1. Introduction

Clustering is an unsupervised classification method to partition a collection of multivariate data points into meaningful groups, where all members with in a group represent similar characteristics and data points between different groups are dissimilar to each other (Tsai & Lin, 2011). There are many methods and algorithms for clustering, based on crisp, fuzzy, probabilistic and possibly approaches (Rezaee, 2010), i.e. k-means, c-means, hierarchical clustering. Fuzzy c-means clustering algorithm (FCM) is one of the popular clustering algorithms. FCM combines the c-means approach with the handling of the existing fuzziness in the data. This combination makes it more powerful, because fuzziness of the data affects the results in a disadvantageous way when creating crisp partitions. In general, soft clustering techniques aim to eliminate this situation and also, FCM is a kind of soft clustering method which is based on fuzzy set theory (Zadeh, 1965). In practical applications of clustering algorithms, several problems must be resolved, including determination of the number of clusters and evaluation of the quality of the partitions (Rezaee, 2010). Moreover, as stated by (Kim, Lee, & Lee, 2004), three major difficulties were drawn attention in fuzzy clustering: (1) determining the natural number of clusters to be created (most algorithms require the user to specify the number of clusters); (2) choosing the initial cluster centroids (most algorithms choose a random selection because such a selection is sure to converge the iterative process); (3) handling data characterized by large variations in cluster shape, cluster density, and the number of points in different clusters (Kim et al., 2004).

In this study, the tool, FUAT (fuzzy clustering analysis tool), is proposed to explore the clusters created with FCM clustering. The reason for the development of FUAT is sourced from the reported difficulties of FCM. First, FCM requires the cluster number as an input parameter but, to know this number is difficult for the decision maker in fact. Because, to forecast the distribution of the data points, which is gained from real world, in the space is hard job and sometimes impossible. In FUAT, natural clustering is embedded to give an advice to the user about possible cluster number. Second, the initial clusters have great effect on the resulted clusters. However, neither getting the resulted cluster centroids nor presentation of data with clusters numbers, and membership degrees is enough to assess clustering performance. Since, size or densities of the clusters, saturation and frequencies of the membership degrees in the clusters, closeness between clusters, intersection size or densities between clusters are required for performing detailed analyses on the clusters and parameters and, assessment of the clustering performance.

The problems and the critical points about fuzzy clustering have been discussed in literature, especially these studies are focused on the subject of validity index. For example partition coefficient (PC) (Bezdek, 1974a) and partition entropy (PE) (Bezdek, 1974b) are basic, simple, but efficient indices which are based on the fuzzy membership values of fuzzy partitions. Moreover, researchers have suggested many cluster validity indices that include both fuzzy membership values and the information of structures (Zalik, 2010). Most indices about validity employ compactness and separation concepts. Compactness is related with the closeness within the cluster, and separation is related with the isolation of clusters between each other. In other words, validity index for fuzzy clustering tries to reflect the ratio of overcoming the difficulties
specified in Rezaee (2010). In fact, validity index in a necessity, because of the black box usage of fuzzy clustering algorithms and, their dependencies to initial parameters and structure. Before clusters obtained by fuzzy clustering algorithm, validity checking can be done by numerically by using selected validity index.

In fact FUAT has a complementary approach to concept of validity index. It is a tool and shows many characteristics of resulted clusters (compactness, separation, overlapping, case distribution, density) visually. In other words, by FUAT, we tried to convert effectively FCM based clusters from black box to transparent boxes for the users. Especially, we concentrate on the creating ability of the clusters analysis separately and all together for helping users to overcome difficulties of FCM usage as a black box. In FUAT design all characteristics of FCM are kept and different data types (integer, real) are supported.

2. Theory

In this study, two important clustering schemes are employed together. Fuzzy c-Means (FCM) and Expectation Maximization (EM) based clustering methods are used because of their soft clustering behaviors. Their major characteristics are explained below.

