A NOVEL ALGORITHM FOR MINIMUM SPANNING CLUSTERING TREE

[1]S.JOHN PETER , [2]S.P.VICTOR
[1.] Assistant Professor [2.] Associate Professor
Department of Computer Science and Research Center
St. Xavier's College, Palayamkottai
Tamil Nadu, India.
E-mail: jaypeeyes@rediffmail.com

ABSTRACT

The minimum spanning tree clustering algorithm is capable of detecting clusters with irregular boundaries. In this paper we propose minimum spanning tree based clustering algorithm. The algorithm produces k clusters with Minimum Spanning Clustering Tree (**MSCT**), a new data structure which can be used as search tree. Our algorithm works in two phases. The first phase produces subtree (cluster). The second phase converts the subtree into binary tree called **MSCT**.

Key Words: *Minimum Spanning Clustering trees, Euclidean Minimum Spanning Tree, Clusters, Subtree*

1. INTRODUCTION

Given a connected, undirected graph $G = (V, E)$, where V is the set of nodes, E is the set of edges between pairs of nodes, and a weight $w (u , v)$ specifying weight of the edge (u, v) for each edge $(u, v) \in E$. A spanning tree is an acyclic subgraph of a graph G, which contains all vertices from G. The Minimum Spanning Tree (**MST**) of a weighted graph is minimum weight spanning tree of that graph. Several well established **MST** algorithms exist to solve minimum spanning tree problem [12, 7, 8] with cost of constructing a minimum spanning tree is $O (m \log n)$, where m is the number of edges in the graph and n is the number of vertices. More efficient algorithm for constructing **MST**s have also been extensively researched [6, 3, 4]. These algorithms promise close to linear time complexity under different assumptions. A Euclidean minimum spanning tree (**EMST**) is a spanning tree of a set of n points in a metric space ($\mathbf{E^n}$), where the length of an edge is the Euclidean distance between a pair of points in the point set.

The hierarchical clustering approaches are related to graph theoretic clustering. Clustering algorithms using minimal spanning tree takes the advantage of **MST**. The **MST** ignores many possible connections between the data patterns, so the cost of clustering can be decreased. The **MST** clustering algorithm is known to be capable of detecting clusters with irregular boundaries [13]. Unlike traditional clustering algorithms, the **MST** clustering algorithm does not assume a spherical shapes structure of the underlying data. The **EMST** clustering algorithm [11, 13] uses the Euclidean minimum spanning tree of a graph to produce the structure of point clusters in the n-dimensional Euclidean space. Clusters are detected to achieve some measures of optimality, such as minimum intracluster distance or maximum inter-cluster distance [1]. The **EMST** algorithm has been widely used in practice. Once the **MST** is built for a given input, there are different ways to produce groups of clusters. If the number of clusters k is given in advance, the simplest way to obtain k clusters is to sort the edges of minimum spanning tree in descending order of their weights, and remove edges with first k-1 heaviest weights [1, 15].

All existing clustering Algorithm require a number of parameters as their inputs and these parameters can significantly affect the cluster quality. In this paper we want to avoid experimental methods and advocate the idea of need-specific as opposed to care-specific because users always know the needs of their applications. We believe it is a good idea to allow users to define their desired similarity within a cluster and allow them to have some flexibility to adjust the similarity if the adjustment is needed. Our Algorithm produces clusters of n-dimensional

points with a given cluster number and a naturally approximate intra-cluster distance.

In this paper we have used **EMST** based clustering algorithm to address the issues of undesired clustering structure and unnecessary large number of clusters. The Algorithm assumes the number of clusters is given. The algorithm constructs an **EMST** of a point set and removes the inconsistent edges that satisfy the inconsistence measure. The process is repeated to create a hierarchy of clusters until *k* clusters are obtained. In section 2 we review some of the existing works on graph based algorithm. In Section 3 we present **MSCT** algorithm which produces *k* clusters with Minimum Spanning Clustering Tree. Finally in conclusion we summarize the strength of our methods and possible improvements.

