

Minimal Spanning Tree based Fuzzy Clustering

Ágnes Vathy-Fogarassy, Balázs Feil, and János Abonyi

Abstract—Most of fuzzy clustering algorithms have some discrepancies, e.g. they are not able to detect clusters with convex shapes, the number of the clusters should be *a priori* known, they suffer from numerical problems, like sensitiveness to the initialization, etc. This paper studies the synergistic combination of the hierarchical and graph theoretic minimal spanning tree based clustering algorithm with the partitional Gath-Geva fuzzy clustering algorithm. The aim of this hybridization is to increase the robustness and consistency of the clustering results and to decrease the number of the heuristically defined parameters of these algorithms to decrease the influence of the user on the clustering results. For the analysis of the resulted fuzzy clusters a new fuzzy similarity measure based tool has been presented. The calculated similarities of the clusters can be used for the hierarchical clustering of the resulted fuzzy clusters, which information is useful for cluster merging and for the visualization of the clustering results. As the examples used for the illustration of the operation of the new algorithm will show, the proposed algorithm can detect clusters from data with arbitrary shape and does not suffer from the numerical problems of the classical Gath-Geva fuzzy clustering algorithm.

Keywords—Clustering, fuzzy clustering, minimal spanning tree, cluster validity, fuzzy similarity.

I. INTRODUCTION

FAST and robust clustering algorithms play an important role in extracting useful information from large databases. The aim of cluster analysis is to partition a set of N objects in c clusters such that objects within clusters should be similar to each other and objects in different clusters should be dissimilar from each other. Clustering can be used to quantize the available data, to extract a set of cluster prototypes for the compact representation of the dataset, to select the relevant features, to segment the dataset into homogenous subsets, and to initialize regression and classification models.

There are two main approaches in the clustering: *Hard clustering* algorithms allocate each object to a single cluster during their operation and in its output. *Fuzzy clustering* methods assign degrees of membership in several clusters to each input pattern. So, the fuzzy clustering methods result more dynamic separation of the patterns.

Manuscript received October 18, 2005.

Ágnes Vathy-Fogarassy, University of Veszprém, Department of Mathematics and Computing Science, P.O. Box 158, Veszprém, H-8201 Hungary (e-mail: vathya@almos.vein.hu).

Balázs Feil, University of Veszprém, Department of Process Engineering, P.O. Box 158, Veszprém, H-8201 Hungary (e-mail: feilb@fmt.vein.hu).

János Abonyi, University of Veszprém, Department of Process Engineering, P.O. Box 158, Veszprém, H-8201 Hungary (e-mail: abonyij@fmt.vein.hu).

In the literature a wide variety of algorithms (partitional, hierarchical, density-based, graph-based, model-based, etc.) have been proposed, but it is a difficult challenge to find a general and powerful method that is quite robust and that does not require the fine-tuning of the user. Most of these algorithms have some discrepancies.

For example the basic *partitional methods* are not able to detect convex clusters; when using hierarchical methods the number of the clusters should be *a priori* known, and they are not efficient enough for large datasets; while linkage-based methods often suffer from the chaining effect. A problem accompanying the use of a partitional algorithm is that the number of the desired clusters should be given in advance. The partitional techniques usually produce clusters by optimizing a criterion function defined either locally (on a subset of the patterns) or globally (defined over all of the patterns). Generally, different cluster shapes (orientations, volumes) are required for the different clusters (partitions), but there is no guideline as to how to choose them *a priori*. The norm-inducing matrix of the cluster prototypes can be adapted by using estimates of the data covariance, and can be used to estimate the statistical dependence of the data in each cluster. The Gaussian mixture based fuzzy maximum likelihood estimation algorithm (Gath-Geva algorithm (GG)) is based on such an adaptive distance measure, it can adapt the distance norm to the underlying distribution of the data which is reflected in the different sizes of the clusters, hence it is able to detect clusters with different orientation and volume. Unfortunately the GG algorithm is very sensitive to initialization, hence often it cannot be directly applied to the data.