2.1. Fuzzy c-means clustering

Generalized Fuzzy c-means (FCM) (Bezdek, 1981) is one of the most popular unsupervised fuzzy clustering algorithm, which is widely used in pattern recognition, image recognition, gene classification, etc (Jingwei & Meizhi, 2008). As can be understood from the name of FCM, it is based on Zadeh’s (Zadeh, 1965) fuzzy set theory and applies c-means clustering approach. By FCM, fuzzy clusters are constructed in that way, ith data, \( x_i \), belongs to jth cluster, \( F_j \), with degree of \( \mu_{ij}(x_i) \). In FCM, data points are partitioned into the c clusters by the minimization of the distance between data points, and the fuzzy cluster centroids iteratively. The general algorithm of FCM is as follows:

Specify number of clusters
Do

Compute centroids of clusters
For each case

Compute membership degrees of case to clusters
While convergence criteria is not met

In FCM, centroid of ith cluster \( (c_i) \) is calculated with Eq. 1.

\[
\text{Specifying number of clusters } \quad \text{Do} \\
\text{Compute centroids of clusters} \\
\text{For each case} \\
\text{Compute membership degrees of case to clusters} \\
\text{While convergence criteria is not met} \\
\]

\[
c_i = \frac{\sum_{j=1}^{N}(\mu_{ij})m \cdot x_j}{\sum_{j=1}^{N}(\mu_{ij})m} \tag{1}
\]

where \( N \) is number of data, \( x_j \) is jth case in N and \( \mu_{ij} \) is the membership degree of jth case to ith cluster. At this point, \( m \) parameter is used as the coefficient of distance, and it enables to control fuzziness of the clustering and value of 2 is suggested in (Bezdek, 1999). Let assume that \( \mu_j \) value is calculated with Eq. (2).

\[
\mu_j = \frac{1}{\sum_{k=1}^{c}(\frac{|x_j - x_k|^2}{m - 1})} \tag{2}
\]

In general definition of FCM, Euclidean distance measurement is used to calculate distance between vectors, and it converts to the data space spherical view. However, it is possible to replace the distance measurement method with another one. The stopping criterion of FCM is dependent on the change in the vector of membership degrees from previous iteration to completed iteration. In other words, the change in membership values is smaller than convergence criteria value that the algorithm stops.

2.2. Expectation maximization

EM (Expectation Maximization) algorithm is an unsupervised clustering method based on discovering the appropriate parameters of a particular statistical model which defines the data. The employed model in this process is called as mixture models which view the data as a set of cases from a mixture of different probability distributions and are being modeled by using a number of statistical distributions that each represents a cluster (Tan, Steinbach, & Kumar, 2005). Moreover, as stated previously in (Tan et al., 2005) that, parameters of each distribution provide a description of the corresponding cluster.

In general lines, EM procedure consists of two important iterative steps: E-step and M-step. Probability of being a member of each case to each cluster is computed at E-Step (Expectation). At the next stage (Maximization), parameter vector of the probability distribution of each cluster is re-approximated. This iteration based mechanism finishes when the maximum number of iterations, or accepted error range to converge is reached. As a result of this procedure, total natural clusters can be obtained without specifying a cluster number.

Consequently, EM based clustering is segmentation method which utilizes maximum likelihood concept. On the other hand, similar to fuzzy clustering, it owns soft segmentation characteristic because of a point being member of more than one cluster with certain probability. Due to these facts and let the users know the probably true number of clusters in data, EM based clustering schema is included to FUAT. More detailed explanations about EM based clustering can be found at (DMR, 2011; Tan et al., 2005).

3. Components utilized

In development of the software subjected to this study, various components are utilized. They are listed below:

3.1. R

In this study, R (http://www.r-project.org), a famous and well known statistical computing program, is employed for EM algorithm usage. As (Zupan & Demsar, 2008), reported that, R involves many techniques for statistics, predictive modeling and data visualization, and has become a de facto standard at open source library for statistical computing. The main benefit of R is having a script language inherited from S (Becker & Chambers, 1984), which enables users to program what they need in clear way. Further, R has got an extensive variety of freely accessible modules for various purposes located at “The Comprehensive R Archive Network” (CRAN, http://cran.r-project.org).

