Calculation of effectiveness measure E

Total number of relevant documents returned	= TN
Total number of returned documents	= TR
Number of relevant documents returned	= RR
Recall (R)	= RR/TN
Precision (P)	= RR/TR
Importance of recall versus precision	= β

Effectiveness measure:

$$E = 1 - \frac{(1 + \beta^2)\,PR}{(\beta^2 P + R)}$$

For TN = 45

Example 1 Let RR = 15
Change number of documents returned:

	Query A	Query B
TR =	**20**	**30**
P =	15/20	15/30
	0.75	**0.5**
R =	15/45	15/45
	0.3	**0.3**
E (β=1) =	1 - (2*0.75*0.3/(0.75+0.3))	1 - (2*0.5*0.3/(0.5+0.3))
	1 - (0.45/1.05)	1 - (0.3/0.8)
	1 - 0.43	1 - 0.375
	0.57	**0.625**

Example 2 Let TR = 30
Change number of relevant documents returned:

	Query A	Query B

RR =	**15**	**27**
P =	15/30	27/30
	0.5	**0.9**
R =	15/45	27/45
	0.3	**0.6**
E (β=1) =	1 - (2*0.5*0.3/(0.5+0.3))	1 - (2*0.9*0.6/(0.9+0.6))
	1 - (0.3/0.8)	1 - (1.08/1.5)
	1 - 0.375	1 - 0.72
	0.625	**0.28**

12. References

Almeidaa, J., Barbosa, L., Pais, A., Formosinho S., 2007, 'Improving hierarchical cluster next term analysis: A new method with outlier detection and automatic clustering', *Chemometrics and Intelligent Laboratory Systems*, Volume 87, Issue 2, p208-217.

Anderberg, M.R., 1973, 'Cluster analysis for applications', *Academic Press.*

Berrar, D.P., Dubitzky, W., Granzow, M., 2003, 'A practical approach to microarray data analysis', *Springer*, p250.

Cimiano, P., 2006, 'Ontology learning and population from text: algorithms, evaluation and applications', *Springer*, p72.

Defays, D., 1977, 'An Efficient Algorithm for a Complete Link Method', *The Computer Journal*, Volume 20, Number 4.

Dunn, G., Everitt, B., 1982, 'An introduction to mathematical taxonomy', *Cambridge Press*

El-Hamdouchi, A., Willett, P., 1989, 'Comparison of hierarchic agglomerative clustering methods for document retrieval', *The Computer Journal*, Volume 32, Issue 3.

Guan, Y., Ghorbani, A., Belacel, N., 2003, 'Y-means : A clustering method for intrusion detection', *IEEE Canadian Conference on Electrical and Computer Engineering Proceedings.*

Hughes, J., 1994, 'Automatically acquiring a classification of words', *Leeds University Press*, p73.

Kimmel, A.R., Oliver, B., 2006, DNA microarrays, Volume 411, *Academic Press*.

Liang, F., Wanga, N., 2007, 'Dynamic agglomerative next term clustering of gene expression profiles', *Pattern Recognition Letters*, Volume 28, Issue 9, p1062-1076.

Myatt, G.J., Johnson, W.P., 2009, 'Making sense of data 2', *John Wiley and Sons*, p87.

Rand, W.M., 1971, 'Objective criteria for the evaluation of clustering methods', *Journal of the American Statistical Association* Volume 66 p846–850.

Rousseeuw, J., 1987, 'Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis', *Computational and Applied Mathematics,* p53–65.

Rijsbergen van, C.J., 1979, 'Information Retrieval', 2nd edition, *Butterworth-Heinemann*.

Sibson, R., 1973, 'SLINK: An optimally efficient algorithm for the single-link cluster method', *The Computer Journal*, Volume 16, Issue 1.

Steinbach, M., Karypis, G., Kumar, V., 2008, 'A comparison of document clustering techniques', *ScientificCommons*

Tryfos P., 1997, 'Methods for business analysis and forecasting: Text and Cases', Chapter 15, *Wiley.*

Tsiptsis, K., Chorianopoulos, A., 2010, 'Data mining techniques in CRM: Inside customer segmentation', *Wiley*.

Vesanto, J., Alhoniemi, E., 2000, 'Clustering of the self-organizing map', *IEEE Transactions on Neural Networks*, Volume 11, Issue 3, p586 - 600.

Yang, C., Zhao, X., Li, N., Wang, Y., 2009, 'Arguing the Validation of Dunn's Index in Gene Clustering', *Biomedical Engineering and Informatics*, 2nd International Conference, p1-4.