2. RELATED WORK.

Clustering by minimal spanning tree can be viewed as a hierarchical clustering algorithm which follows the divisive approach. Clustering Algorithm based on minimum and maximum spanning tree were extensively studied. Avis [2] found an $O(n^2 \log^2 n)$ algorithm for the min-max diameter-2 clustering problem. Asano, Bhattacharya, Keil and Yao [1] later gave optimal $O(n \log n)$ algorithm using maximum spanning trees for minimizing the maximum diameter of a bipartition. The problem becomes NP-complete when the number of partitions is beyond two [5]. Asano, Bhattacharya, Keil and Yao also considered the clustering problem in which the goal to maximize the minimum intercluster distance. They gave a *k*-partition of point set removing the *k*-1 longest edges from the minimum spanning tree constructed from that point set [1]. The identification of inconsistent edges causes problem in the **MST** clustering algorithm. There exist numerous ways to divide clusters successively, but there is not a suitable choice for all cases.

Zahn [16] proposes to construct an **MST** of point set and delete inconsistent edges – the edges, whose weights are significantly larger than the average weight of the nearby edges in the tree. Zahn's inconsistent measure is defined as follows. Let *e* denote an edge in the **MST** of the point set, v_1 and v_2 be the end nodes of *e, w* be the weight of *e*. A *depth neighborhood* ***N*** of an end node *v* of an edge *e* defined as a set of all edges that belong to all the path of length *d* originating from *v*, excluding the path that include the edge *e*. Let N_1 and N_2 be the depth *d* neighborhood of the node v_1 and v_2. Let $\hat{W}_{N1}$ be the average weight of edges in N_1 and σN_1 be its standard deviation. Similarly, let $\hat{W}_{N2}$ be the average weight of edges in N_2 and σN_2 be its

standard deviation. The inconsistency measure requires on of the three conditions hold:

1. $w > \hat{W}N_1 + c \text{ x } \sigma N_1$ or $w > \hat{W}N_2 + \text{c x } \sigma N_2$

2. $w > \max(\hat{W}N_1 + c \text{ x } \sigma N_1 , \hat{W}N_2 + \text{c x } \sigma N_2)$

3. $$\frac{w}{\max (c \text{ x } \sigma N_1 , c \text{ x } \sigma N_2)} > f$$

Where *c* and *f* are preset constants. All the edges of a tree that satisfy the inconsistency measure are considered inconsistent and are removed from the tree. This result in set of disjoint subtrees each represents a separate cluster. More recently, Paivinen [10] proposed a scale free minimum spanning tree **(SFMST)** clustering algorithm which constructs a scale free networks and outputs clusters containing highly connected vertices and those connected to them.

The **MST** clustering algorithm has been widely used in practice. Xu (Ying), Olman and Xu (Dong) [15] use an **MST** as multidimensional gene expression data. They point out that **MST**- based clustering algorithm does not assume that data points are grouped around centers or separated by regular geometric curve. Thus the shape of the cluster boundary has little impact on the performance of the algorithm. They described three objective functions and the corresponding cluster algorithm for computing *k*-partition of spanning tree for predefined $k > 0$. The algorithm simply removes *k*-1 longest edges so that the weight of the subtrees is minimized.

The second objective function is defined to minimize the total distance between the center and each data point in the cluster. The algorithm removes first *k*-1 edges from the tree, which creates a *k*-partitions. Sanpawat Kantabutra and Chumphol Bunkhumpornpat [13] proposed **MST** based clustering algorithm works on a data structure called *"Clustering Tree"* which naturally generates an approximate inter-cluster distance. The inter-cluster distance tells us the amount of similarity that points between the two clusters differ.

