OPTIMAL DUAL SIMILARITY NOISE-FREE CLUSTERS USING DYNAMIC MINIMUM SPANNING TREE

S. JOHN PETER
Assistant Professor
Department of Computer Science and Research Center
St. Xavier's College, Palayamkottai
Tamil Nadu, India.

ABSTRACT

Clustering is a process of discovering groups of objects such that the objects of the same group are similar, and objects belonging to different groups are dissimilar. A number of clustering algorithms exist that can solve the problem of clustering, but most of them are very sensitive to their input parameters. Minimum Spanning Tree clustering algorithm is capable of detecting clusters with irregular boundaries. Detecting outlier in database (as unusual objects) is a big desire. In data mining detection of anomalous pattern in data is more interesting than detecting inliers. In this paper we propose a Minimum Spanning Tree based clustering algorithm for noise-free or pure clusters. The algorithm constructs hierarchy from top to bottom. At each hierarchical level, it optimizes the number of cluster, from which the proper hierarchical structure of underlying dataset can be found. The algorithm uses a new cluster validation criterion based on the geometric property of data partition of the data set in order to find the proper number of clusters at each level. The algorithm works in two phases. The first phase of the algorithm produces subtrees(noise-free clusters). The second phase converts the subtrees into dendrogram. The key feature of our algorithm is it finds noise-free/error-free clusters for a given dataset without using any input parameters. The key feature of the algorithm is it uses both divisive and agglomerative approaches to find optimal Dual similarity noise-free clusters.

Key Words: *Center, Clustering, Cluster validity, Cluster Separation Dendrogram, Euclidean minimum spanning tree, Eccentricity, Hierarchical clustering, Outliers, Subtree,*

1. INTRODUCTION

An outlier is an observation of data that deviates from other observations so much that it arouses suspicious that was generated by a different mechanism from the most part of data [14]. Inlier, on the other hand, is defined as observation that is explained by underlying probability density function. In clustering, outliers are considered as noise observations that should be removed in order to make more reasonable clustering [15] Outlier may be erroneous or real in the following sense. Real outliers are observations whose actual values are very different than those observed for rest of the data and violate plausible

relationship among variables. Outliers can often be individual or groups of clients exhibiting behavior outside the range of what is considered normal. Outliers can be removed or considered separately in *regression modeling* to improve accuracy which can be considered as benefit of outliers. Identifying them prior to modeling and analysis is important [47]. In clustering–based methods, outlier is defined as observation that does not fit to the overall clustering pattern [52]

The importance of outlier detection is due to the fact that outliers in the data translate to significant (and often critical) information in a wide variety of application domains. For example, an anomalous traffic pattern in a computer network could mean that a hacked computer is sending out sensitive data to an unauthorized destination. In public health data, outlier detection techniques are widely used to detect anomalous pattern in patient medical records which could be symptoms of new diseases. Similarly, outliers in credit card transaction data could indicate credit card theft or misuse. Outliers can also translate to critical entities such as in military surveillance, whereas the presence of unusual region in a satellite image of enemy are could indicate enemy troop movement. Or anomalous readings from space craft would signify a fault in some of the craft. Outlier detection has been found to be directly applicable in large number of domains.

Many data-mining algorithms find outliers as a side-product of clustering algorithms. However these techniques define outlier as points, which do not lie in clusters. Thus, the techniques implicitly define outliers as the background noise in which the clusters are embedded. Another class of techniques defines outlier as points, which are neither a part of a cluster nor part of background noise; rather they are specifically points which behave very differently from the norm [1].

The outlier detection problem in some cases is similar to the classification problem. Clustering is a popular technique used to group similar data points or objects in groups or clusters [4]. Clustering is an important tool for outlier analysis. Several clustering–based outlier deduction techniques have been developed. Most of these techniques rely on the key assumption that normal objects belong to large and dense clusters, while outliers form very small clusters [32, 34]. The main concern of *clustering–based* outlier detection algorithms is to find clusters and outliers, which are often regarded as noise that should be removed in order to make more reliable clustering [19]. Some noisy points may be far away from the data points, whereas the others may be close. The far away noisy points would affect the result more significantly because they are more different from the data points. It is desirable to identify and remove the outliers, which are far away from all the other points in cluster [26]. So, to improve the clustering such algorithm use the same process and functionality to solve both clustering and outlier discovery [9].