As the R mainly supports command line scripting, it has two important advantages: (1) whole analysis procedure can be operated by clearly stated definitions, and they can be stored for later use; (2) R can be accessed and directed via COM interfaces in supported programming languages and platforms such as C++, VB or .NET. Therefore, a seamlessly integrated component called R(D)COM which bridges a wide range of different languages (e.g., C#, C++, Python, VBA, VB, Java) to R is developed by Statconn (Statconn, 2011). Further, it supports various numbers of platforms (COM/DCOM, .NET, Uno, C, Web Services SOAP/http) to create and integrate solutions (Statconn, 2011). Statconn also provides other components called RExcel which enables usage of R functions in Excel natively and ROOo at Open Office environment.
Statconn's R(D)COM allows for transferring of variables, arrays, commands and even graphics between the programming languages mentioned above and R. However, one should be aware of the type of the objects which will be sent to R and does a type check at R environment to ensure that R evaluates them in right type.

In this study, to be able to suggest the natural cluster number to user in selected data, EM algorithm in MClust package is employed via interfacing R and C#.NET by R(D)COM.

3.2. MClust

MClust is a free, non-commercial and contributed R package designed for normal mixture modeling and model-based clustering by providing methods for parameter estimation via the EM algorithm for normal mixture models with a variety of covariance structures (Fraley & Raftery, 2006). It is developed and licensed by University of Washington and distributed through CRAN. In general lines, MClust presents three types of cluster analysis schemes: (1) combining model based hierarchical clustering; (2) EM for Gaussian mixture models; (3) estimation of clusters by BIC (Bayesian Information Criterion) for parameterized mixture models (Fraley, 2006). On the other hand, MClust package is equipped with functionality for displaying, visualizing and simulation of the models such as uncertainty plots.

As one of the design consideration of FUAT is to find out the possible natural cluster number prior to fuzzy clustering stage. MClust package is utilized because of its ability to reveal it by the help of its Bayesian Information Criterion (BIC) computation. With the help of this metric, MClust computes the natural cluster number.

3.3. ZedGraph

ZedGraph is a .NET based, open source charting class library for plotting wide variety of charting types for both desktop and web based platforms (ZedGraph, 2011). ZedGraph mainly includes bar, stack, line and curve based charts by presenting rich drawing and decoration options. Moreover, it provides native .NET API for both Windows and Web environments. Due to these features, ZedGraph is selected as the main charting solution in this study.

3.4. Microsoft GLEE

Microsoft GLEE (Graph Layout Execution Engine) is a non-commercial .NET tool for lying out and visualizing directed graphs (Microsoft GLEE, 2011). Actually Microsoft GLEE is the freeware and limited version of Microsoft Automatic Graph Layout (MSAGL) tool which employs principle of the Sugiyama approach that generates layered or hierarchical layouts aiming maximization of space. However it implements novel methods and presents features such as heuristic for creating a graph layout with a given aspect ratio and an efficient method of edge-crossings counting in performing adjacent vertex swaps (GLEE Technical Notes, 2011).

In general lines, GLEE includes three important components: (1) layout engine, (2) drawing module and (3) viewer control. Further, it presents a clear and understandable API to define nodes, links and specify layout manager in C#, VB.NET and C++.NET environments. Besides, Microsoft GLEE module serves high performance graph production performance. It is reported that, graphs which have 15,000 edges or 50,000 elements can be rendered in reasonable time (Microsoft GLEE, 2011).

As FUAT requires a cluster dependency viewer and GLEE is a native .NET module, it is clearly chosen for interactive visualization of fuzzy clusters and degree of their neighborhoods. Moreover, its ability to output graph images in popular image formats (i.e. JPEG or PNG) has played an important role in preferring this .NET based integrated tool.