The *hierarchical clustering* approaches are related to graph-theoretic clustering. These algorithms are able to detect clusters of various shapes and sizes, and they do not require initialization. One of the best-known graph-based divisive clustering algorithm is based on the construction of the *minimal spanning tree (MST)* of the objects [3,7,9,13,16]. By the elimination of any edge from the MST we get subtrees which correspond to clusters. Clustering methods using a minimal spanning tree take advantages of the MST. The MST ignores many possible connections between the data patterns, so the cost of clustering can be decreased. Single-link clusters are subgraphs of the minimum spanning tree of the data [10,11] which are also the connected components. Complete-link clusters are maximal complete subgraphs, and are related to the node colorability of graphs [2]. The maximal complete subgraph was considered the strictest definition of a cluster in [1,15]. Clustering, as an unsupervised learning, is mainly

carried out on the basis of the data structure itself, so the influence of the user should be minimal on the results of the clustering. However, the MST based clustering algorithm has many user-defined parameters that significantly influence the clustering results.

In this paper we propose a hybrid MST and GG clustering algorithm to handle the above mentioned discrepancies. Based on the fuzziness of the resulted clusters (fuzzy membership values) the goodness and the similarities of the received clusters are also evaluated. This information can be effectively used for the analysis and visualization of the clustering results, hence the proposed tool is really useful for data mining. In Section II we discuss the possibilities of the use of minimal spanning trees in the clustering procedure. In this section we cover some well-known terminating criteria of the MST algorithm too. In Section III the new algorithm will be described. We introduce the major steps of the process, and suggest some evaluation criteria for the possible results. Section IV contains application examples based on illustrative datasets to illustrate the usefulness of the proposed method. Section V. concludes the paper.

II. CLUSTERING BASED ON MINIMAL SPANNING TREE

The use of the minimal spanning tree in the clustering methods was initially proposed by Zahn [17]. Fig. 1 depicts a minimal spanning tree, on which points are distributed into three clusters. The objects belonging to different clusters are marked with different dot notations.

A minimal spanning tree is a weighted connected graph, where the sum of the weights is minimal. In a $G=(V,E)$ graph an element of E , called edge, is $e_{ij}=(v_i,v_j)$, where $v_i,v_j \in V$ (vertices). There is a w weight function is defined, which function determines a $w_{i,j}$ weight for each e_{ij} edge. Creating the minimal spanning tree means, that we are searching the $G'=(V,E')$ connected subgraph of G , where $E' \subset E$ and the cost is minimum. The cost is computed in the following way:

$$\sum_{e \in E'} w(e) \quad (1)$$

where $w(e)$ denotes the weight of the $e \in E$ edge. In a G graph, where the number of the vertices is N , MST has exactly $N-1$ edges. The major advantage of the clustering with using MST is, that while the complete graph including N vertices has exactly $\binom{N}{2}$ edges, in the MST we can find only $N-1$ edges.

So the answering the possible most exciting question, namely which edge is the best choice for the elimination, becomes less expensive. A minimal spanning tree can be efficiently computed in $O(N^2)$ time using either Prim's [14] or Kruskal's [12] algorithm.