The problem of determining the correct number of clusters in a data set is perhaps the most difficult and ambiguous part of cluster analysis. The "true" number of clusters depends on the "level" on is viewing the data. Another problem is due to the methods that may yield the "correct" number of clusters for a "bad" classification [9]. Furthermore, it has been emphasized that mechanical methods for determining the optimal number of clusters should not ignore that the fact that the overall clustering process has an unsupervised nature and its fundamental objective is to uncover the unknown structure of a data set, not to impose one. For these reasons, one should
be well aware about the explicit and implicit assumptions underlying the actual clustering procedure before the number of clusters can be reliably estimated or, otherwise the initial objective of the process may be lost. As a solution for this, Hardy [20] recommends that the determination of optimal number of clusters should be made by using several different clustering methods that together produce more information about the data. By forcing a structure to a data set, the important and surprising facts about the data will likely remain uncovered.

In some applications the number of clusters is not a problem, because it is predetermined by the context [21]. Then the goal is to obtain a mechanical partition for a particular data using a fixed number of clusters. Such a process is not intended for inspecting new and unexpected facts arising from the data. Hence, splitting up a homogeneous data set in a "fair" way is much more straightforward problem when compared to the analysis of hidden structures from heterogeneous data set. The clustering algorithms [27, 35] partitioning the data set in to k clusters without knowing the homogeneity of groups. Hence the principal goal of these clustering problems is not to uncover novel or interesting facts about data.

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 [39, 31, 33]. The 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 [30, 11, 24]. 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** based clustering algorithm is known to be capable of detecting clusters with various shapes and size [50]. Unlike traditional clustering algorithms, the **MST** clustering algorithm does not assume a spherical shapes structure of the underlying data. The **EMST** clustering algorithm [38, 39] 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 intra-cluster distance or maximum inter-cluster distance [5]. The **EMST** algorithm has been widely used in practice.

Clustering by minimal spanning tree can be viewed as a hierarchical clustering algorithm which follows a divisive approach. Using this method firstly **MST** is constructed for a given input. There are different methods to produce group 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 [5, 48].

Geometric notion of centrality are closely linked to facility location problem. The distance matrix *D* can computed rather efficiently using Dijkstra's algorithm with time complexity $O(|V|^2 \ln |V|)$ [44].

The *eccentricity* of a vertex *x* in *G* and radius ρ (G), respectively are defined as

$e(x) = \max_{y \in V} d(x, y)$ and $\rho(G) = \min_{x \in V} e(x)$

The *center* of *G* is the set

$$\boldsymbol{C}(G) = \{x \in V \mid e(x) = \rho(G)\}$$

C (G) is the center to the "*emergency facility location problem*" which is always contain single block of *G*. The length of the longest path in the graph is called *diameter* of the graph *G*. we can define diameter D (G) as

$$D(G) = \max_{x \in V} e(x)$$

The *diameter* set of *G* is

$$Dia(G) = \{x \in V \mid e(x) = D(G)\}$$

All existing clustering Algorithm require a number of parameters as their inputs and these parameters can significantly affect the cluster quality. Our algorithm does not require a predefined cluster number. 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 naturally approximate intra-cluster distance.

Hierarchical clustering is a sequence of partitions in which each partition is nested into the next in sequence. An Agglomerative algorithm for hierarchical clustering starts with disjoint clustering, which places each of the *n* objects in an individual cluster [4]. The hierarchical clustering algorithm being employed dictates how the proximity matrix or proximity graph should be interpreted to merge two or more of these trivial clusters, thus nesting the trivial clusters into second partition. The process is repeated to form a sequence of nested clustering in which the number of clusters decreases as a sequence progress until single cluster containing all *n* objects, called the *conjoint clustering*, remains[4].

Nearly all hierarchical clustering techniques that include the tree structure have two short comings: (1) they do not properly represent hierarchical relationship and (2) once the data are assigned improperly to a given cluster it cannot later reevaluate and placed in another cluster.

In this paper, we propose a new algorithm *Dynamically Growing Euclidean Minimum Spanning Tree Algorithm for Noise-free Clusters* (**DGEMSTANFC**) which can overcome these shortcomings. The **DGEMSTANFC** algorithm optimizes the number of clusters at each hierarchical level with the cluster validation criteria during the minimum spanning tree construction process. Then the hierarchy constructed by the algorithm can properly represent the hierarchical structure of the underlying dataset, which improves the accuracy of the final clustering result.

Our **DGEMSTANFC** clustering algorithm addresses the issues of undesired clustering structure and unnecessary large number of clusters. Our algorithm does not require a

predefined cluster number. 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 optimal numbers of noise-free clusters (regions) are obtained. Hence the title! In section 2 we review some of the existing works on outliers and graph based clustering algorithms. In Section 3 we propose **DGEMSTANFC** algorithm which produces optimal number of noise-free clusters with Dendrogram. Hence we named this new cluster as *Optimal Dual similarity noise-free clusters*. Finally in conclusion we summarize the strength of our methods and possible improvements.

