Hierarchical Clustering Algorithm - A Comparative Study

Dr.N.Rajalingam
Dept. of Management Studies
Manonmaniam Sundaranar University
Tirunelveli, India

K.Ranjini
Dept of Computer Science and Engg
Einstein College of Engineering
Tirunelveli, India

ABSTRACT
Clustering is a data mining (machine learning) technique used to place data elements into related groups without advance knowledge on the group definitions. In this paper the authors provides an in depth explanation of implementation of agglomerative and divisive clustering algorithms for various types of attributes. Database - the details of the victims of Tsunami in Thailand during the year 2004, was taken as the test data. The algorithms are implemented using Visual programming and the formation of the clusters and running time needed of the algorithms using different linkages (agglomerative) to different types of data are taken for analysis.

Keywords: Agglomerative, Divisive, Clustering, Tsunami Database, Data mining

1. INTRODUCTION

Data mining is a discovery process that allows users to understand the substance of and the relationships between their data. Data mining uncovers the patterns and trends in the contents of this information. In operational or data warehouse system, the data architect and designer meticulously define entities and relationships. Data mining analyses data in different perspective: classifies the data: and summarizing it into useful information. The results of the data mining can be used to increase the effectiveness of the performance of the user. For this type of analyzing purpose data mining uses a number of techniques such as Cluster analysis, Induction, Decision trees etc. Among the technique used clustering is the most important and widely used technique.

Clustering is the most popular method that makes an attempt to separate data into disjoint groups such that same-group data points are similar in its characteristics with respect to a referral point whereas data points of different-groups differs in its characteristics. Such described groups are called as clusters. Thus clusters are comprised of several similar data or objects with respect to a referral point. Cluster is one of the most important methods in the disciplines of engineering and science, including data compression, statistical data analysis, pattern recognition, data mining, artificial intelligence, and so on. Some real time applications such as handwritten character recognition, fingerprint recognition, speech/speaker recognition, and document classification, require the use of clustering techniques in order to reduce the training data amount or to find representative data points.[1][2]

Clustering methods are broadly understood as hierarchical and partitioning clustering. A hierarchical clustering is a nested sequence of partitions. This method works on both bottom-up and top-down approaches. Based on the approach hierarchical clustering is further subdivided into agglomerative and divisive. The agglomerative hierarchical technique follows bottom up approach whereas divisive follows top-down approaches. Hierarchical clustering use different metrics which measures the distance between 2 tuples and the linkage criteria, which specifies the dissimilarity in the sets as a function of the pair-wise distances of observations in that sets. The linkage criteria could be of 3 types' single linkage, average linkage and complete linkage.[3]

In this paper, both the agglomerative and divisive hierarchical clustering algorithms with three linkages are implemented using Visual programming and tested against Tsunami Victim database. This database possesses the details about the people who were affected by Tsunami during the year 2004, in and around Thailand. Since this database contains different types of fields such as numeric, string, and binary it was chosen for the study.

2. SIMILARITY MEASURE

The basic objective of using cluster analysis is to discover natural groupings of the items (or variables). To measure the association between objects a quantitative scale is developed. These scales are referred as similarity measures and are mainly statistical measures that indicate the distances between each of the objects.

2.1 Similarity Measures for Numeric Data

An important step in any clustering is to select a distance measure, which will determine how the *similarity* of two elements is calculated. This will influence the shape of the clusters, as some elements may be close to one another according to one distance measure and may be away according to another distance measure.

For clustering Numeric field there are many well known methods such as Euclidean distance, Minkowski distance, Manhattan (City-Block), etc., but all the distance measures discussed yields the same result for 1-norm distance. So, **Euclidean Method** is selected for this research.

2.1.1 Euclidean Distance

This is the most commonly chosen type of distance. It simply is the geometric distance in the multidimensional space. [6][12][13] The Euclidean distance between points $\mathcal{P}=(p_1,p_2,\ldots p_n)$ and $\mathcal{Q}=(q1,q2,\ldots qn)$, in Euclidean *n*-space, is calculated using:

$$\sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$$

Where, p_i is the data point in x-axis
Where, q_i is the data point in y-axis

2.2 Similarity Measures for Binary Data

When items cannot be represented by meaningful p - dimensional measurements, pairs of items are often compared on the basis of the presence and absence of certain characteristics. Similar items have more characteristics in common than dissimilar items. The presence or absence of certain characteristics is described mathematically by introducing a binary variable, which assumes value 1 if the characteristic is present and value 0 if the characteristic is not present. [3]

The bit strings that characterize two objects may also be used to calculate a "distance." This effective distance may then be used with a clustering algorithm to place the objects into groups.