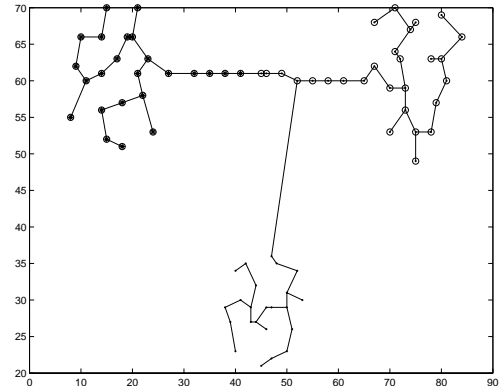


Fig. 1 Example of a minimal spanning tree

A minimal spanning tree can be used in clustering in the following way: let $X=\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ be a set of the data with N distinct objects which we want to distribute in different clusters. \mathbf{x}_i denotes the i -th object, which consists n measured variables, grouped into an n -dimensional column vector $\mathbf{x}_i=[x_{1,i}, x_{2,i}, \dots, x_{n,i}]^T$, $\mathbf{x}_i \in R^n$. Let $d_{i,j}=d(\mathbf{x}_i, \mathbf{x}_j)$ be the distance defined between any \mathbf{x}_i and \mathbf{x}_j . This distance can be computed in different ways (e.g. Euclidean distance, Manhattan distance, Mahalanobis distance, mutual neighbour distance, etc.). Removing edges from the MST leads to a collection of connected subgraphs of G , which can be considered as clusters. Using MST for clustering we are interested in finding the *inconsistent edges*, which lead to the best clustering result.

Clustering by minimal spanning tree can be viewed as a hierarchical clustering algorithm which follows the divisive approach. Using this method firstly we construct a linked structure of the objects, and then the clusters are recursively divided into subclusters. Elimination of k edges from a minimal spanning tree results in $k+1$ disconnected subtrees. Denote δ the length of the deleted edge, and let V_1, V_2 be the sets of the points in the resulting two clusters. In the set of clusters we can state that there are no pairs of points $(\mathbf{x}_1, \mathbf{x}_2)$, $\mathbf{x}_1 \in V_1, \mathbf{x}_2 \in V_2$, such that $d(\mathbf{x}_1, \mathbf{x}_2) < \delta$.

The identification of the inconsistent edges causes problems in the MST clustering algorithms. There exist numerous ways to divide clusters successively, but there is not a suitable choice for all cases. In special cases the elimination is carried out in one step. In these cases a global parameter is used, which determines the edges to be removed from the MST. When this elimination is repeated, we must determine a terminating criterion, when the running of the algorithm is finished, and the current trees can be seen as clusters. Determination of the terminating criterion is also a difficult challenge. The methods which use recursive cutting define some possible terminating criteria. In the next paragraphs we will overview some well-known cutting conditions and terminating criteria, then we introduce our suggestions for using the minimal spanning tree for clustering with new cutting criteria.

Criterion-1: The simplest way to delete edges from the minimal spanning tree is based on the distance between the vertices. By deleting the longest edge in each iteration step we get a nested sequence of subgraphs. As other hierarchical methods, this approach also requires a terminating condition. Several ways are known to stop the algorithms, for example the user can define the number of clusters, or we can give a threshold value on the length also.

Zahn [17] suggested a global threshold value, δ which considers the distribution of the data in the feature space. In [17] this δ threshold is based on the average weight (distances) of the MST :

$$\delta = \lambda \frac{1}{N-1} \sum_{e \in E'} w(e) \quad (2)$$

where λ is a user defined parameter. Of course, λ can be defined in several manner.

Criterion-2: Long edges of the MST do not indicate cluster separation always. When the hidden clusters show different densities, the recursive cutting of the longest edges does not results the expected cluster scheme. Solving this problem Zahn [17] proposed also another idea to detect the hidden separations in the data. Zahn's suggestion is based on the distance of the separated subtrees. He suggested, that an edge is inconsistent if its length is at least f times as long as the average of the length of nearby edges. The input parameter f must be adjusted by the user. To determine which edges are "nearby" is another question. It can be determined by the user, or we can say, that point x_i is nearby point of x_j if point x_i is connected to the point x_j by a path in a minimal spanning tree containing k or fewer edges. This method has the advantage of determining clusters which have different distances separating one another. Another use of the MST based on this criterion is to find dense clusters embedded in a sparse set of points. All that has to be done is to remove all edges longer than some predetermined length in order to extract clusters which are closer than the specified length to each other. If the length is chosen accordingly, the dense clusters are extracted from a sparse set of points easily. The drawback of this method is that the influence of the user is significant at the selection of the f and k parameters.