2. RELATED WORK.

There is no single universally applicable or generic outlier detection approach [32, 34]. Therefore there is many approaches have been proposed to deduct outliers. These approaches are classified into four major categories as *distribution-based, distance-based, density-based and clustering-based* [51]. Here we discuss about density-based and clustering-based outlier detection approaches.

In *Density-based* methods outlier is defined from local density of observation. These methods used different density estimation strategies. A low local density on the observation is an indication of a possible outlier. Brito et al [8] proposed a *Mutual k-Nearest-Neighbor* (**MkNN**) graph based approach. **MkNN** graph is a graph where an edge exits between vertices v_i and v_j if they both belong to each others k-neighborhood. **MkNN** graph is undirected and is special case of *k-Nearest-Neighbor* (**kNN**) graph, in which every node has pointers to its k nearest neighbors. Each connected component is considered as cluster, if it contains more than one vector and an outlier when connected component contains only one vector. Connected component with just one vertex is defined as an outlier. The problem with this approach is that outlier too close to inliers, can be misclassified. *Density-based* approaches [7, 36] compute the density of the regions in the data and declare the objects in low dense regions as outliers.

Clustering-based approaches [32, 12, 22, 26], consider clusters of small sizes as outliers. In these approaches, small clusters (clusters containing significantly less points than other clusters) are considered as outliers. The advantage of *clustering- based* approaches is that they do not have to be supervised.

Jiang et al., [26] proposed a two-phase method to detect outliers. In the first phase, clusters are produced using modified K-means algorithm, and then in the second phase, an Outlier-Finding Process **(OFP)** is proposed. The *small clusters* are selected and regarded as outliers. *Small cluster* is defined as a cluster with fewer points than half the average number of points in the k number of clusters. Loureio [32] proposed a method for detecting outlier. Hierarchical clustering technique is used for detecting outliers. The key idea is to use the size of the resulting clusters as indicators of the presence of outliers. Almedia [2] is also used similar approach for detecting outliers. Using the K-means clustering algorithm Yoon [49] proposed a method to detect outliers.

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. In the mid of 80's, Avis [6] found an $O\ (n^2\ log^2\ n)$ algorithm for the min-max diameter-2 clustering problem. Asano, Bhattacharya, Keil and Yao [5] 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 [29]. Asano, Bhattacharya, Keil and Yao also considered the clustering problem in which the goal to maximize the minimum inter-cluster 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 [5]. 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 [50] proposes to construct **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 one of the three conditions hold:

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

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

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

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. Paivinen [37] proposed a Scale Free Minimum Spanning Tree **(SFMST)** clustering algorithm which constructs 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) [48] use 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.

Hierarchical clustering algorithm proposed by S.C.Johnson [28] uses proximity matrix as input data. The algorithm is an agglomerative scheme that erases rows and columns in the proximity matrix as old clusters are merged into new ones. The algorithm is simplified by assuming no ties in the proximity matrix. Graph based Hierarchical Algorithm was proposed by Hubert [23] using single link and complete link methods. He used threshold graph for formation of hierarchical clustering. An algorithm for single-link hierarchical clustering begins with the minimum spanning tree (MST) for G (∞), which is a proximity graph containing $n(n-1)/2$ edge was proposed by Gower and Ross [25]. Later Hansen and DeLattre [19] proposed another hierarchical algorithm from graph coloring.

Many different methods for determining the number of clusters have been developed. Hierarchical clustering methods provide direct information about the number of clusters by clustering objects on a number of different hierarchical levels, which are then presented by a graphical tree structure known as *dendrogram.* One may apply some external criteria to validate the solutions on different levels or use the dendrogram visualization for determining the best cluster structure.

The selection of the correct number of clusters is actually a kind of validation problem. A large number of clusters provides a more complex "model" where as a small number may approximate data too much. Hence, several methods and indices have been developed for the problem of cluster validation and selection of the number of clusters [42, 18, 41, 43, 45]. Many of them based on the within and between-group distance.

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 preset **DGEMSTANFC** clustering algorithm which produce optimal number of clusters, with dendrogram for each of them.

