Journal of Theoretical and Applied Information Technology
20th May 2013. Vol. 51 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

TOWARDS NEW ESTIMATING INCREMENTAL DIMENSIONAL ALGORITHM (EIDA)

[1] S. ADAEKALAVAN , [2] DR. C. CHANDRASEKAR

[1]Assistant Professor, Department of Information Technology, J.J. College of Arts and Science, Pudukkottai, Tamil Nadu, India

[2]Associate Professor, Department of Computer Science, Periyar University,Salem, Tamil Nadu, India

E-Mail: [1] kingsmakers@gmail.com , [2] ccsekar@gmail.com

ABSTRACT

Hierarchical clustering is the grouping of objects of interest according to their similarity into a hierarchy, with different levels reflecting the degree of inter-object resemblance. It is an important area in data analysis and pattern recognition. In this paper, the scholar proposes a new approach for robust hierarchical clustering based on the distance function between each data object and the cluster centers. This method avoids the need to compute the distance of each data object to the cluster center. It saves running time. The experimental results showed that the best clusters were obtained using EIDA method, this suggests that this similarity measure would be applicable to biological data sets.

Keywords: *Data Mining, Clustering Analysis, Agglomerative Clustering, Hierarchical Clustering Algorithm*

1. INTRODUCTION

Clustering is the process of grouping similar objects into clusters or classes. It is an important data-exploration task used in diversified areas such as market segmentation in business, gene-categorization in biology, spatial discovery, and document classification on the web. A popular method of clustering is hierarchical agglomerative clustering (HAC). It starts with each point in separate clusters and iteratively agglomerates the closest pair of clusters in each iteration until all points belong to a single cluster. The final hierarchical cluster structure is called a dendrogram (See Figure 1), a tree like structure that shows which clusters are agglomerated at each level. Each level of a dendrogram can be evaluated by a cluster validation method and the best level and its corresponding clusters are returned.

HAC algorithms are non-parametric. They assume little about data are natural and simple in grouping objects, and capable of finding clusters of different shapes by using different similarity measures for surveys on HAC. However, they have several drawbacks. First, HAC algorithms have high time and memory complexities. For example, an efficient algorithm for centroid method (that represents each cluster by its centroid)—priority queue algorithm—has a time complexity of $O(N^2 \log N)$. Although, to apply HAC on large data, some techniques are proposed in BIRCH and CURE, they do not make the traditional HAC algorithms faster. Instead, they use approximations (e.g., summarized (BIRCH) or samples (CURE) points) so as to reduce the computational cost without losing too much of accuracy. The accuracy here is with respect to the dendrogram produced by the traditional HAC algorithms. In addition, the summaries or samples still need to run the traditional HAC algorithms to generate the dendrogram.

Journal of Theoretical and Applied Information Technology
20th May 2013. Vol. 51 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