Criterion-3: The first two criteria are based on the merging or splitting of the objects or clusters using a distance defined between them. Occurrence of a data chain between two clusters can cause that these methods can not separate these clusters. In many approaches the separation is specified with the *goodness of the obtained partitions*. Cluster validity refers to the problem whether a given partition fits to the data all. The clustering algorithm always tries to find the best fit for a fixed number of clusters and the parameterized cluster shapes. However this does not mean that even the best fit is meaningful at all. Either the number of clusters might be wrong or the cluster shapes might not correspond to the groups in the data, if the data can be grouped in a meaningful way at all. Two main approaches to determining the appropriate number of clusters in data can be distinguished:

- The *compatible cluster merging* approaches start with a sufficiently large number of clusters, and successively reduce this number by merging clusters that are similar (compatible) with respect to some predefined criteria.
- Many approaches use *validity measures* to assess the goodness of the obtained partitions. This can be done in two ways:
 - The first approach defines a validity function which evaluates a complete partition. An upper bound for the number of clusters must be estimated (c_{max}), and the algorithms have to be run with each $c \in \{2, 3, \dots, c_{max}\}$. For each partition, the validity function provides a value such that the results of the analysis can be compared indirectly.
 - The second approach consists of the definition of a validity function that evaluates individual clusters of a cluster partition. Again, c_{max} has to be estimated and the cluster analysis has to be carried out for c_{max} . The resulting clusters are compared to each other on the basis of the validity function. Similar clusters are collected in one cluster, very bad clusters are eliminated, so the number of clusters is reduced. The procedure can be repeated until there are *bad* clusters.

In the literature for the hard clustering different scalar validity measures have been proposed, but none of them is perfect on its own. For example *partition index* [5] is the ratio of the sum of compactness and separation of the clusters. *Separation index* [5] uses a minimum distance separation for partition validity. *Dunn's index* [6] is originally proposed to be used at the identification of compact and well separated clusters. They are known as *Dunn-like indices* since they are based on Dunn index. One of the three indices uses for the definition the concepts MST. The number of clusters at which this index takes its maximum value indicates the number of clusters in the underlying data.

Varma and Simon [16] used the Fukuyama-Sugeno clustering measure for deleting different edges from the MST. Denote S the sample index set, and let S_1, S_2 be partitions of S . N_k denotes the number of the objects for each S_k . The Fukuyama-Sugeno clustering measure is defined in the following way:

$$FS(S) = \sum_{k=1}^2 \sum_{j=1}^{N_k} \left[\left\| \mathbf{x}_j^k - \mathbf{v}_k \right\|^2 - \left\| \boldsymbol{\mu}_k - \mathbf{v} \right\|^2 \right] \quad (3)$$

where \mathbf{v} denotes the global mean of all objects, \mathbf{v}_k denotes the mean of the objects in S_k . The symbol \mathbf{x}_j^k refers to the j -th object in the cluster S_k . If the value of $FS(S)$ is small, it indicates tight clusters with large separations between them. Varma and Simon found, that the Fukuyama-Sugeno measure gives the best performance in a dataset with a large number of noisy features.

III. HYBRID MST-GG CLUSTERING ALGORITHM

A. Fuzzy Validity Measures

The previous cluster validity measures are good for the evaluation of crisp clustering results. There are several validity indices based on fuzzy approach. In the literature two categories of fuzzy validity indices are discussed. The first category uses only the membership values μ_{ij} , of a fuzzy partition of data. The second involves both the fuzzy partition matrix and the dataset itself.

For the evaluation of the result of fuzzy clustering Bezdek proposed the use of *partition coefficient* [4]. This measure is defined in the following way:

$$PC = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^c \mu_{ij}^2, \quad (4)$$

where N denotes the number of data points, and c is the number of clusters. The value of the PC index is in the range of $[1/c, 1]$. PC values closer to $1/c$ indicate no clustering tendency in the considered dataset or the clustering algorithm failed to reveal it. The drawbacks of this index are: i) monotonous dependency on the number of clusters; ii) it is sensitive to the fuzzifier parameter (m). If $m \rightarrow 1$ the index give the same values for all values of c . On the other hand when $m \rightarrow \infty$ PC exhibits significant knee at $c=2$; iii) the lack of direct connection to the geometry of the data, since they do not use the data themselves.