4. FUAT

As stated previously, several fuzzy clustering algorithm packages exist in open source and commercial products. However, they generally focus on the underlying method in segmentation of data in terms of different or novel approaches. Additionally, they mostly visualize or report the attributes of the clusters. Therefore, they generally have the lack of some features related to discovering and revealing the hidden relationships of the individual observations belong to owner clusters. Beyond, fuzzy clustering approach relies on membership concept. Therefore, exploring the interactions of the soft clusters and their respective elements (individual observations) become more important than ever. The findings sourced from clustering results have significant advantages such that the cases' membership values of an outlier cluster can lead to important decisions. Another instance that can be easily given is imbalanced data exploration. In case of imbalanced data, segmentation of minorities should be tracked carefully than the major ones. These and other related reasons became important motivation for us to focus on this issue.

In light of this information, FUAT, a fuzzy based cluster explorer and analyzer tool is designed and implemented. Whole study is constituted on C# language on .NET Framework 2.0 platform. The only and valid data format is selected as comma separated values (CSV) files. All the features of the study are grouped in three visualizations: (1) data loading tab; (2) fuzzy clustering tab; (3) viewer and analyzer tab. To be explained detail, FUAT currently has the following properties and facilities:

(1) Fuzzy c-means based fuzzy cluster algorithm supporting “Euclidean” and “Manhattan” distances,
(2) BIC based automatic cluster number estimator,
(3) Cluster population viewer,
(4) Parallel coordinates style dimensional centroids of clusters viewer,
(5) Membership histogram
(6) Point of interest viewer
(7) Graph based cluster neighborhood viewer.

All of these features are explained in details at rest of this section. Meanwhile, all figures are generated at the analyze stage of a sample dataset named “Concrete” located at UC Irvine Machine Learning Repository (UCI, 2011).

4.1. Fuzzy c-means clustering engine

In this study, fuzzy c-means algorithm is chosen as the main clustering engine therefore instead of using a third party code or package, whole FCM module is implemented by authors. Fuzziness factor, number of clusters, accuracy of convergence, maximum iteration count and distance type, “Euclidean” and “Manhattan” distance types are included as input parameters of FCM.

At first, instead of assigning random continuous numbers, randomly chosen cases are selected as cluster centroids. Then, when there is no change higher than specified accuracy threshold in membership values, rest of the clustering algorithm works as it supposed to. Membership values and respective cluster labels are assigned to cases upon completion of algorithm and they are viewed at the table view (Fig. 1).

4.2. Cluster number estimation

Cluster number detection is crucial when there is no domain expert or prior information about cluster study. Especially, in case of fuzzy or k-means clustering which pre-requires cluster number, it...
becomes significant. Therefore, an automatic cluster number detector (ACND) is designed and included in this tool. As one of the main considerations of this study is to create an easy to use environment even for non-domain experts, ACND feature is designed and implemented as easy as possible for users. With just one click, FUAT connects to local R server and send the data and get the number of natural components inside of it.

At this stage, MClust package and EM algorithm are employed in the process which is responsible from determination of natural cluster number. However, at the backstage, Bayesian Information Criterion is utilized in parameterization and number of cluster selection. Actually, the BIC is defined as “the value of the maximized log likelihood with a penalty for the number of parameters in the model, and allows comparison of models with differing parameterizations and/or differing numbers of clusters” (Fraley, 2006). Higher the value in BIC has the meaning of strength in evidence of model and number of components in data set (See Fig. 2). As stated in (Fraley, 2006), by default, MClust compares BIC values for parameters optimized for up to nine components (clusters) and all ten covariance structures currently available in the MClust software. However, it can be extended by making small configuration changes. Estimation of BIC is given as follows:

$$BIC = 2 \log \text{lik}_{M}(x, \theta_{k}) - (# \text{params})_{M} \log(n)$$

According to (Fraley, 2006), $x$ is the data and $n$ is the total number of cases in this data, $\log \text{lik}_{M}(x, \theta_{k})$ represents the maximized log likelihood of data and model and finally $# \text{params}$ is the number of model's ($M$) independent parameters which will be computed. As stated before, MClust, provides different covariance matrix structures in four points of view: (1) distribution (univariate, spherical, diagonal, ellipsoidal), (2) volume (equal, variable), (3) shape (equal, variable) and as the last one (4) orientation (coordinate axes, equal, variable).