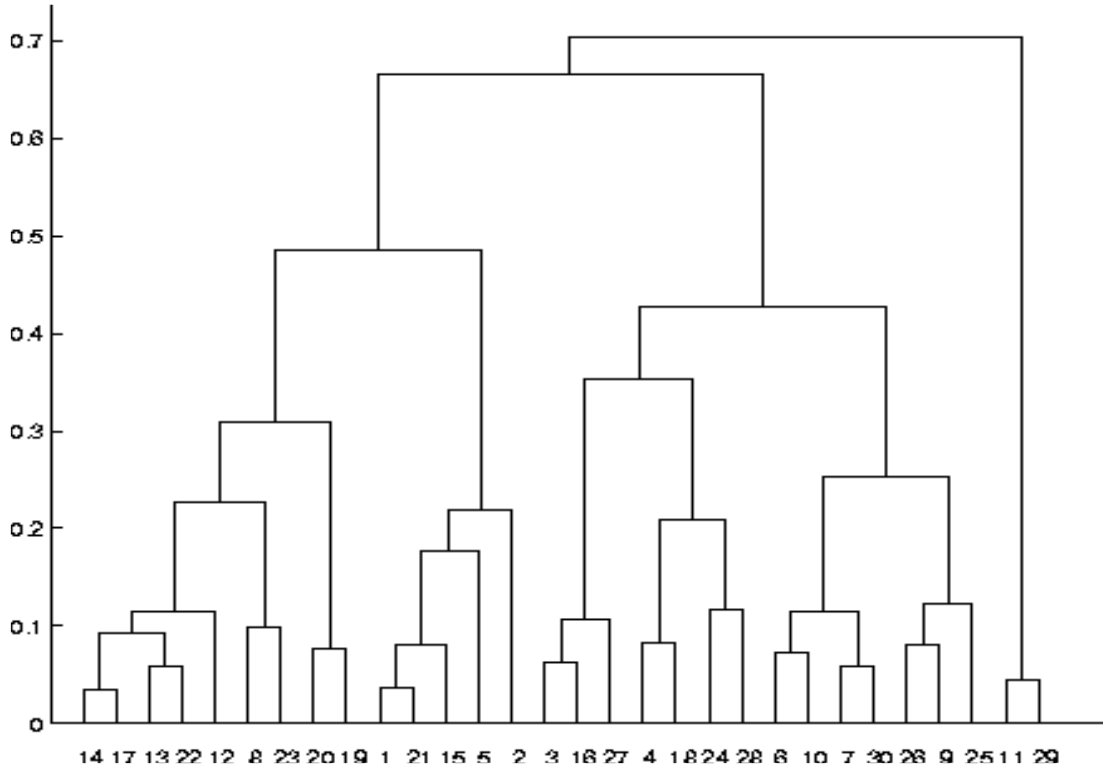

Figure – 1 : Dendrogram

Secondly, cluster validation using the dendrogram has several limitations. Cluster validation is the task of determining an 'optimal' number of clusters in a dataset. Typically, validation methods evaluate all N levels of the dendrogram to output the best level. The time complexity for evaluating all N levels is $O(N^2 \log N)$ or higher which is prohibitive for large datasets. Furthermore, many validation methods show some distracting patterns for lower levels of the dendrogram leading to inaccurate estimation of optimal number of clusters.

2. AN OVERVIEW OF HIERARCHICAL AGGLOMERATIVE CLUSTERING ALGORITHMS:

A wide range of hierarchical agglomerative clustering algorithms have been proposed at one time or another. Such hierarchical algorithms may be conveniently broken down into two groups of methods. The first group is that of linkage methods—the single, complete, weighted, and unweighted average linkage methods. These are methods for which a graph representation can be used.

The second group of hierarchical clustering methods are those which allow the cluster centers to be specified (as an average or a weighted average of the member vectors of the cluster). These methods include the centroid, median, and minimum variance methods.

The centroid may be specified either in terms of dissimilarities, alone, or alternatively in terms of cluster-center coordinates and dissimilarities. A very convenient formulation, in dissimilarity terms, which embraces all the hierarchical methods mentioned so far, is the Lance–Williams dissimilarity update formula. If points (objects) i and j are agglomerated into cluster i∪j, then we must simply specify the new dissimilarity between the cluster and all other points (objects or clusters).

Hierarchical agglomerative clustering algorithms are run once and they create a dendrogram, which is a tree structure containing a k-block set partition for each value of k between 1 and n, where n is the number of data points to cluster. These algorithms not only

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

allow a user to choose a particular clustering granularity, but in many domains clusters naturally form a hierarchy; that is, clusters are part of other clusters such as in the case of phylogenetic (evolutionary) trees. The popular hierarchical agglomerative clustering algorithms are easy to implement as they just begin with each point in its own cluster and progressively join two closest clusters to reduce the number of clusters by 1 until k = 1. The basic hierarchical agglomerative clustering algorithm considered in this paper is shown in Figure. 2. However, these added benefits come at the cost of efficiency since a typical implementation with symmetric distances uses $O(mn^2)$ time and space, where m is the number of attributes used to represent each instance. For large data sets, where the space needed to store all pairwise distances is prohibitively large, distances may need to be recomputed at each level of the dendrogram, thus leading to a running time of $O(n^3)$. Typically, a single application of a non-hierarchical clustering algorithm has a better asymptotic running time than a hierarchical clustering algorithm.