3. OUR CLUSTERING ALGORITHM

A tree is a simple structure for representing binary relationship, and any connected components of tree is called *subtree*. Through this **MST** representation, we can convert a multi-dimensional clustering problem to a tree partitioning problem, ie., finding particular set of tree edges and then cutting them. Representing a set of multi-dimensional data points as simple tree structure will clearly lose some of the inter data relationship. However many clustering algorithm proved that no essential information is lost for the purpose of clustering. This is achieved through rigorous proof that each cluster corresponds to one subtree, which does not overlap the representing subtree of any other cluster. Clustering problem is equivalent to a problem of identifying these subtrees through solving a tree partitioning problem. The inherent cluster structure of a point set in a metric space is closely related to how objects or concepts are embedded in the point set. In practice, the approximate number of embedded objects can sometimes be acquired with the help of domain experts. Other times this information is hidden and unavailable to the clustering algorithm. In this section we propose clustering algorithm which produces k number of clusters with k number of binary tree called Minimum Spanning Clustering Tree.

3.1 EMSCT CLUSTERING ALGORITHM.

Given a point set S in $\boldsymbol{E^n}$ and the desired number of clusters k, the hierarchical method starts by constructing an **MST** from the points in S. The weight of the edge in the tree is Euclidean distance between the two end points. Next the average weight $\hat{W}$ of the edges in the entire **EMST** and its standard deviation σ are computed; any edge with $w > \hat{W} + \sigma$ or *current longest edge* is removed from the tree. This leads to a set of disjoint subtrees $S_T = \{T_1, T_2 \ldots\}$. Each of these subtrees T_i is treated as cluster. Minimum Spanning Clustering Tree is constructed for each subtree T_i. Oleksandr Grygorash et al proposed algorithm [9] which generates k clusters. We modified the algorithm in order to generate k clusters with k number *Minimum Spanning Clustering Tree (***MSCT).** Using the **MSCT** the cluster membership of any object can be easily identified. Our Algorithm will produce more compact clusters.

Figure 1 illustrate a typical example of cases in which simply remove the k-1 longest edges does not necessarily output the desired cluster structure. Figure 2 shows the possible distribution of the points in the cluster for k = 2. Our algorithm will generate two clusters (subtrees). Minimum Spanning Clustering trees are constructed using the above subtrees is shown in figure 3. Minimum Spanning Clustering Tree is a binary tree that assists in finding similarities in objects which are present in the cluster. We give the definition for the **MSCT** tree and recursive function algorithm to build it.

Definition 1. Given an **EMST** T, **MSCT** is a binary tree in which each node contains, among other things, a pair of points a, b in T and the distance between them. Each parent node always contains a smaller distance than that of its children.

Algorithm: EMSCT (k)
Input : S the point set
Output: k number of clusters with binary tree

Let e be an edge in the **EMST** constructed from S
Let W_e be the weight of e
Let σ be the standard deviation of the edge weights
Let n_c be the number of clusters
Let n be the root number for binary tree

1. Construct an **EMST** from S
2. Compute the average weight of $\hat{W}$ of all the edges
3. Compute standard deviation σ of the edges
4. $S_T = ø$; $n_c = 1$; $n = 0$;
5. **Repeat**
6. **For** each $e \in$ **EMST**
7. **If** ($W_e > \hat{W} + \sigma$) or (Current longest edge e)
8. Remove e from **EMST** which result T', a new disjoint subtree
9. $S_T = S_T \cup \{ T'' \}$ // T'' is new disjoint subtree
10. $n_c = n_c + 1$; $n = n + 1$;
11. Tree (T', n) // Construction of Minimum Spanning Clustering Tree
12. **Until** $n_c = k$
13. **Return** k Minimum spanning clustering tree

Input : T' the Subtree
Output: Minimum Spanning Clustering Tree
Tree (T', r_i)
1. **If** (T' has some edge)
2. e = get-min-edge(T')
3. rs = create-blank-node( )
4. fill-infor(rs, e)

5. **Return** (connect (Tree (left-subtree(r_i), r_i.left-node-id), Tree (right-subtree(r_i), r_i.right-node-id),*rs*)
6. **Else**
7. *rs* = create-blank-node( )
8. **Return** *rs*

