2011 2nd International Conference on Biotechnology and Food Science
IPCBEE vol.7 (2011) © (2011) IACSIT Press, Singapore

A DYNAMIC CLUSTERING TECHNIQUE USING MINIMUM-SPANNING TREE

V. M. K. Prasad Goura[1], N. Madhusudana Rao[2], M. Rajasekhar Reddy[3]
[1]Department of Computer Science and Engineering,
Columbia Institute of Engineering and Technology, Raipur (C.G)-493111, India
[2] School of Computing, Sastra University, Tanjavoor-613402, India
[3] Department of Computer Science and Engineering,
Srinivasa Ramanujan Centre Sastra University, Kumbakonam, Tamilnadu-612001,India
E-mail:gvmprasad160@yahoo.com,madhu031083@gmail.com,rajasekharmanyam04@gmail.com

Abstract— **Clustering technique is one of the most important and basic tool for data mining. In this paper, we present a clustering algorithm that is inspired by minimum spanning tree. Given the minimum spanning tree over a data set, selects or rejects the edges of the MST in process of forming the clusters, depending on the threshold value. The Algorithm is invoked repeatedly until all the clusters are fully formed. We present experimental results of our algorithm on some synthetic data sets as well as real world data sets.**

Keywords— Clustering algorithm, MST, validity index.

I. INTRODUCTION

Clustering is a powerful tool that plays a significant role for data analysis in various fields such as data mining [1], computational biology [2] and many more. Gene expression data analysis has been extensively studied for finding genes with similar expression patterns (co-expressed genes) from DNA microarray data [3]. A number of clustering algorithms [4, 5, 6] have been developed, they usually face some problems for real situations. For example, K-means [17] is very simple and robust; however, it requires users to provide the number of clusters, which is usually unknown in advance [7]. Similarly, the Density-based Hierarchical Clustering (DHC) [22] suffers from the computational complexity which makes it inefficient for large data sets. Moreover, there is no versatile algorithm that can handle all types of clustering problems due to the arbitrary shape, variable densities and unbalanced sizes [8]. As a result, an enormous attention has been created to design more effective clustering algorithms in the recent decades.

Among various graph-based clustering methods, minimum spanning tree (MST) has been paid more attention for its intuitive and effective data representation. MST has been extensively studied for biological data analysis [10], image processing [11, 12], pattern recognition [13] and outlier detection [14]. Cluster design using MST was initiated by Zahn [15]. The approach used in their paper, is to construct MST over the given dataset and then remove the inconsistent edges to create connected components. The repeated application of this approach eventually leads to different clusters that are represented by the connected components. Xu et al. [10] proposed three approaches, i.e., clustering by removing longest MST-edges, an iterative clustering and a globally optimal clustering. Although, the methods of Zhan and Xu are effective, users do not know how to select the inconsistent edges for their removal without any prior knowledge of the structure of the data patterns. The approach used in [13] is based on maximizing or minimizing the degree of the vertices. But the method is computationally expensive. Recently, the authors of this paper have proposed a MST-based clustering algorithm in [16]. Wang et al. [9] proposed a divide and conquer algorithm that uses the cut and the cycle property of MST. Zhong et al. [21] reported a two rounds minimum spanning tree based clustering algorithm. However, the algorithm is not robust to detect the outliers and overlapped clusters. Gryorash et al. proposed two MST based clustering algorithms that can be found in [18]. Their first algorithm produces *k*-partition of a set of data points for a given value of *k* which is difficult to predict for unknown data sets. Their second algorithm produces the clusters by removing the inconsistent edges from the MST by maximizing the overall standard deviation which is not usually an efficient measure to judge the inconsistency. A rich literature has been developed over the past decades on clustering analysis, a survey of which can be found in [23].