A. DGEMSTANFC Clustering Algorithm

Given a point set S in $\mathbf{E^n}$, the hierarchical method starts by constructing a Minimum Spanning Tree (**MST)** from the points in *S*. The weight of the edge in the tree is Euclidean distance between the two end points. So we named this **MST** as **EMST1.** Next the average weight $\hat{W}$ of the edges in the entire **EMST1** 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 $S_T = \{T_1, T_2 \ldots\}$ *(divisive approach)*. Each of these subtrees T_i is treated as cluster. Oleksandr Grygorash et al proposed algorithm [35] which generates *k* clusters. Our previous algorithm [27] generates *k* clusters with centers, which used to produce Dual similarity clusters. Both of the minimum spanning tree based algorithms assumed the desired number of clusters in advance. In practice, determining the number of clusters is often coupled with discovering cluster structure. Hence we propose a new algorithm named, *Dynamically Growing Euclidean Minimum Spanning Tree algorithm (***DGEMSTANFC**), which does not require a predefined cluster number. The algorithm works in two phases. The first phase of the algorithm partitioned the **EMST1** into sub trees (clusters/regions). The centers of clusters or regions are identified using eccentricity of points. These points are a representative point for the each subtree S_T. A point c_i is assigned to a cluster *i* if $c_i \in T_i$. The group of center points is represented as $C = \{c_1, c_2 \ldots\ldots c_k\}$. These center points $c_1, c_2 \ldots . c_k$ are connected and again minimum spanning tree **EMST2** is constructed is shown in the Figure 4. This **EMST2** is used for finding optimal number clusters. A Euclidean distance between pair of clusters can be represented by a corresponding weighted edge. Our algorithm is also based on the minimum spanning tree but not limited to two-dimensional points. There were two kinds of clustering problem; one that minimizes the maximum intra-cluster distance and the other maximizes the minimum inter-cluster distances. Our Algorithm produces clusters with intra-cluster similarity. The Second phase of the algorithm converts the subtree/cluster into dendrogram (*agglomerative approach*). This algorithm use both divisive as well as agglomerative approach to find Dual similarity clusters. Since the subtrees are themselves are clusters, are further, classified in order to get more informative similarity clusters. To detect the outliers from the clusters we use the *degree number* of points in the clusters. For any undirected graph *G* the *degree* of a vertex *v*, written as *deg (v),* is equal to the number of edges in G which contains *v,* that is, which are incident on *v*[13].

We propose the following definition for outliers based on **MST,**

Definition 1: Given a **MST** for a data set *S*, outlier is a vertex *v*, whose *degree* is equal to 1, with *dist (v, Nearest-Neighbor (v)) > THR.*

where *THR* is a threshold value used as control parameter. The optimal number of subtrees (clusters) T_i, are created from the **EMST1** using the first phase of the **DGMSTANFC** algorithm. Each T_i is treated as a **MST.** Then vertices *v,* which have *degree* 1 are identified. Then we find *Nearest-Neighbors* for the above vertices *v.* The *distance* between the vertices *v* and its nearest neighbor vertex is computed. If the computed *distance* exceeds the threshold value *THR then* the corresponding vertices are identified as an outlier is shown in the Fig 2.

Here, in this algorithm we use a cluster validation criterion based on the geometric characteristics of the clusters, in which only the inter-cluster metric is used. The **DGMSTANFC** algorithm is a nearest centroid-based clustering algorithm, which creates region or subtrees (clusters/regions) of the data space. The algorithm partitions a set *S* of data of data *D* in data space in to *n* regions (clusters). Each region is represented by a centroid reference vector. If we let *p* be the centroid representing a region (cluster), all data within the region (cluster) are closer to the centroid *p* of the region than to any other centroid *q*:

$$R(p) = \{x \in D \mid dist(x, p) \leq dist(x, q) \ \forall q\} \quad (1)$$

Thus, the problem of finding the proper number of clusters of a dataset can be transformed into problem of finding the proper region (clusters) of

the dataset. Here, we use the **MST** as a criterion to test the inter-cluster property. Based on this observation, we use a cluster validation criterion, called Cluster Separation (CS) in **DGMST** algorithm [10].