The question is how hierarchical agglomerative clustering algorithms can be modified to satisfy all instance-level cluster-level constraints. These classes of constraints restrict the set of possible clustering. An instance-level constraint specifies a condition to be satisfied by two different instances in any valid clustering. A cluster- level constraint specifies a condition to be satisfied by a single cluster or a pair of clusters in any valid clustering. The present scholar believes that his work is the first to modify hierarchical agglomerative clustering using instance-level constraints.

The Traditional Hierarchical Agglomerative Clustering Algorithms is as follows

Note : In the following algorithm, the term " Closest pair of clusters" refers to a pair of distinct clusters which are separated by the smallest distance (Euclidian Distance Matrix) over all pairs of distinct clusters.

Input : $S = \{x_1, x_2, x_3, \ldots\ldots\ldots x_n\}$ of points (instances).

Output : $Dentrogram_k$ for each k, $1 \le k \le n = |S|$.

Notation : For any pair of Clusters pi_i and pi_j,
$i \neq j$, the distance between pi_i and pi_j, is denoted by $d(i,j)$.

1. $pi_i = \{x_i\}, 1 \le i \le n. Dentrogram_n = \{pi_1, pi_2, pi_3, \ldots\ldots\ldots pi_n\}$.

2. ***for*** $k = n - 1$ ***down to*** 1 ***do***

   (a) /* Find a closest pair of clusters.*/

   Let $(a, b) = argmin_{(i,j)} \{d(i,j): 1 \le i \le j \le k+1\}$.

   (b) Obtain $Dendrogram_k$ from $Dendrogram_{k+1}$
   by merging pi_b into pi_a and then deleting pi_b

endfor

Figure – 2 : Traditional Hierarchical Agglomerative Clustering Algorithms

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3. PROPOSED EFFICIENT ESTIMATING INCREMENTAL DIMENSIONAL ALGORITHM (EIDA)

Many studies have addressed the topic of non-hierarchical clustering under constraints, with the goal of satisfying all or a maximum number of constraints. Some researchers have also used constraints to examine a distance function in the learnt metric space-points involved. So far, only two studies (those by Davidson and Ravi 2005b; Klein et al. 2002) have examined the general purpose use of instance-level constraints in hierarchical clustering. In the first, Banerjee et al. (2002) investigate the problem of learning a distance matrix (which may not be metric) that satisfies all the constraints. The aim of their work is to produce a distance matrix so that must-linked points are closed cluster together and cannot be linked points are far apart; This matrix can then be "plugged" into any hierarchical clustering algorithm. The Distance algorithm runs as follows

CREATE DISTANCE MATRIX

Input : A set $S = \{x_1, x_2, x_3, \ldots\ldots\ldots x_n\}$ of data points (instances), a set C = of must link and a set $C_{\pm}$ of cannot link constraints

Output : Modified Distance Matrix D

1. Initialize k center in the problem space
2. Compute the sigmoid function S

$$S = \frac{1}{1+e^{-a(x-c)}}$$

3. Find closest centroid by

$$J = \left(\sum_{i=1}^{s} \| x_i^{(j)} - c_j \|^2\right)^{1/2} \qquad j=1.....k$$

4. Update new centroid with respect to D (Distance)

$$c_j = \sum_{i=1}^{n} \frac{X_i^{(j)}}{n}, j = 1 \ldots\ldots k$$

Figure – 2 : Proposed Efficient Estimating Incremental Dimensional Algorithm (Eida)

Journal of Theoretical and Applied Information Technology
20th May 2013. Vol. 51 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

To make a point weighty and more robust, it should meet the weight of the abnormal points and noise points should be fewer, and the weight of the compact point with data concentration should be greater. The measurement used in this study precisely meets this requirement.