If the bit string has a length of L, it is possible to go down this string and count the number of times a bit is ON in both strings, ON in one and OFF in the other, or OFF in both strings. The four sums are presented in table 1.

The first subscript refers to the value of the bit for object *i* and the second refer to the value of the bit for object *j*, and it's summed to over all L bits. Therefore, B_{10} is the number of times a bit is ON in *i* and OFF in *j*.

Table 1 : Contingency table for binary variables

		Object *j*	
		0	**1**
Object *i*	**0**	B_{00}	B_{01}
	1	B_{10}	B_{11}

Some other symbols that will be used are

- B_i (= B_{10} + B_{11}) is the total number of ON bits in object *i*.
- B_j (= B_{01} + B_{11}) is the total number of ON bits in object *j*.
- B_C (= B_{11}) is the total number of times a bit is ON in both bit strings.
- B_I (= B_{00} + B_{11}) is the total number of times the two bit strings agree.
- L (= B_{00} + B_{01} + B_{10} + B_{11}) is the length of the bit string.

With these definitions, commonly used similarity metrics are Simple Matching - Sokal & Michener, Russel & Rao, Tanamoto Coefficient etc. In this paper for clustering **binary** field Simple Matching Sokal & Michener distance measure is used.

Simple Matching - Sokal & Michener is calculated using the formula

$$SM = B_I/L$$

2.3 Similarity Measures for String Data

String metrics (also known as **similarity metrics**) are a class of textual based metrics resulting in a similarity or dissimilarity (distance) score between two pairs of strings for approximate matching or comparison.

For clustering **string field** there are two well known methods

- Hamming Distance
- Levenshtein Distance

But Hamming distance has a drawback that the string must be of equal length. So **Levenshtein Distance** is chosen for clustering.

2.3.1 Levenshtein Distance (LD)

The most widely known string metric is Levenshtein Distance, also known as Edit Distance, which operates between two input strings, returning a score equivalent to the number of transpositions, substitutions and deletions needed in order to transform one input string into another.

If character string S has length N, s(i) is the character in the i^{th} position. This string can be compared to string T of length M, with t(j) representing the character in its j^{th} position. This procedure compares the difference between the characters over all positions, d(i,j), where

d(i,j) = 0 if s(i) = t(j)

= 1 if s(i) ≠ t(j)

To calculate this distance a matrix L(0:N,0:M) needs to be determined. It is an (N+1)×(M+1) matrix with elements L(i,j), where i varies from 0 to N and j varies from 0 to M. Each element is determined from the following equations. [8][9][10]

L(i,0) = i for i=0,1,….,N
L(0,j) = j for j=0,1,……,M
L(i,j) = min[L(i-1,j) +1, L(i,j-1)+1, L(i-1,j-1)+d(i.j)]

The matrix element in the lower-right corner, L(N,M), is the Levenshtein distance between strings S and T, LD(S,T).

3. HIERARCHICAL ALGORITHMS

3.1. Agglomerative Algorithm

For ***n*** samples, agglomerative algorithms [1] begin with ***n*** clusters and each cluster contains a single sample or a point. Then two clusters will merge so that the similarity between them is the closest until the number of clusters becomes 1 or as specified by the user. [4] [7][14]

1. Start with ***n*** clusters, and a single sample indicates one cluster.
2. Find the most similar clusters C_i and C_j then merge them into one cluster.
3. Repeat step 2 until the number of cluster becomes one or as specified by the user.

The distances between each pair of clusters are computed to choose two clusters that have more opportunity to merge. There are several ways to calculate the distances between the clusters C_i and C_j.

Table 2: Linkage Methods or Measuring Association d_{12} Between Clusters 1 and 2

Single Linkage	$\boldsymbol{d_{12} = {}^{min}_{ij} d(X_i, Y_i)}$	This is the distance between the closest members of the two clusters.
Complete Linkage	$\boldsymbol{d_{12} = {}^{max}_{ij} d(X_i, Y_i)}$	This is the distance between the farthest apart members.
Average Linkage	$d_{12} = \frac{1}{kl} \sum_{i=1}^{k} \sum_{j=1}^{l} d(X_i, Y_j)$	This method involves looking at the distances between all pairs and averages all of these distances

Notation:

- $X_1, X_2, \ldots, X_k$ =Observations from cluster 1
- $Y_1, Y_2, \ldots, Y_l$ = Observations from cluster 2
- d (**x**,**y**) = Distance between a subject with observation vector **x** and a subject with observation vector **y**

Methods for measuring association between clusters are called linkage methods and are presented in the table 2.[5]

3.2 Divisive Algorithms

Divisive algorithms begin with just only one cluster that contains all sample data. Then, the single cluster splits into 2 or more clusters that have higher dissimilarity between them until the number of clusters becomes number of samples or as specified by the user. The following algorithm is one kind of divisive algorithms using splinter party method.