Cluster separation (CS) is defined as the ratio between minimum and maximum edge of MST. ie

$$CS = E_{min} / E_{max}, \quad (2)$$

where E_{max} is the maximum length edge of **MST**, which represents two centroids that are at maximum separation, and E_{min} is the minimum length edge in the MST, which represents two centroids that are nearest to each other. Then, the CS represents the relative separation of centroids. The value of CS ranges from 0 to 1. A low value of CS means that the two centroids are too close to each other and the corresponding partition is not valid. A high CS value means the partitions of the data is even and valid. In practice, we predefine a threshold to test the CS. If the CS is greater than the threshold, the partition of the dataset is valid. Then again partitions the data set by creating subtree (cluster/region). This process continues until the CS is smaller than the threshold. At that point, the proper number of clusters will be the number of cluster minus one. The CS criterion finds the proper binary relationship among clusters in the data space. The value setting of the threshold for the CS will be practical and is dependent on the dataset. The higher the value of the threshold the smaller the number of clusters would be. Generally, the value of the threshold will be > 0.8[10]. Figure 3 shows the CS value versus the number of clusters in hierarchical clustering. The CS value < 0.8 when the number of clusters is 5. Thus, the proper number of clusters for the data set is 4. Furthermore, the computational cost of CS is much lighter because the number of subclusters is small. This makes the CS criterion practical for the **DGEMSTANFC** algorithm when it is used for clustering large dataset.

Algorithm: DGEMSTANFC ( )

Input : *S* the point set, *THR*
Output : Optimal number of clusters with dendrograms

Let *e1* be an edge in the **EMST1** constructed from *S*
Let *e2* be an edge in the **EMST2** constructed from C
Let W_e be the weight of *e1*
Let σ be the standard deviation of the edge weights
in EMST1
Let S_T be the set of disjoint subtrees of the **EMST1**
Let n_c be the number of clusters

1. Construct an **EMST1** from *S*
2. Compute the average weight of $\hat{W}$ of all the Edges from **EMST1**
3. Compute standard deviation σ of the edges
4. $S_T = ø$; $n_c = 1$
5. **Repeat**
6. **For** each *e1* ∈ **EMST1**
7. **If** ($W_e > \hat{W} + \sigma$) or (current longest edge *e1)*
8. Remove *e1* from **EMST1** which result *T', a*
is new disjoint subtree
9. $S_T = S_T \cup \{T'\}$ // *T'* is new disjoint subtree
10. $n_c = n_c + 1$
11. $C = U_{Ti} \in S_T \{C_i\}$
12. Construct an **EMST2** *T* from *C*
13. E_{min} = get-min-edge (*T*)
14. E_{max} = get-max-edge (*T*)
15. $CS = E_{min} / E_{max}$
16. **For** p = *1 to* $|T_i|$ *do*
17. **If** deg (v_p) == 1 **and** *dist* (v_p, *Nearest-Neighbor* (v_p)) > *THR* **then** remove v_p *from* T_i
18. Begin with *T'*, disjoint clusters with level
$L_{nc}(0) = 0$ and sequence number $m = 0$
19. **While** (*T'* has some edge)
20. *e2* = get-min-edge(*T'*) // for least dissimilar pair of clusters
21. *(a, b)* = get-vertices (*e2*)
22. Increment the sequence number $m = m+1$, merge the clusters *(a)* and *(b),*
into single cluster to form next clustering *m* and set the level of this cluster to $L_{nc}(m) = e2$;
23. Update *T'* by forming new vertex by combining the vertices *a, b;*
24. **Until** CS < 0.8
25. **Return** optimal noise-free clusters with dendrogram

Figure 1 shows a typical example of **EMST1** constructed from point set S, in which inconsistent edges are removed to create subtree (clusters/regions). Our algorithm finds the center of the each cluster, which will be useful in many applications. Figure 2 shows the possible distribution of the points in the two cluster structures with their center vertex as 5 and 3.

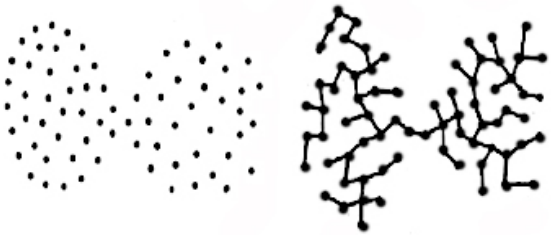
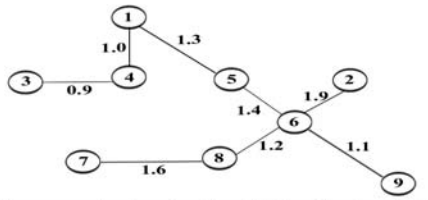


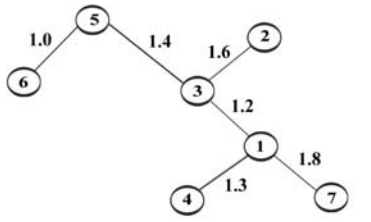
Figure 1. EMST1 - Clusters connected through a point



Vertex v	1	2	3	4	5	6	7	8	9
Eccentricity (v)	5.5	6.7	7.4	6.5	4.2	4.6	7.4	5.8	5.7
Degree (v)	2	1	1	2	2	4	1	2	1