Oleksandr Grygorash et al proposed algorithm [9] which generates *k* clusters. The Algorithm proposed by Sanpawat Kantabutra and Bunkhumpornapt generates naturally appropriate inter-cluster distances [13]. We modified both the algorithm to generate *k* number of clusters with appropriate intra-cluster distance. The intra-cluster distance is the distance between objects within the cluster. The **EMSCT** algorithm works in two phases. The first phase generate subtree (cluster) and the second phase transform the subtree (cluster) into binary tree which is shown in the figure 3. It first constructs **EMST** form set of point S (line 1). Average weight of edges and standard deviation are computed (lines 2-3). Inconsistent edges are identified and removed from **EMST** to generate subtree *T'* (lines 5-8). This subtree is used as argument for the recursive function *Tree*, which converts the subtree in to binary tree called Minimum Spanning Clustering Tree. The recursive function *Tree* first checks to see if *T'* is still contains some edge (line 1). If so it finds a minimum edge *e* and creates blank node *rs* (lines 2-4). It then recursively solves the left and right minimum spanning subtrees (line 6) until *T'* has no edge and binary is returned (lines 6-9). The time complexity of **EMSCT** algorithm is $O(k|E|^2)$. Our algorithm generates binary trees with intra-cluster distance. This distance tells us that least amount of similarity that points between the two objects in a cluster differ. This piece information can be very useful in several applications that include web mining, image processing, event mining etc.

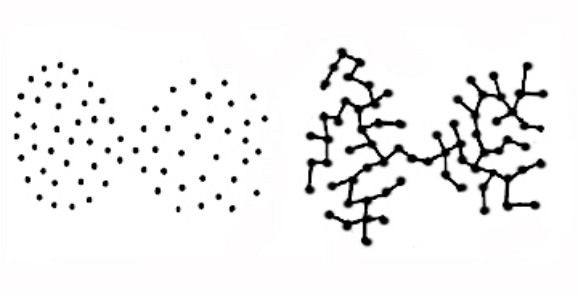

Figure 1. Clusters connected through a point

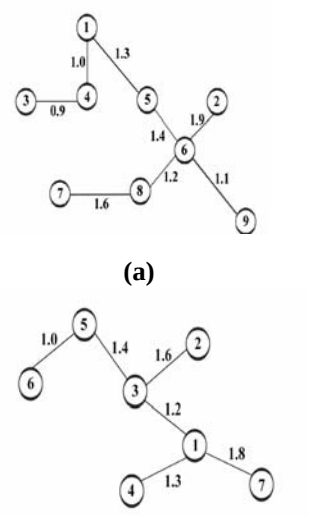

Figure 2. Two Subtrees (Clusters)

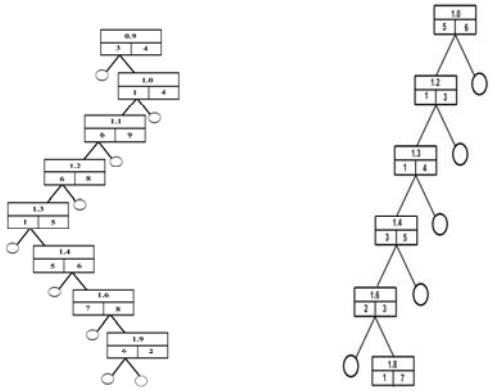

Figure 3. Minimum Spanning Clustering Tree

4. CONCLUSION:

Our **EMSCT** clustering algorithm assumes a given cluster number. The algorithm gradually finds k clusters. Our algorithm does not require the users to select and try various parameters combinations in order to get the desired output. The benefit of the algorithm is to find similarity structures within clusters. The same **MSCT** tree can also be used as search tree. All of these look nice from theoretical point of view. However from practical point of view, there is still some room for improvement for

running time of the clustering algorithm. This could perhaps be accomplished by using some appropriate data structure. With available the efficient tree search algorithm one can use our **MSCT** tree to find out an object to which a given point belongs, that is desired page in a given set of pages that is produced by search engine.