Divisive algorithm using splinter party method

1. Start with one cluster that contains all samples.

2. Calculate diameter of each cluster. Diameter is the maximal distance between samples in the cluster. Choose one cluster C having maximal diameter of all clusters to split.

3. Find the most dissimilar sample x from cluster C. Let x depart from the original cluster C to form a new independent cluster N (now cluster C does not include sample x). Assign all members of cluster C to M_C.

4. Repeat step 6 until members of cluster C and N do not change.

5. Calculate similarities from each member of M_C to cluster C and N, and let the member owning the highest similarities in M_C move to its similar cluster C or N. Update members of C and N.

6. Repeat the step 2, 3, 4, 5 until the number of clusters becomes the number of samples or as specified by the user. [15].

4. IMPLEMENTATION

In this paper the agglomerative clustering algorithm with different linkages and divisive algorithm are implemented using Visual Programming and their performance are compared for all the basic data types and for various numbers of records.

The algorithms are implemented and tested against **Tsunami Victim** database. This database possesses the details about the people who were affected by **Tsunami** during the **year 2004,** in and around **Thailand**. Since this database contains different types of fields such as **numeric, string, and binary** it was chosen for the study. This database is downloaded from the net and it contains up to **8000 records. [11]**

4.1 Tsunami Database Structure

Tsunami Victim database contains the fields as shown in the table 3.

4.2 Implementation of Hierarchical Clustering Algorithms

Hierarchical algorithms are applied to the following **types of data fields**
1. Character or String Field
2. Numeric Field and
3. Binary Field

User may select the field based on which they require to cluster.

This database contains three numeric fields, Id, Record number, and Age. However, Id and Record number are unique fields and they cannot be clustered. So, the only remaining field Age is used for clustering.

Table 3: Tsunami Database

S.No	**Field Name**	**Type**
1.	Id	Numeric
2.	Record Number	Numeric
3.	Name	Character
4.	Sur-name	Character
5.	Sex	Binary
6.	Age	Numeric
7.	Province	Character
8.	Nationality	Character
9.	Injured / Dead	Binary

5. PERFORMANCE ANALYSIS OF THE ALGORITHMS

The performance is analyzed based on the running time needed to execute agglomerative and divisive algorithm depending on the nature of the field and the number of records.

From the database, one field in each type of data is taken for comparison. Therefore, for binary data type sex field is selected, for numeric type age field is selected and for string, province field is selected. In addition, two fields are combined together and the performance of the algorithm is compared as a special category. For that Sex and Injured/Dead fields are selected.

5.1 Comparative Performance of Hierarchical Clustering Algorithms with respect to the Nature of the Field of Reference

Different Hierarchical algorithms are compared for their performances using the time required to cluster the database based on the selected binary, numerical, and string fields and based on the combination of two (binary) fields.

5.1.1 Clustering based on Binary Field (sex)

Table 4 shows the execution times of the different clustering algorithms for varied number of records when clustering based on the sex field (Binary Field) in the database.

Table 4: Execution Time For Clustering Based On Binary Field (Sex)

Algorithms	**Execution Time (in Seconds)**			
	Number of Records			
	250	**500**	**750**	**1000**
Agglomerative –Single Linkage	9	62	227	473
Agglomerative –Complete Linkage	9	62	228	476
Agglomerative –Average Linkage	9	62	229	471
Divisive Algorithm	8	65	162	363

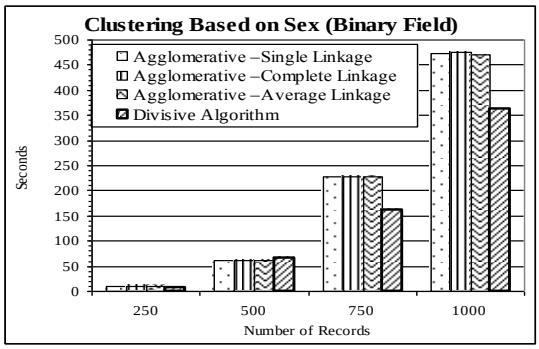

Figure 1: E**xecution time for clustering based on binary field**

It is shown from figure 1 that the execution time for clustering the database based on a binary field using the agglomerative algorithms are more or less equal and the execution time increases as the size of the database increases. However, divisive algorithm require lesser time than agglomerative algorithms when the size of the database increases.