There is an extension of *Fukuyama-Sugeno index*, which involves the membership values and also the dataset. In this form the Fukuyama-Sugeno index is defined as

$$FS_m = \sum_{i=1}^N \sum_{j=1}^c \mu_{ij}^m \left[\left\| \mathbf{x}_i - \mathbf{v}_j \right\|^2 - \left\| \mathbf{v}_j - \mathbf{v} \right\|^2 \right], \quad (5)$$

where \mathbf{v} is the mean vector of the objects, and \mathbf{v}_j is the mean vector of the j -th cluster. It is clear that for compact and well-separated clusters we expect small values for FS_m . The first term in brackets measures the compactness of the clusters while the second one measures the distances of the cluster representatives.

Other fuzzy validity indices are proposed by Gath and Geva [8], which are based on the concepts of hyper volume and density. Let Σ_j the fuzzy covariance matrix of the j -th cluster defined as

$$\Sigma_j = \frac{\sum_{i=1}^N (\mu_{ij})^m (\mathbf{x}_i - \mathbf{v}_j)(\mathbf{x}_i - \mathbf{v}_j)^T}{\sum_{i=1}^N (\mu_{ik})^m}, \quad (6)$$

where \mathbf{v}_j denotes the mean vector of C_j .

The *fuzzy hyper volume* of j -th cluster is given by equation

$$V_j = \det(\Sigma_j)^{1/2} \quad (7)$$

Then the *total fuzzy hyper volume* is defined as

$$FH = \sum_{j=1}^c V_j \quad (8)$$

The resulting clusters can be compared to each other on the basis of their volume. Very bad clusters with large volumes can be further partitioned, so the number of clusters can be recursively increased. The procedure can be repeated until there are *bad* clusters.

Since in this paper the MST will be used for the initialization of GG clustering, the splitting criterion based on the fuzzy hyper volume will be used as the third criterion of the MST based clustering algorithm. In the following the whole algorithm will be described.

B. The Hybrid MST-GG Clustering Algorithm

STEP 0 Perform the classical MST-based clustering based on *Criterion-1* and *Criterion-2*. This step detects the well separated clusters and separates clusters with significantly different densities.

STEP 1 Binary Splitting. At the sub-cluster with the largest volume V_i in the so-far formed hierarchical tree, each of the edges in the corresponding sub-MST is cut. With each cut a binary split of the objects is formed. If the current sub-MST includes N_i objects then N_i-1 such splits are formed. The two sub-clusters, formed by the binary split, plus the clusters formed so far (excluding the current node) compose a potential partition.

STEP 2 Best split. The hyper volumes (FH) of all formed N_i-1 potential partitions are computed. The one that exhibits the lowest FH is selected as the best partition of the objects in the current sub-MST. (*Criterion-3*).

STEP 3 Iteration and Termination criterion. Following a depth-first tree-growing process, steps 1 and 2 are iteratively performed. The final outcome is a hierarchical clustering tree where the termination nodes are the final clusters. Special parameters control the generalization level of the hierarchical clustering tree (e.g., minimum number of objects in each sub-cluster).

STEP 4 When the compact parametric representation of the result of the clustering is needed, the GG clustering is performed, where the number of the Gaussians is equal to the termination nodes, and the iterative algorithm is initialized based on the partition obtained at the previous step.