REFERENCES:

[1] T.Asano, B.Bhattacharya, M.Keil and F.Yao. "Clustering Algorithms based on minimum and maximum spanning trees". In *Proceedings of the 4th Annual Symposium on Computational Geometry,*Pages 252-257, 1988.

[2] D.Avis "Diameter partitioning". *Discrete and Computational Geometry,* 1:265-276,1986

[3] M.Fredman and D.Willard. "Trans-dichotomous algorithms for minimum spanning trees and shortest paths". In *Proceedings of the 31st Annual IEEE Symposium on Foundations of Computer Science,*pages 719-725, 1990.

[4] H.Gabow, T.Spencer and R.Rarjan. "Efficient algorithms for finding minimum spanning trees in undirected and directed graphs". *Combinatorica,*6(2):109-122, 1986.

[5] D,Johnson. "The np-completeness column: An ongoing guide". *Journal of Algorithms*,3:182-195, 1982.

[6] D.Karger, P.Klein and R.Tarjan. "A randomized linear-time algorithm to find minimum spanning trees". *Journal of the ACM*,42(2):321-328,1995.

[7] J.Kruskal. "On the shortest spanning subtree and the travelling salesman problem". *In Proceedings of the American Mathematical Society,* Pages 48-50, 1956.

[8] J.Nesetril, E.Milkova and H.Nesetrilova. Otakar boruvka "On minimum spanning tree problem: Translation of both the 1926 papers, comments, history. DMATH"*: Discrete Mathematics,*233, 2001.

[9] Oleksandr Grygorash, Yan Zhou, Zach Jorgensen. "Minimum spanning Tree Based Clustering Algorithms". *Proceedings of the 18th IEEE International conference on tools with Artificial Intelligence(ICTAI'06)* 2006.

[10] N.Paivinen. "Clustering with a minimum spanning of scale-free-like structure". *Pattern Recogn.Lett.,*26(7):921-930, 2005.

[11] F.Preparata and M.Shamos. "Computational Geometry: An Introduction". *Springer-Verlag, Newyork,NY,USA*, 1985

[12] R.Prim. "Shortest connection networks and some generalization". *Bell systems Technical Journal,* 36:1389-1401, 1957.

[13] Sanpawat Kantabutra and Chumphol Bunkhumpornpat. "Two Birds With One Stone: A Similarity Guaranteed Clustering Algorithm and Its Search Tree" *Proceeding of IEEE TENCON Conference, Chiang Mai ,Thailand November* ,2004.

[14] Stefan Wuchty and Peter F. Stadler. "Centers of Complex Networks". 2006

[15] Y.Xu, V.Olman and D.Xu. "Minimum spanning trees for gene expression data clustering". *Genome Informatics,* 12:24-33, 2001.

[16] C.Zahn. "Graph-theoretical methods for detecting and describing gestalt clusters". *IEEE Transactions on Computers*, C-20:68-86, 1971.

BIOGRAPHY OF AUTHORS

S. John Peter is working as Assistant professor in Computer Science, St. Xavier's college (Autonomus), Palayamkottai, Tirunelveli. He earned his M.Sc degree from Bharadhidasan University, Trichirappli. He also earned his M.Phil from Bhradhidasan University, Trichirappli. Now he is doing Ph.D in Computer Science at Manonmaniam Sundranar University, Tirunelveli. He has presented research papers on clustering algorithm in various national seminars.

Dr. S. P. Victor earned his M.C.A. degree from Bharathidasan University, Tiruchirappalli. The M. S. University, Tirunelveli, awarded him Ph.D. degree in Computer Science for his research in Parallel Algorithms. He is the Head of the department of computer science, and the Director of the computer science research centre, St. Xavier's college (Autonomous), Palayamkottai, Tirunelveli. The M.S. University, Tirunelveli and Bharathiar University, Coimbatore have recognized him as a research guide. He has published research papers in international, national journals and conference proceedings. He has organized Conferences and Seminars at national and state level.