Vertex v	2	3	7	9
Degree (v)	1	1	1	1
Nearest-Neighbour NN(v)	6	4	8	6
Dist(v,NN(v))	1.9	0.9	1.6	1.1

Center vertex = 5
Outlier vertex = 2



Vertex v	1	2	3	4	5	6	7
Eccentricity (v)	3.6	4.6	3.0	4.9	4.4	5.4	5.4
Degree(v)	3	1	3	1	2	1	1

Vertex v	2	4	6	7
Degree (v)	1	1	1	1
Nearest-Neighbour NN(v)	3	1	5	1
Dist(v,NN(v))	1.6	1.3	1.0	1.8

Center vertex = 3
Outlier vertex = 7

Figure. Two Clusters/regions (**MST**) with center points 5 and 3 (outliers 2 and 7)

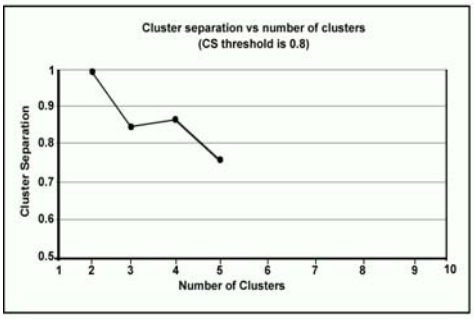


Figure 3. Number of Clusters vs. Cluster Separation

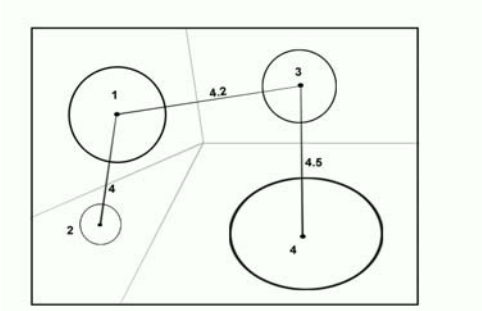
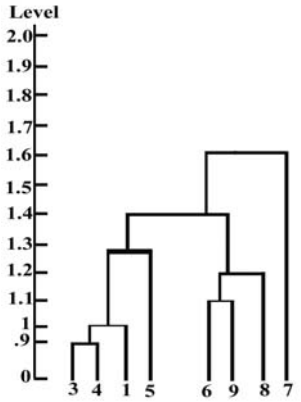


Figure 4. EMST2 From 4 region/cluster center points



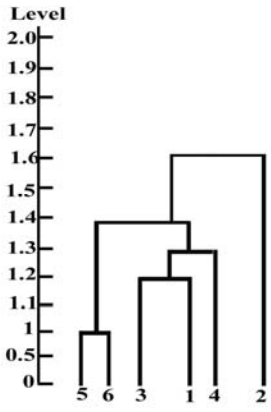


Figure 5. Dendrogram for optimal noise-free clusters

Our **DGEMSTANFC** algorithm works in two phases. The first phase of the algorithm (lines 1-17) uses *divisive approach* of hierarchical clustering. Euclidean minimum spanning tree **EMST1** is constructed in line 1. The average and standard deviation of the weighted edges of the Euclidean minimum spanning tree are computed to find inconsistent edges are specified in the lines 2-3. The inconsistent edges are identified and removed from Euclidean minimum spanning tree **EMST1** in order to generate subtree *T'* is specified in the lines 7-9. The center of each subtree is computed. Lines 13-15 in the algorithm are used find the value of cluster separation (CS). This value is useful to find optimal number of clusters. Outliers are identified and removed from clusters lines (16-17).

The second phase of the algorithm converts the subtrees *T'* into dendrograms is shown in the figure 5 (only two dendrograms are shown). For the newly created subtree *T'* again further hierarchical clustering is performed (lines 18-23). It places the entire disjoint cluster at level 0 (line 18). It then checks to see if *T'* still contains some edge (line 19). If so, it finds minimum edge *e2* (line 20). It then finds the vertices *a, b* of an edge *e2* (line 21). It then merges the vertices and forms a new vertex (*agglomerative approach*). At the same time the sequence number is increased by one and the level of the new cluster is set to the edge weight (line 22). Finally, updation of Euclidean minimum spanning tree is performed at line 23. The lines 19-23 in the algorithm are repeated until *T'* has no edge to merge. Our algorithm uses both divisive as well as agglomerative approach in the **DGEMSTANFC** algorithm to find optimal Dual similarity noise-free clusters.