C. Analysis of the Clustering Results

The previously introduced hybrid MST-GG clustering algorithm results a dendrogram, including different possible clustering outcomes. Choosing the best consequence from these embedded results, we must define the term of the *best clustering result*. In our approach we construct this concept on the following tree criteria: i) goodness of a particular cluster; ii) goodness of the separation; iii) goodness of the whole clustering result. The second criterion refers to the cluster separation. In the Step 3 of the hybrid MST-GG a binary splitting is executed. This means an elimination of an edge from the minimal spanning tree. Let e_{ij} be the selected edge

with the end points x_i and x_j , and the produced clusters C_a and C_b . We can say that this edge was selected adequate, if in the recomputed fuzzy partition matrix the object x_i belongs more to the partition say C_a and less to the partition C_b . Similarly the same must hold for x_j , the other way around. The cluster separation in the l -th iteration step with the use of the values of the fuzzy partition matrix formally can be expressed in the following way:

$$CP(e_{ij}) = \sum_{k=a,b} \left| \mu_{ki}^{(l)} - \mu_{kj}^{(l)} \right| \quad (9)$$

Values of $CP(e_{ij}) \rightarrow 2$ denotes an excellent cluster separations.

The last evaluation criterion takes the goodness of the result into consideration. The result of clustering shows suitable partitions, if objects belong with high probability to own cluster. The expression (5) can be used for the evaluation of the considered result.

In this work, the visualization of fuzzy clustering is also in focus. In the first step after clustering, there is a need to determine how similar the resulted clusters are. For that purpose, a fuzzy set similarity measure can be used because fuzzy clusters can be seen as fuzzy sets. The similarity of two sets, A and B can be expressed as follows:

$$S_{A,B} = \frac{A \cap B}{A \cup B} \quad (10)$$

In case of the analysis of fuzzy clusters as multivariate fuzzy sets, the *min* and *max* operators can be used as set operators intersection and union.

$$S_{i,j} = \frac{\sum_{k=1}^N \min(\mu_{k,i}, \mu_{k,j})}{\sum_{k=1}^N \max(\mu_{k,i}, \mu_{k,j})} \quad (11)$$

In this way, all clusters can be compared to each other. Based on the obtained symmetric similarity matrix, *dendrogram* can be drawn to visualize and hierarchically cluster the fuzzy clusters (an example can be seen in Fig. 4). Using this diagram, the human “data miner” can get a conception how similar the clusters are in the original space and are able to determine which clusters should be merged if it is needed.

IV. APPLICATION EXAMPLES

In this section we present the results obtained on the clustering of some illustrative datasets.

A. Handling the Chaining Effect

The first example is intended to illustrate that the proposed cluster volume based splitting extension of the basic MST based clustering algorithm is able to handle (avoid) the chaining phenomena. For this toy example the classical MST based algorithm detects only two clusters. With the use of the volume-based partitioning criterion, the first cluster has been splitted (Fig. 1).