Motivated with this, we propose here, a new clustering algorithm based on MST in which we incorporate coefficient of variation as a measure of inconsistency. Our algorithm overcomes some of the problems faced by the classical and the above mentioned MST based algorithms. The algorithm does not depend on any prior knowledge of the parameters such as number of clusters, initial cluster centres and the dimensionality of the data sets. The experimental results on some already known two dimensional data sets are shown for visual purpose. The results on the multidimensional real world data sets are also compared with the classical K-means algorithm and shown along with their corresponding values of the intra vs. inter ratio validity index.

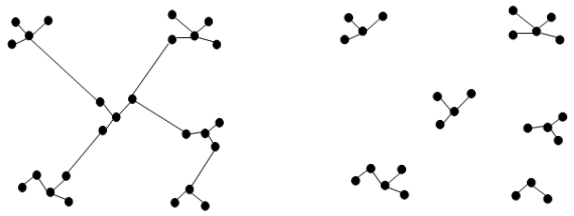
The rest of the paper is organized as follows. The concept of MST is presented in section 2. The proposed algorithm is described in section 3. The experimental results are shown in section 4 followed the conclusion in section 5.

II. RELATED WORK

A. Minimum Spanning Tree

Minimum Spanning Tree (MST) is a subgraph that spans over all the vertices of a given graph without any cycle

and has minimum sum of weights over all the included edges. In MST-based clustering analysis, the weight for each edge, is usually considered as the Euclidean distance between the end points forming that edge. As a result, any edge that connects two subtrees in a minimum spanning tree must be the shortest in distance. In such clustering methods, inconsistent edges which are unusually longer are removed from the MST. The connected components (subtrees) of the MST obtained by removing these edges are treated as the clusters. Each subtree is also a minimum spanning tree. Elimination of the longest edge results into two-group clustering. Removal of the next longest edge results into three-group clustering. Again next removal of the longest edge results into four-group clustering and so on [17]. Six-group clustering after removal of successive longest edges (five) is shown in Fig. 1.



(a) (b)

Fig. 1 Minimum spanning tree representation and six-group clustering.

B. *Validity Index*

Validity index (*Val_Index*) is used to evaluate the quality of Validity index is generally used to evaluate the clustering results quantitatively. In this paper we focus on the validity index, which is based on compactness and isolation. Compactness measures the internal cohesion among the data elements whereas isolation measures separation between theclusters [17]. We measure the compactness by Intra-cluster distance and separation by Inter-cluster distance, which are defined as follows.

Intra-cluster distance: This is the average distance of all the data points within a cluster from the cluster centre given by

$$intra\_dist = \frac{1}{N}\sum_{i=1}^{K}\sum_{x\in C_i}\left\|x - c_i\right\|^2 \quad (1)$$

Inter-cluster distance: This is the minimum of the pair wise distance between any two cluster centres given by

$$inter\_dist = Min\left\|c_i - c_j\right\|^2 \quad (2)$$

Where i = 1, 2,..., k - 1; $j = i + 1, i + 2, i + 3, ..., k$

Many validity indices have been proposed that are based on intra vs. inter distance as they are quite natural to measure the compactness and separation of the data points. In our proposed algorithm, we use the validity index proposed by Ray and Turi [20] which is also based on intra vs. inter distance defined by

$$val\_index = \frac{intra\_dist}{inter\_dist} \quad (3)$$

Threshold value: This denotes the limit when two points get disconnected if the distance between them is greater than this limit.

III. PROPOSED ALGORITHM

The basic idea of our proposed algorithm is as follows. We first construct MST using Prim's algorithm and then set a threshold value and a step size. We then remove those edges from the MST, whose lengths are greater than the threshold value. We next calculate the ratio between the intra-cluster distance and inter-cluster distance using equation (3) and record the ratio as well as the threshold. We update the threshold value by incrementing the step size. Every time we obtain the new (updated) threshold value, we repeat the above procedure. We stop repeating, when we encounter a situationsuch that the threshold value is maximum and as such no MST edges can be removed. In such situation, all the data points belong to a single cluster. Finally we obtain the minimum value of the recorded ratio and form the clusters corresponding to the stored threshold value. Therefore, the proposed algorithm searches for that optimum value of the threshold for which the Intra-Inter distance ratio is minimum. It needs not to mention that this optimum value of the threshold must lie between these two extreme values of the threshold. However, in order to reduce the number of iteration we never set the initial threshold value to zero.