4. NUMERICAL RESULTS AND ANALYSIS :

The present study experimentally evaluated the performance of the various clustering methods to obtain hierarchical solutions using a number of different datasets. The various datasets used and our the experimental methodology followed and the results obtained are listed below and later presented in a graph.

Table – 1 : Table of Experimental Result

Number of Data Points	Alpha (a) Value	Running Time for Standard Hierarchical Agglomerative Clustering (HAC) Algorithm (Sec)	Running Time for Estimating Incremental Dimensional Algorithm (Sec)
2000	100	0.04	0.03
4000	100	0.158	0.128
6000	100	0.326	0.29
8000	100	0.568	0.53
10000	100	0.931	0.807

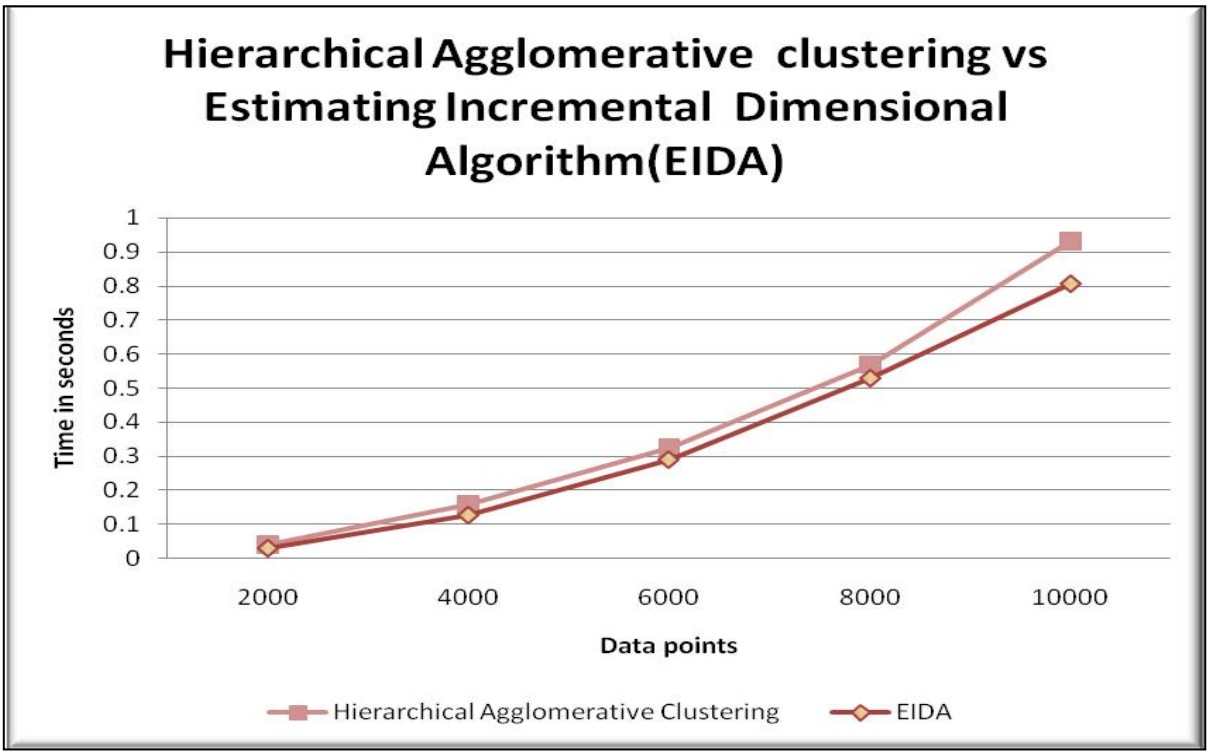

Figure – 3 : Graphical Representation of Experimental Result

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In the suggested Estimating Incremental Dimensional Algorithm (EIDA) running time is lower compared to those in the Standard Hierarchical Agglomerative Clustering Algorithms. It can generate the final clustering results in a relatively short period of time thus it enhances the speed of clustering.