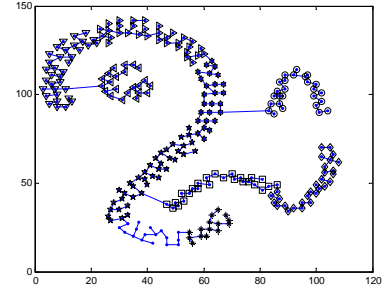


Fig. 2 Minimal spanning tree resulted by the proposed algorithm

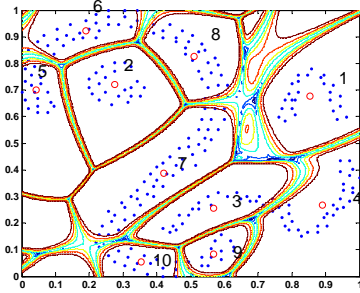


Fig. 3 Result of MST-GG algorithm

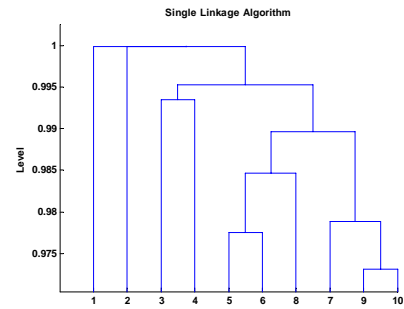


Fig. 4 Similarity of clusters resulted by MST-GG algorithm

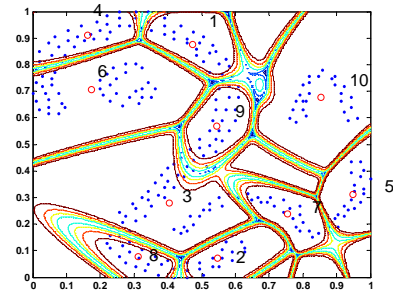


Fig. 5 Result of GG algorithm initialized from FCM

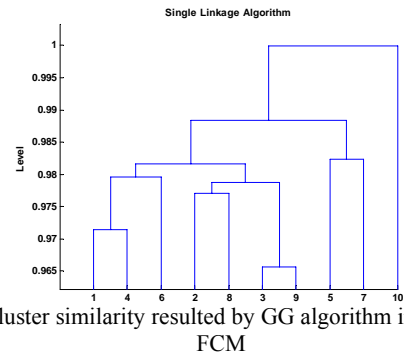


Fig. 6 Cluster similarity resulted by GG algorithm initialized by FCM

B. Handling the Convex Shapes of Clusters – Effect of the Initialization

The previous short example illustrated the main benefit of the incorporation of the cluster validity based criterion into the classical MST based clustering algorithm. In the following it will be shown how the resulted nonparametric clusters can be approximated by mixture of Gaussians, and how this approach is beneficial for the initialization of these iterative partitional algorithms.

Let us consider a more complex clustering problem with convex shape of clusters. As Fig. 2 shows, the proposed MST based clustering algorithm is able to detect properly cluster of these data. The partitioning of the clusters has not been stopped at the detection of the well separated clusters but the resulting clusters have been further splitted to get clusters with small volumes. The main benefit of the resulted partitioning is that it can be easily approximated by mixture of multivariate Gaussians (ellipsoids). This approximation is useful since the obtained Gaussians give a compact and parametric description of the clusters, and the result of the clustering is soft (fuzzy). Fig. 3 shows the results of the clustering we obtained after performing the iteration steps of the Gaussian mixtures based EM algorithm given in [8]. In this figure the dots represent the data points and the 'o' markers are the cluster centers. The membership values are also shown, since the curves represent the isosurfaces of the membership values that are inversely proportional to the distances. As can be seen, the clusters provide an excellent description of the distribution of the data. The clusters with complex shape are approximated by a set of ellipsoids. It is interesting to note, that this clustering step only slightly modifies the placement of the clusters. In order to demonstrate the effectiveness of the proposed initialization scheme, Fig. 5 illustrates the result of the Gaussian mixture based clustering, where the clustering was initialized by the classical fuzzy *c*-means (FCM) algorithm. As can be seen, this widely applied approach failed to find the proper clustering of the data set, only a sub-optimal solution has been found. The difference between these two approaches can be seen in the dendrograms as well (Fig. 4 and Fig. 5).

V. CONCLUSION

The best-known graph-theoretic divisive clustering algorithm is based on construction of the minimal spanning tree (MST). This paper presented a new splitting criterion to improve the performance of this MST based clustering algorithm based on the calculation of the hyper volume of the clusters that are approximated by a multivariate Gaussian functions. The result of this clustering can be effectively used for the initialization of Gaussian mixture model based (Gath-Geva) clustering algorithms. Illustrative clustering results showed the advantages of the proposed hybridization of the hierarchical graph-theoretic and partitional model based clustering algorithm. The chaining effect of the MST and the sensitivity to the initialization of the Gath-Geva clustering algorithms have been properly handled, and the resulted clusters are easily interpretable since the compact parametric

description of the multivariate Gaussian clusters (fuzzy covariance matrices). The resulted fuzzy clusters can be effectively analyzed based on the proposed similarity criterion. The proposed MST-GG algorithm has been implemented in MATLAB, and it is downloadable from the website of the authors: www.fmt.vein.hu/softcomp.