4. CONCLUSION

Our **DGEMSTANFC** clustering algorithm does not assumes any predefined cluster number. The algorithm gradually finds clusters with center for each cluster. These clusters ensure guaranteed intra-cluster similarity. Our algorithm does not require the users to select and try various parameters combinations in order to get the desired output. Our **DGEMSTANFC** clustering algorithm uses a new cluster validation criterion based on the geometric property of partitioned regions/clusters to produce optimal number of "true" clusters with center for each of them. The inter-cluster distances between centers of clusters/regions are used to find optimal number of clusters. This could perhaps be accomplished by using some appropriate data structure. The **DGEMSTANFC** clustering algorithm generates dendrogram for optimal noise-free clusters, which is used to find the relationship between objects with in a cluster. 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. In the future we will explore and test our proposed clustering algorithm in various domains. The **DGEMSTANFC** algorithm uses both divisive as well agglomerative approaches. In this paper we used both the approaches to find optimal Dual similarity clusters. We will further study the rich properties of **EMST**-based clustering methods in solving different clustering problems.

REFERENCES:

[1] C. Aggarwal and P. Yu, "Outlier Detection for
High Dimensional Data". *In Proceedings of the ACM SIGMOD International Conference on Management of Data, Volume* 30, Issue 2,pages 37 – 46, May 2001.

[2] J.Almeida, L.Barbosa, A.Pais and S.Formosinho, "Improving Hierarchical Cluster Analysis: A new method with OutlierDetection and Automatic Clustering," *Chemometrics and Intelligent Laboratory Systems* 87:208-217, 2007.

[3] F.Angiulli, and C.Pizzuti, "Outlier Mining in Large High-Dimensional Data sets",*IEEE Transactions on Kaoeledge and Data Engineering,* 17(2): 203-215, 2005

[4] Anil K. Jain, Richard C. Dubes *"*Algorithm for Clustering Data"*, Michigan State University, Prentice Hall, Englewood Cliffs, New Jersey* 07632.1988.

[5] 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.

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

[7] M. Breunig, H.Kriegel, R.Ng and J.Sander, Lof: "Identifying density-based local outliers". *In Proceedings of 2000 ACM SIGMOD International Conference on Management of Data. ACM Press,* pp 93-104, 2000.

[8] M. R. Brito, E. L. Chavez, A. J. Quiroz, and J. E. Yukich. "Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection". *Statistics & Probability Letters,* 35(1):33-42, 1997.

[9] B.Custem and I.Gath,,"Detection of Outliers and Robust Estimation using Fuzzy clustering", *Computational Statistics & data Analysis* 15,pp.47-61, 1993.

[10] Feng Luo,Latifur Kahn, Farokh Bastani, I-Ling Yen, and Jizhong Zhou, "A dynamically growing self-organizing tree(DGOST) for hierarchical gene expression profile",Bioinformatics,Vol 20,no 16, pp 2605-2617, 2004.

[11] 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.

[12] Gath and A.Geva, "Fuzzy Clustering for the estimation of the Parameters of the components of Mixtures of Normal distribution", *Pattern Recognition letters*, 9, pp.77-86, 1989.

[13] Gary Chartrand and Ping Zhang "Introduction to Graph Theory", *Tata MgrawwHill, Paperback*-2008.

[14] B. Ghosh-Dastidar and J.L. Schafer, "Outlier Detection and Editing Procedures for Continuous Multivariate Data". *ORP Working Papers,* September 2003.

[15] S. Guha, R. Rastogi, and K. Shim. "CURE an efficient clustering algorithm for large databases". *In Proceeding of the 1998 ACM SIGMOD Int. Conf. on Management of Data , pp 73-84, Seattle, Washington,* 1998.

[16] M. Halkidi, Y.Batistakis and M. Vazirgiannis "On clustering validation techniques", *J.Intel. Inform. System.,* 17, 107-145, 2001

[17] M. Halkidi, Y.Batistakis and M. Vazirgiannis, "Clustering validity checking methods:part II" *SIGMOD record.,* 31, 19-27, 2002

[18] G. Hamerly and C. Elkan, "Learning the k in k-means, in Advances in Neural Information Processing Systems" 16, S. Thrun, L. Saul, and B. Schölkopf, eds., *MIT Press, Cambridge, MA,* 2004.

[19] P. Hansen and M. Delattre, "Complete-link cluster analysis by graph coloring" *Journal of the American Statistical Association* 73, 397-403, 1978.

[20] A. Hardy, "On the number of clusters", *Computational Statistics and Data Analysis,* 23, pp. 83–96, 1996.