5.1.2 Clustering based on Numeric Field (Age)

Table 5 shows the execution times of the different clustering algorithms for varied number of records when clustering based on the age field (Numeric Field) in the database.

Table 5: Execution Time for Clustering Based On Numeric Field – Age

Algorithms	Execution Time (in Seconds)			
	Number of Records			
	250	500	750	1000
Agglomerative –Single Linkage	9	60	227	467
Agglomerative –Complete Linkage	9	60	225	464
Agglomerative –Average Linkage	8	60	228	464
Divisive Algorithm	3	9	20	38

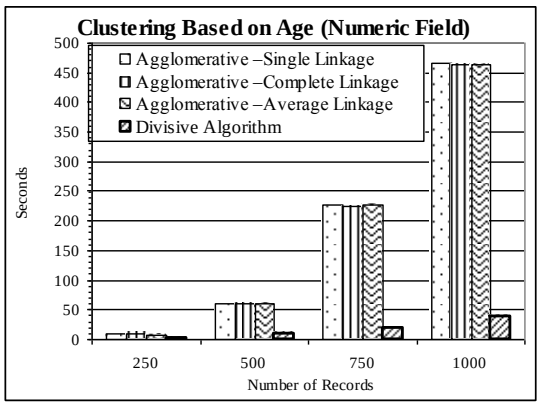

Figure 2: Execution time for clustering based on numeric field - age

It is seen from Figure 2 that divisive algorithm require very less time comparing to agglomerative algorithms when clustering the database with reference to a numeric field. Whereas agglomerative algorithms require more or less equal time to cluster the database using a numeric field and the required time increases with the increase in the size of the database.

5.1.3 Clustering based on String Field (Provinces)

Table 6 shows the execution times of the different clustering algorithms for varied number of records when clustering based on the provinces field (String Field) in the database.

Table 6: Execution Time for Clustering Based On String Field Provinces

Algorithms	Execution Time (in Seconds)			
	Number of Records			
	250	500	750	1000
Agglomerative –Single Linkage	15	93	312	597
Agglomerative –Complete Linkage	15	94	275	599
Agglomerative –Average Linkage	15	93	312	600
Divisive Algorithm	12	52	112	215

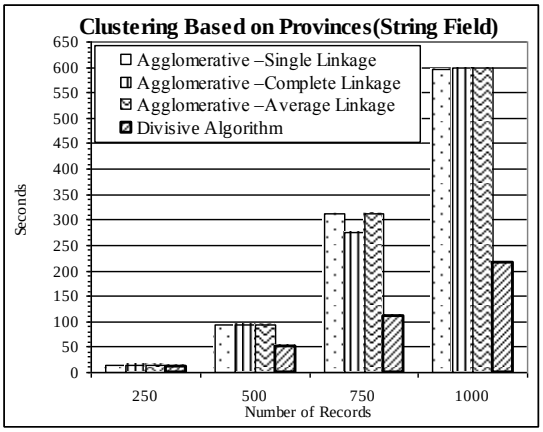

Figure 3: Execution Time for Clustering Based On String Field Provinces

It is obvious from Figure 3 that divisive algorithm require comparatively lesser time to cluster the database using the string field. Agglomerative algorithms require more or less equal time to cluster the database using a string field. The required time to cluster the database using all algorithms increases with the increase in the size of the database.

5.1.4 Clustering based on Two Binary Fields (Injured / Dead and Sex)

Table 7 shows the execution times of the different clustering algorithms for varied number of records when clustering based on the combination of two binary fields (injured / dead and sex) in the database.