REFERENCES

- [1] Augustson J.G., Minker J., "An analysis of some graph theoretical clustering techniques", *J. ACM* Vol.17, 4, pp. 571-588., 1970.
- [2] Backer F.B., Hubert L.J., "A graph-theoretic approach to goodness-of-fit in complete-link hierarchical clustering" *J. Am. Stat. Assoc.* Vol. 71, pp. 870-878., 1976.
- [3] Barrow, J.D., Bhavsar, S.P., Sonoda, D.H., "Minimal spanning trees, filaments and galaxy clustering" *In Monthly Notices of the Royal Astronomical Society*, 216:17-35., 1985.
- [4] Bezdek, J.C., Ehrlich, R., Full, W., "FCM:Fuzzy C-Means Algorithm", *Computers and Geoscience*, 1984.
- [5] Bezdek, J.C., Clarke, L.P., Silbiger, M.L., Arrington, J.A., Bensaïd, A.M., Hall, L.O., Murtagh, R.F., "Validity-guided (re)clustering with applications to image segmentation" *IEEE Transactions on Fuzzy Systems*, 4:112-123., 1996.
- [6] Dunn, J.C., "Well separated clusters and optimal fuzzy partitions", *Journal Cybernetics*, 4: 95-104., 1974.
- [7] Forina, M., Oliveros, C., Concepcin M., Casolino, C., Casale, M., "Minimum spanning tree: ordering edges to identify clustering structure" *In Analytica Chimica Acta*, 515: 43-53., 2004.
- [8] Gath I., Geva, A.B. "Unsupervised optimal fuzzy clustering", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11(7), 1989.
- [9] Gonzales-Barrios, J.M., Quiroz, A.J., "A clustering procedure based on the comparison between the k nearest neighbors graph and the minimal spanning tree" *In Statistics & Probability Letter*, 62: 23-34., 2003.
- [10] Gower J.C., Ross G.J.S., "Minimal Spanning Trees and Single Linkage Cluster Analysis", *Applied Statistics*, Vol. 18, pp. 54-64., 1969.
- [11] Jain, A.K., Dubes, R.C., "Algorithms for Clustering Data", *Prentice-Hall advanced reference series*, Prentice-Hall, Inc., Upper Saddle River, NJ., 1988.
- [12] Kruskal, J.B., "On the shortest spanning subtree of a graph and the traveling salesman problem" *In American Mathematical Society*, 7: 48-50., 1956.
- [13] Päivinen, N., "Clustering with a minimum spanning tree of scale-free-like structure", *In Pattern Recognition Letters*, 26(7): 921-930., 2005.
- [14] Prim, R., "Shortest connection networks and some generalizations" *Bell System Technical Journal*, Vol. 36, pp. 1389-1401., 1957.
- [15] Raghavan V.V., YU C.T., "A comparison of the stability characteristics of some graph theoretic clustering methods", *IEEE Trans. Pattern Anal. Mach. Intell.* 3, pp. 3-402., 1981.
- [16] Varma, S., Simon, R., "Iterative class discovery and feature selection using Minimal Spanning Trees" *BMC Bioinformatics*, Vol. 5, pp. 126-134., 2004.
- [17] Zahn, C.T., "Graph-theoretical methods for detecting and describing gestalt clusters" *IEEE Transaction on Computers*, C20: 68-86., 1971.

Janos Abonyi received the M.Eng. and Ph.D. degrees in chemical engineering from the University of Veszprem, Veszprem, Hungary, in 1997 and 2000, respectively. Currently, he is the Head of the Department of Process Engineering of the University of Veszprem. From 1999 to 2000, he was a Research Fellow at the Control Laboratory at Delft University of Technology, Delft, The Netherlands. His research interests include data mining and the applications of fuzzy systems, genetic algorithms, and neural networks in process engineering and data-mining. He has authored the research monograph *Fuzzy Model Identification for Control* (Boston, MA: Birkhäuser, 2003), and more than 60 journal papers and chapters in books in these areas.