One of the main reasons of preferring BIC based EM clustering is assumption of the FUAT users that they have no prior information about the distribution on working datasets and need a global and robust solution. In the implementation of ACND, R(D)COM component is highly utilized as it plays a “bridge” role in this R - C#.NET collaboration. The code sample listed below, represents how C# access to R via R(D)COM. There are two important details which must be kept in mind: (1) primitive objects, arrays and graphic objects are transferrable on R(D)COM and (2) “mixclust”, an object created by MClust(mix), has some properties including $G$ = detected cluster number. The way of connection establishment between R and C# is sampled following code segment.
int CLcnt = 0; //Cluster number
rconn.Evaluate("library(mclust)"); //include MClust
rconn.SetSymbol("z", my_data);
//my_data is a two dimensional double array
rconn.Evaluate("mix = as.matrix(z)");
rconn.SetSymbol("answer", answer);
rconn.EvaluateNoReturn("mixclust = Mclust(mix)");
rconn.EvaluateNoReturn("answer <- mixclust$G");
CLcnt = (int)((object)rconn.GetSymbol("answer"));

4.3. Cluster population viewer

The most fundamental viewer of FUAT is the cluster population viewer to present the distribution of clusters in population point of view. (See Fig. 3). ZedGraph’s bar chart features are utilized in this viewer by adding the support of saving chart image. As can be seen in Fig. 3, number of cases in each cluster is indicated as bar graph.

4.4. Cluster centroids viewer

One of the important findings of the clustering results is the cluster centroids. Therefore, cluster centroids viewer is implemented in FUAT by using parallel coordinates approach. According to Oliveira and Levkowitz (2003), parallel coordinates visualization technique projects n-dimensional data onto the two dimensional plane by drawing n equally spaced axes parallel to one of the display axes. Meanwhile, each of the axes corresponds to a feature and is linearly scaled within its respective data range. Then, each data item is drawn as a polygonal line which intersects each axis at the point corresponding to the item’s associated feature value (Oliveira & Levkowitz, 2003). In their study, Oliveira and Levkowitz (2003) point out the effectiveness of parallel coordinates technique in revealing a wide range of data characteristics, such as different data distributions and functional dependencies. Therefore, parallel coordinates technique is adopted for the centroids viewer. In principle, all the attribute names are located at x axis without an order and the average value of the attributes for all clusters are drawn onto intersected y axis to build line segments for each cluster. ZedGraph’s available zooming; panning and mouse hover labeling features are very helpful at this point for precisely investigations. As can be seen at Fig. 4, cluster centroids are plotted for each the attributes, hence users enabled to realize significant and discriminator features at that clustering session.

4.5. Membership histogram

In fuzzy clustering and other soft clustering techniques, it is common that the observations to be member of more than one distinct cluster. Instead of hard clustering methods, in soft clustering approaches every observation is a member of every cluster with a certain membership degree. The highest membership degree identifies the final cluster of observation to be assigned. In this schema, investigation of memberships becomes important such as sparse clusters have low membership values and dense clusters own higher membership valued observations. Thus, exploration of these membership distributions can lead effective description of obtained clusters. For these reasons, it is decided to implement a membership histogram viewer in this study.

As can be seen at Fig. 5, three clusters are shown with their membership histogram values. While the clusters colored red and blue are broadly located in 30%-50% range, green colored cluster performs more characteristic distribution. Its distribution starts from 33% levels and ends at 88%. Thus, it can be concluded that green colored cluster has a denser structure.

4.6. Point of interest viewer

In this viewer, it aimed to show two-dimensional projections of clustered observations. However, shown observations must meet...
some requirements (filters). A simple but effective membership value based filtering algorithm is developed. Algorithm's details are as follows:

1. Filtering mechanism scans all membership matrix and removes (resets) the values lower than specified “minimal membership percentage” value.
2. In each row of membership matrix, membership values are sorted at decreasing order and difference of top two ones are calculated.
3. If the calculated difference is lower than “membership difference threshold” which can be set by user, this point is plotted onto x-y graph with appropriate cluster color.