Table 7 : Execution Time for Clustering Based On Two Binary Fields Injured / Dead & Sex

Algorithms	Execution Time (in Seconds)			
	Number of Records			
	250	500	750	1000
Agglomerative –Single Linkage	9	64	209	475
Agglomerative –Complete Linkage	9	64	208	477
Agglomerative –Average Linkage	10	64	209	477
Divisive Algorithm	5	36	107	262

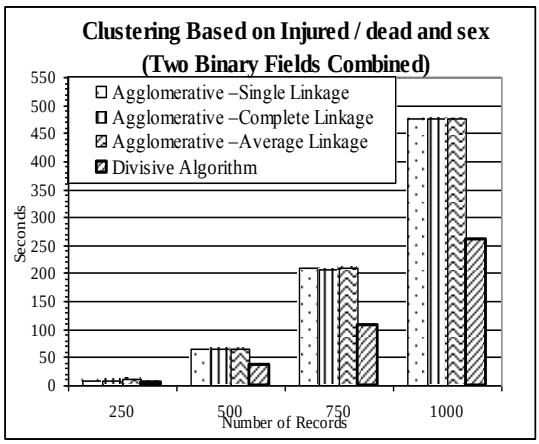

Figure 4: Execution Time for Clustering based on Injured / Dead & Sex

It is obvious from Figure 4 that divisive algorithm require comparatively lesser time to cluster the database using the combination of two binary fields. Agglomerative algorithms require more or less equal time to cluster the database. The required time to cluster the database using all algorithms increases with the increase in the size of the database.

6. CONCLUSION

This paper analyzes the performance of agglomerative and divisive algorithm for various data types. From this work it is found that the divisive algorithm works as twice as fast as that of agglomerative algorithm. It is also found that the time needed for string data type is high when compared to the other. The next observation is, in the case of binary field, the time needed to execute a two combined binary field is slightly larger or less equal to the time needed for single binary field. It is also found that the running time get increased on an average of 6 times when the number of records get doubled. More over the running time for all the agglomerative algorithms for same type of data and for same amount of records is more or less equal.

7. REFERENCES

[1] Sung Young Jung, and Taek-Soo Kim, "An Agglomerative Hierarchical Clustering Using Partial Maximum Array and Incremental Similarity Computation Method", Proceedings of the 2001 IEEE International Conference on Data Mining, p.265-272, November 29-December 02, 2001

[2] R.J. Gil-Garcia; J.M. Badia-Contelles, "A General Framework for Agglomerative Hierarchical Clustering Algorithms A Pons-Porrata Pattern Recognition, 2006. ICPR 2006. 18th International Conference on Volume 2, 2006 Page(s):569 – 572

[3] K.P.Soman, Shyam Diwakar, and V.Ajay, "Insight into Data Mining- Theory and Practice", Eastern Economy Edition, Prentice Hall of India Pvt. Ltd, New Delhi, 2006

[4] "Measuring Association d_{12} Between Clusters 1 and 2" in http://www.stat.psu.edu/online/courses/stat505/18_cluster/05_cluster_between.html

[5] Margaret H.Dunham "Data Mining Introductory and Advance Topics", Low price Edition – Pearson Education, Delhi, 2003.

[6] "Euclidean Distance" in http://people.revoledu.com /kardi/tutorial/Similarity/EuclideanDistance.html

[7] "Cluster analysis" in http://en.wikipedia.org/ wiki/Cluster_ analysis

[8] "Levenshtein_Distance" in http://en.wikipedia.org/wiki/ Levenshtein_Distance

[9] "Similarity Metrics" in http://www.dcs.shef.ac.uk/~sam/ stringmetrics.html#hamming

[10] "Levenshtein_Distance" in http://www.dcs.shef.ac.uk/ ~sam/ stringmetrics.html#Levenshtein

[11] "Tsunami victim list" http://www.ems.narenthorn. thaigov. net/tsunami_e/tsunamilist.php

[12] "Euclidean distance" in http://en.wikipedia.org/wiki/ Euclidean_distance#One-dimensional_distance

[13] "Distance" in http://en.wikipedia.org/wiki/Distance# Mathematics

[14] **"**Hierarchical Clustering Algorithms" in http://home.dei. polimi.it/matteucc/Clustering/tutorial_html/hierarchical.html

[15] Hui-Chuan Lin (2009)"Survey and Implementation of Clustering Algorithms" an Unpublished master's thesis for master's degree, Hsinchu, Taiwan, Republic of China

[16] Jiawei Han and Micheline Kamber, "Data Mining Concepts and Techniques", Second Edition- Morgan Kaufmann Publishers, San Francisco, 2006.

[17] Evangelos Petroutsos, "Mastering Visual Basic 6", BPP publications, New Delhi.

[18] Yu Zhong , Anil K. Jain , M.-P. Dubuisson-Jolly, "Object Tracking Using Deformable Templates", IEEE Transactions on Pattern Analysis and Machine Intelligence, v.22 n.5, p.544-549, May 2000.

[19] Gary Cornell, "Visual Basic 6 from the Groung Up", Tata McGraw Hill, New Delhi, 2003.