***Notations Used*:**

N: Number of data points;

Step_size: Step size by which to increment the threshold after each iteration.

ARatio(N): Array which holds ratio (equation 3) after each iteration.

AThreshold(μ): Array which holds threshold value after each iteration.

T[*N*][*N*]: Adjacency matrix returned by Prim's algorithm.

Storage[2(N-1)][D+1]: Matrix used to store the end points of
the edges that are removed from MST .

Counter1: Variable to keep track of the rows of the Storage matrix.

Index[N]: Array to hold the cluster number, a data point belongs to.

Store[N - 1]: Array used to store connected components.

Cluster_no[ ]: Stores the number of cluster center.

Counter2, Counter3, Cluster_no, iteration: Temporary
variable.

d(i, j): denotes Euclidean distance between data points i and j.

Prim's Algorithm:

Let E be the set of edges in the graph G and Cost [1:N][1:N] is adjacency of an N vertex graph such that cost[i, j] is either a positive real number or ∞ if no edge (i,j) exists.

A minimum spanning is constructed and stored as a set of edges in the array. The final cost is returned

```
Let (k, l) be an edge of minimum cost in E
mincost = cost [k, l]
T[1, 1]= k, T[1, 2]=l
For i=1 to N
  If (cost (i, l) < cost (i, k))
     Then near (i)=l
  Else near (i)=k
near (k)=near (l)=0;
for i=2 to n-1 do
{
   Let j be an index such that near(j)≠0 and cost(j,near(j) is
   minimum)
   T (i,1)=j; T(i,2)=near(j)
   mincost = mincost + cost(j, near(j));
   near (j)=0
   for k=1to N do
   {
      If(near(k≠0) &&(cost(k, near(k)) > cost(k,j) ))
      Then near(k)=j
    }
   return mincost
}
 val_index[iteration]:=intra_dist/inter_dist;
iteration:= iteration + 1
for i := 1 to N do
                if (Index[i] ≠ 1)
                   {
                     Signal := 0
                   }
                else
                   {
                     break
                   }
          endfor
          if (i = N+1)
            {
              Signal:= 1
            }
          Threshold:= Threshold + Step_size
```

We now formalize our algorithm stepwise as follows:

Step 1: Given N data points, construct the MST using Prim's algorithm.

Step 2: Store all the distinct edge weights (Euclian distance) in the array edge list.

Step 3: Assign first element of the array edge list to μ and delete the element from the array.

Step 4: Find the clusters by removing all the edges of the MST (constructed in step 1) whose length is more than μ.

Assume the number of clusters formed in this step is p.

Step 5: Find the p centers of the clusters formed in step 4.

Step 6: Pass the p centers found in step 5 as the new data points from 2 to p and obtain the *Val_index* and store the minimum one in the array Min_V.

Step 7: Assign the next element of the array edge list to μ and delete the element from the array.

Step 8: If the array edge list is not exhausted then go to Step 4.

Step 9: Find the minimum value of the array MIN_V and obtain the final clusters corresponding to this value.

Step10: stop.

Time Complexity: It can be seen that the proposed algorithm consists of two phases. In the first phase, we construct the MST using Prim's algorithm that requires $O(N^2)$ time. In the second phase we find the connected components by removing those edges from MST whose edge list (length) is greater than the chosen threshold. Therefore, the overall time complexity is $O(kN^2)$. This is comparable with k-means.

IV. EXPERIMENTAL RESULTS

In order to test our algorithm, we consider several synthetic as well as real world data sets.