With the help of this viewer, close membership valued cases can be identified and this leads discovering uncertainty and the uncertain observations. On the other hand, by specifying high “minimal membership percentage” ratios, only the hard clustered observations can be revealed. Moreover, important features that will also play role at classification tasks can be identified by investigating 2D mappings.

Fig. 4. Cluster centroids viewer.

Fig. 5. Membership histogram viewer.
As can be seen at Fig. 6, colored cluster observations are located at different places in space. Thus, it can be concluded that selected features are good discriminator features.

4.7. Cluster neighborhood viewer

In the other viewers of FUAT, except the cluster population viewer, all the explorations are done at cases level. As the goal of this study states, FUAT also aims to discover the relationships between clusters. Therefore, cluster neighborhood viewer is designed and implemented due to it is a crucial requirement.

In FUAT, the main principle of cluster neighborhood analysis is based on determining similar membership valued clusters. Thus, the overall membership table is scanned regarding the user selected single cluster and the observations that have similar membership degrees (fall in the range given) constitutes neighborhood clusters. The strength of neighborhood is determined by the number of observations which two clusters share in terms membership degree similarity.

As can be seen on Fig. 7, three clusters are depicted with ellipsoidal nodes and the strength of neighborhoods is figured with black arrows (edges). In Fig. 7, it is seen that cluster 1 is selected.
as the target cluster prior to analysis and cluster 2 is more similar to cluster 1 than cluster 0 in the range of 0.03% membership degree difference. In other words, this figure indicates that cluster 1 and cluster 2 share 30 cases which are inseparable at > 0.03% membership value difference. Similarly, cluster 1 and cluster 0 share only 11 cases. Thus, cluster 1 shares more “common” cases with cluster 2 than cluster 0. Based on this observation, it is concluded that cluster 1 owns a stronger neighborhood with cluster 2. As can be derived from these explanations, membership values are used as a metric that defines distances among the clusters and with the help of the parameter “minimal membership difference” the distance range is specified.

Microsoft GLEE module is heavily utilized in this viewer. Its native viewer supports features zooming, panning and saving rendered image in popular graphic formats. However, at the behind of the scene, whole graph, created by the node/edge definition interface, is rendered by Sugiyama schema. As the numbers of nodes are very limited in this study, GLEE gave superior performance in rendering phase.

5. Conclusion
The usage of FCM is common as a clustering method. However, FCM has some difficulties including initialization of the first centroid vector, determination of cluster number, being dependent on data size and density. To analyze the effect of these difficulties on clusters and to overcome them, detailed analysis on resulted clusters becomes necessity. In literature, concept of validity index is presented for this purpose. However, it is a numeric value. Additionally, open source or commercial products do not have these types of facilities; mostly they focused on the accurate and robust implementation of FCM. In this study, the complementary visual tool called as FUAT is designed and developed for the practitioners of FCM. FUAT enables them making detailed analysis on clusters by using its abilities such as cluster population viewer, point of interest viewer, cluster neighborhood viewer and membership histograms.

By using FUAT, practitioners have some remarkable utilities for exploring fuzzy clusters such as (1) gaining insight at cluster centroids and discovering which attribute(s) cause(s) discrimination (2) discovering membership value distributions to understand characteristics of obtained clusters, (3) projecting individual cases over two dimensional attribute space for better understanding of case distribution over clusters and (4) unveiling the strength of hidden neighborhoods among clusters via presenting a membership difference threshold value as the last one's: two different clusters are real if from each other, which parameters are discriminants, how much cases are placed on the clusters boundaries or how much cases belong to each cluster i.e. The future planning for FUAT is to add some new abilities for the management of FCM algorithm effectively by specialization of distance measure and fuzziness control.

References
Tao, P. N., Steinbach, M., & Kumar, V. (2005). Introduction to Data Mining. Addison Wesley.