5. CONCLUSION

This paper investigated the problem of clustering sequences. Sequences with items in different orders are considered different sequences. In order to reduce time a new distance measure of similarity between sequences was proposed in it the greater the number of sequence elements that are common to two comparable sequences, the higher is the similarity. A hierarchical clustering algorithm was developed for determining the similarity measure. Thus clustering algorithm generated better-quality clusters than customary standard clustering algorithms. The proposed algorithm can be used in applications requiring clustering of large sets of numerical data, sequence-data and also on text and web document collections.

REFERENCES :

[1] Bae E, Bailey J (2006) COALA: a novel approach for the extraction of an alternate clustering of high quality and high dissimilarity. In: Proceedings of the 6th IEEE international conference on data mining (ICDM 2006), Hong Kong, Dec 2006, pp 53–62

[2] Basu S, Banerjee A, Mooney R (2002) Semi-supervised clustering by seeding. In: Proceedings of the 19th international conference on machine learning (ICML 2002), Sydney, Australia, Jul 2002, pp 27–34

[3] Basu S, Bilenko M, Mooney RJ (2004) Active semi-supervision for pairwise constrained clustering. In: Proceedings of the 4th SIAM international conference on data mining (SDM 2004), Lake Buena Vista, FL, Apr 2004, pp 333–344

[4] Davidson I, Ravi SS (2005a) Clustering with constraints: feasibility issues and the *k*-means algorithm. In: Proceedings of the SIAM international conference on data mining (SDM 2005), Newport Beach, CA, Apr 2005, pp 138–149

[5] Davidson I, Ravi SS (2005b) Agglomerative hierarchical clustering with constraints: theoretical and empirical results. In: Proceedings of the 15th european conference on principles and practice of knowledge discovery in databases (PKDD 2005), Porto, Portugal, Oct 2005, pp 59–70

[6} Nanni M (2005) Speeding-up hierarchical agglomerative clustering in presence of expensive metrics. In: Proceedings of the 9th pacific asia conference on knowledge discovery and data mining (PAKDD 2005), Hanoi, Vietnam, May 2005, pp 378–387

[7] Schaefer TJ (1978) The complexity of satisfiability problems. In: Proceedings of the 10thACMinternational symposium on theory of computing (STOC 1978), San Diego, CA, May 1978, pp 216–226

[8] Wagstaff K, Cardie C (2000) Clustering with instance-level constraints. In: Proceedings of the 17th international conference on machine learning (ICML 2000), Stanford, CA, Jun–Jul 2000, pp 1103–1110

[9] Wagstaff K, Cardie C, Rogers S, Schroedl S (2001) Constrained K-means clustering with background knowledge. In: Proceedings of the 18th international conference on machine learning (ICML 2001),

[10] Davidson I, Ravi SS (2009) Using instance – level constraints in Agglomerative hierarchical clustering : Theoretical and empirical results, In: Springer Science, May - 2008.Data Mining Knowledge Discovery (2009) 18 : pp 257 - 282

[11] Williamstown, MA, Jun–Jul 2001, pp 577–584 Xing E, Ng A, Jordan M, Russell S (2002) Distance metric learning with application to clustering with side-information. In: Advances in neural information processing systems (NIPS 2002), Dec 2002, vol 15. Vancouver, Canada, pp 505–512

[12] Zho Y, Karypis G (2005) Hierarchical clustering algorithms for document datasets. Data Min Know Disc 10(2):141–168