A. Results on synthetic data sets

For visualization purpose, we considered to run our algorithm on two dimensional data sets. The first one is a simple data set having 1890 data points. It resulted in 5 different clusters as shown in Fig. 2. It was next tested on two complex clustering problems for kernel data set of 1100 points and the curve data set having 664 points. The results are shown in Figs. 3 and 4 respectively.

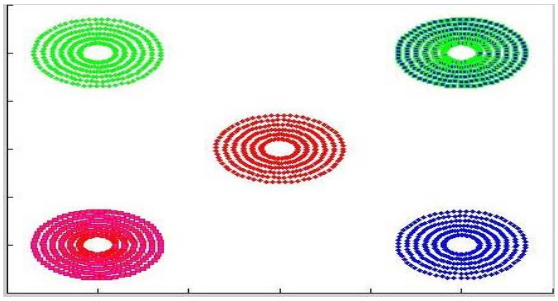


Fig.2 Result of proposed algorithm on synthetic data of size 1890 points

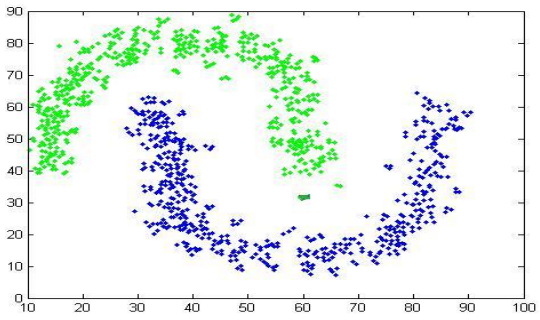


Fig.3 Result of proposed algorithm on banana data of size 1000 point

B. Results on the real world data sets

For the real world data sets, we have experimented on Wine, Spec Heart and Ecoli, data sets. The description of all these data sets can be seen from UCI machine learning repository [19]. Their results are shown as depicted in Table 1.

Name	Data size	Cluster No.	Ratio(K-means	Ratio(Ours)
Wine	**150**	**3**	**0.1895**	**0.1153**
Spec Heart	**187**	**2**	**1.8806**	**0.5101**
Ecoli	**336**	**8**	**1.3412**	**0.9835**

Table 1 :Clustering Results On Wine, Spec Heart,And Ecoli Data Sets

The above results show that the performance of our algorithm is better than the classical K-means algorithm.

V. CONCLUSION

A clustering algorithm based on the minimum spanning tree has been presented. The algorithm has been shown to be very effective in clustering multidimensional data sets. The algorithm has been tested on synthetic data sets as well as real world data sets such as Wine, Spec heart and Ecoli from UCI machine learning repository. Their experimental results clearly show that it performs better than the classical K-means algorithm.

REFERENCES

[1] J.Han, M.Kamber, Data mining: Concepts andTechniques, Morgan-Kaufman, 2006.

[2] Z. Yu, H. S. Wong, H. Wang, Graph-based consensus clustering for class discovery from gene expression data, Bioinformatics 23(2007) 2888-2896.

[3] Xu Rui, S. Damelin, B. Nadler, and D. C. Wunsch, Clust-ering of High-Dimensional Gene Expression Data with Feature Filtering Methods and Diffusion Maps, in: Proc. of Intl. Conf. on Biomedical Engg. and Informatics(BE-MI), 2008, pp. 245 -249.

[4] G.Kerr, H.J. Ruskin, M. Crane and P. Doolan,Techniques for clustering gene expression data, Computers in Biology and Medicine, 38(2008) 283-293.

[5] Zhihua Du,YiweiWang, Zhen Ji, PK-means: A new algorithm for gene clustering, Computational Biology and Chemistry, 32(2008) 243–247.

[6] S.Seal, S.Komarina and S. Aluru, An optimal hierarchical clustering algorithm for gene expression data, Information Processing Letters, 93(2005) 143-147.

[7] Judong Shen, Shing I. Chang, E. Stanley Lee, Youping Deng, and Susan J. Brown, Determination of cluster number in clustering microarray data, J. Applied Mathe-matics and Computation, 169(2005) 1172 -1185.