[21] T. Hastie, R. Tibshirani, and J. Friedman, "The elements of statistical learning: Data mining, inference and prediction", *Springer-Verlag, 2001.*

[22] Z. He, X. Xu and S. Deng, "Discovering cluster-based Local Outliers", *Pattern Recognition Letters,* Volume 24, Issue 9-10, pp 1641 – 1650, June 2003.

[23] Hubert L. J "Min and max hierarchical clustering using asymmetric similarity measures" *Psychometrika* 38, 63-72, 1973.

[24] 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.

[25] J.C. Gower and G.J.S. Ross "Minimum Spanning trees and single-linkage cluster analysis" *Applied Statistics* 18, 54-64, 1969.

[26] M. Jaing, S. Tseng and C. Su, "Two-phase Clustering Process for Outlier Detection", *Pattern Recognition Letters,* Volume 22, Issue 6 – 7, pp 691 – 700, May 2001.

[27] S. John Peter, S.P. Victor, "A Novel Algorithm for Dual similarity clusters using Minimum spanning tree". *Journal of Theoretical and Applied Information technology,* Vol.14. No.1 pp 60-66, 2010.

[28] S. C. Johnson, "Hierarchical clustering schemes" *Psychometrika* 32, 241-254, 1967.

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

[30] 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.

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

[32] A.Loureiro, L.Torgo and C.Soares, "Outlier detection using Clustering methods: A data cleaning Application", *in Proceedings of KDNet Symposium on Knowledge-based systems for the Public Sector.* Bonn, Germany, 2004.

[33] 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.

[34] K.Niu, C.Huang, S.Zhang and J.Chen, "ODDC: Outlier Detection Using Distance Distribution Clustering", *T.Washio et al. (Eds.): PAKDD 2007 Workshops, Lecture Notes in Artificial Intelligence (LNAI)* 4819,pp.332-343,*Springer-Verlag,* 2007.

[35] 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.

[36] S.Papadimitriou, H.Kitawaga, P.Gibbons and C. Faloutsos, "LOCI: Fast outlier detection using the local correlation integral" *Proc. Of the International Conference on Data Engineering* ,pp.315-326, 2003.

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

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

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

[40] R. Rezaee, B.P.F. Lelie and J.H.C. Reiber, "A new cluster validity index for the fuzzy C-mean", *Pattern Recog. Lett., 19*,237-246, 1998.

[41] D. M. Rocke and J. J. Dai, "Sampling and subsampling for cluster analysis in data mining: With applications to sky survey data", *Data Mining and Knowledge Discovery,* 7, pp. 215–232, 2003.

[42] S. Salvador and P. Chan, "Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms", *in Proceedings Sixteenth IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2004, Los Alamitos, CA, USA*, *IEEE Computer Society,* pp. 576–584 , 2004.

[43] S. Still and W. Bialek, "How many clusters?*" , An information-theoretic perspective*, *Neural Computation*, 16, pp. 2483–2506, 2004.

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

[45] C. Sugar and G. James, "Finding the number of clusters in a data set *", An information theoretic approach*, Journal of the American Statistical Association, 98 pp. 750–763, 2003.

[46] R. Tibshirani, G. Walther and T.Hastie "Estimating the number of clusters in a dataset via the gap statistic". *J.R. Stat. Soc.Ser.B,63*.411-423, 2001.

[47] G. Williams, R. Baxter, H. He, S. Hawkins and L. Gu, "A Comparative Study for RNN

for Outlier Detection in Data Mining", *In Proceedings of the 2nd IEEE International Conference on Data Mining*, page 709, Maebashi City, Japan, December 2002.

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

[49] K.Yoon, O.Kwon and D.Bae, "An approach to outlier Detection of Software Measurement Data using the K-means Clustering Method", *First International Symposium on Empirical Software Engineering and Measurement(ESEM 2007), Madrid.,*pp:443-445, 2007.

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

[51] J. Zhang and N. Wang, "Detecting outlying subspaces for high-dimensional data: the new task, Algorithms and Performance", *Knowledge and Information Systems,* 10(3):333-555, 2006.

[52] T. Zhang, R. Ramakrishnan, and M. Livny. "BIRCH: A new data clustering algorithm and its applications". *Data Mining and Knowledge Discovery*, 1(2):141-182, 1997.

AUTHOR PROFILES:

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, Trichirappalli. He also earned his M.Phil from Bharadhidasan University, Trichirappalli. Now he is doing Ph.D in Computer Science at Manonmaniam Sundranar University, Tirunelveli. He has published research papers on clustering algorithm in various national and international Journals.]

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 has 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.