[8] R. Xu, D Wunsh II, Survey of clustering algorithms, IEEE Trans. on Neural Networks, 15(2005) 645-678.

[9] Xiaochun Wang, Xiali Wang and D. Mitchell Wilkes, A Divide – and - conquer approach for minimum spanning Tree - based clustering, IEEE Trans. On Knowledge and Data Engg., 21(2009) 945-958.

[10] Y. Xu, V. Olman and D. Xu, Clustering gene expression data using a graph-theriotic approach: An application of minimum spanning trees, Bioinformatics, 18(2002) 536-545.

[11] Zhi Min Wang, Yeng Chai Soh, Qing Song, Kang Sim, Adaptive spatial information - theoretic clustering for image segmentation, Pattern Recognition, 42(2009) 2029-2044.

[12] Li Peihua, A clustering-based color model and integral images for fast object tracking, Signal Processing: Image Communication, 21(2006) 676-687.

[13] N. Paivinen, Clustering with a minimum spanning tree of scale-free-like structure, Pattern Recognition Letters,*26* (2005) 921-930.

[14] J. Lin, D. ye, C. Chen and M. Gao, Minimum spanning tree based special outlier minimg and its applications, Lecture Notes in Computer Science, Springer-Verlag, 509 (2008) 508-515.

[15] C. T. Zahn, Graph-theriotical methods for detecting and describing getalt clusters, IEEE Trans. on Computers,20 (1971), 68-86.

[16] P. K. Jana, Azad Naik, An efficient minimum spanning tree based clustering algorithm, in: Proc. of Intl.Conference on Methods and Models in Computer Science (ICM2CS09), 2009, pp. 1-5.

[17] A.K. Jain and R.C. Dubes, Algorithms for Clustering, prentice Hall, 1988.

[18] O. Grygorash, Y. Zhou ands Z, Jorgenssn, Minimum spanning tree - based clustering algorithms, in: Proc. IEEE Int'l Conf. on Tools with Artificial Intelligence, 2006, pp. 73-81.

[19] UCI Machine Learning Repository, http://archive.ics.uci.edu/ ml/ datasets.html.

[20] S. Ray, and R. H. Turi, Determination of number of Clusters in K-Means Clustering and application in Colour image Segmentation, in: Proc. 4th Intl. Conf.(ICAPRDT99), 1999, pp. 137-143.

[21] C. Zhong, D. Miao, R. Wang, A graph - theoretical clustering method based on two rounds of minimum spanning trees, Pattern Recognition, 43(2010), 752-766.

[22] Zhiqiang Xie, Liang Yu, Jing Yang,A clustering algorithm based on improved minimum spanning tree, in: Proc.of Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007), 2007, pp. 149 –152.

[23] Kerr, G., Ruskina, H.J., Crane, M., Doolan, P.: Techniques for Clustering Gene Expression Data. Computers in Biology and Medicine. 38, 283-293, (2008).

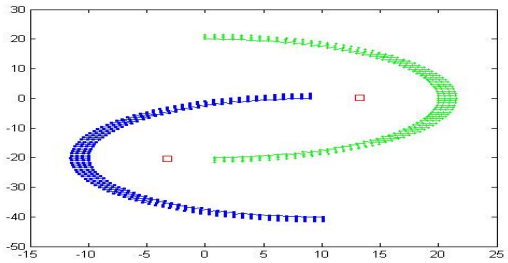


Fig. 4 Result on curve data set of 646 points.

M.Rajasekhar Reddy
Assistant Professor
S.R. C. Sastra University,
Kumbakonam.
rajasekharmanyam04@gmail.comm

N. Madhusudana Rao
Assistant Professor
Sastra University, Tanjavoor.
madhu031083@gmail.com

V. M. K. Prasad Goura
[1] Assistant Professor
Columbia I.E.T, Raipur.
gvmprasad160@yahoo.com