A MATLAB Toolbox and its Web based Variant for Fuzzy Cluster Analysis

Tamas Kenesei, Balazs Balasko, and Janos Abonyi

University of Pannonia, Department of Process Engineering,
P.O. Box 158, H-8201 Veszprem, Hungary, abonyij@fmt.uni-pannon.hu
www.fmt.vein.hu/softcomp

Abstract. *Nowadays due to the yearly multiplying data comes always the need for useful methods, algorithms, that make the processing of these data easier. For the solution of this problem data mining tools come into existence, to which clustering algorithms belong. The purpose of this paper is to propose a continuously extensible, standard tool, which is useful for any MATLAB user for one's aim. The toolbox contains crisp and fuzzy clustering algorithms, validity indexes and linear and nonlinear visualization methods for high-dimensional data. The web-based prototype version of the toolbox already has been developed. It means that users do not need to have MATLAB software and programming knowledge, but only a web browser and they can load their own data into the web server and download the results because the program codes run on Matlab Web Server with a developed data mining portal, which is found in the following url:* `www.pe.vein.hu/datamine`*. The portal is in test phase, with limited services. The Fuzzy Clustering and Data Analysis Toolbox with User's Guide is available at* `www.mathworks.com/fileexchange`.

1 Introduction

In this paper we propose a MATLAB toolbox for data analysis based on clustering and its application via Internet. Data analysis and data mining methods are more and more important because lots of data is being collected and warehoused in recent years since these data definitely have the potential to provide information. The definition of data mining is extraction of interesting (non-trivial, implicit, previously unknown and potentially useful) information or patterns from data in large databases. The tasks of data mining can be very different. We can group the data mining tools and processing algorithms in the following primary data mining methods: *Classification*, *Regression*, *Clustering*, *Summarization*, *Dependency Modeling*, *Change and Deviation Detection*.

Many MATLAB toolboxes have been developed in several research fields in recent years. A MATLAB toolbox is presented for Self Organizing Map in [1], another one is for Bayes Net in [2] and another is for dimensional analysis in [3]. The so called KERNEL toolbox can be used for knowledge extraction and refinement based on neural learning [4]. Robert Babuska developed a toolbox for fuzzy model identification [5]. These toolboxes are available in the Internet but none of them can be used without MATLAB. The web based variant of the proposed toolbox enables

MATLAB independent usage and it does not make demand on the client computers because it runs on the web server.

The proposed toolbox contains clustering methods and visualization techniques based on clustering. A cluster is a collection of data objects that are similar to one another within the same cluster and dissimilar to the objects in other clusters. Cluster analysis is grouping a set of data objects into clusters without any predefined classes so clustering is unsupervised classification. Clustering algorithms can be partitioning, hierarchy, density-based, grid-based or model-based methods.

Objective function based fuzzy clustering algorithms have been used extensively for various problems such as pattern recognition [6], data analysis [7], image processing [8] and fuzzy modelling [9]. Fuzzy clustering algorithms partition the data set into (partially) overlapping groups in a way that clusters describe an underlying structure within the data. To obtain a good result, a number of issues are of importance. These concern the shape and the volume of the clusters, the initialization of the algorithm, the distribution of the data patterns and the number of clusters.

This toolbox contains objective function based partitioning algorithms: they construct various partitions and then evaluate them by some criterion to minimize an objective function that is based on the distance between the cluster prototypes and the data points. The toolbox contains the k-means, k-medoid (crisp), fuzzy c-means, Gustafson-Kessel and Gath-Geva (fuzzy) clustering methods and other important tools such as methods for determining the number of clusters and for visualization of the clustering results.

The toolbox contains method for visualization of high-dimensional data. Visualization is a technique that projects data in higher dimensions to data in lower dimensions while trying to preserve the distances between all points. It can be very useful because it can (i) identify meaningful underlying dimensions that could explain similarities or dissimilarities in the data, (ii) detect underlying structure and (iii) reduction the data dimension and reveal relationships. Nonlinear mapping methods are often based on the results of a clustering algorithm so the clustering and visualization algorithms have a strong connection.

The so-called online data mining has greater and greater importance in our information society. For this purpose we developed a tool which enables us to use data mining methods via the internet, where the application is running in the server side not in the client computers, so the resources of client computers are free for other applications. For this purpose the client has to have only a web browser.

The paper is organized as follows. Section 2 presents the theoretical base of the toolbox and Section 3 gives application examples to prove the applicability of this Toolbox. Section 4 presents an application by which it can be used via internet without downloading and installing the toolbox. Section 5 contains the conclusions.

2 Fuzzy Clustering and Data Analysis Toolbox

The objective of cluster analysis is the classification of objects according to similarities among them, and organizing of data into groups. Clustering techniques are among the *unsupervised* methods, they do not use prior class identifiers. The main potential of clustering is to detect the underlying structure in data, not only for classification and pattern recognition, but for model reduction and optimization. Clustering techniques can be applied to data that is quantitative (numerical), qualitative (categoric), or a mixture of both. In this paper, the clustering of quantitative data is considered.

Since clusters can formally be seen as subsets of the data set, one possible classification of clustering methods can be according to whether the subsets are fuzzy or crisp (hard). Hard clustering methods are based on classical set theory, and require that an object either does or does not belong to a cluster. Fuzzy clustering methods allow objects to belong to several clusters simultaneously, with different degrees of membership. In many real situations, fuzzy clustering is more natural than hard clustering, as objects on the boundaries between several classes are not forced to fully belong to one of the classes, but rather are assigned membership degrees between 0 and 1 indicating their partial memberships.

Different classifications can be related to the algorithmic approach of the clustering techniques. In this work we have worked out a toolbox for the partitioning methods, especially for hard and fuzzy partition methods.

In the following part of this section we briefly discuss the applied and well-known clustering methods, validity indices and algorithms for visualization of clusters. As generic notation, c will denote the number of clusters, N the number of data points and n the dimension of each data point.

2.1 Clustering Algorithms

The **k-means** and **k-medoid algorithms** are hard partitioning methods and they are simple and popular, though them results are not always reliable and these algorithms have numerical problems as well. The k-means and k-medoid algorithms allocates each data point to one of c clusters to minimize the within-cluster sum of squares:

$$\sum_{i=1}^{c} \sum_{k \in A_i} ||\mathbf{x_k} - \mathbf{v_i}||_2 \tag{1}$$

where A_i is a set of objects (data points) in the i-th cluster and $\mathbf{v_i}$ is the mean for that points over cluster i. In k-means clustering the cluster prototype is a point. In k-medoid clustering the cluster centers are the nearest objects to the mean of data in one cluster

The **fuzzy c-means algorithm** (FCM) can be seen as the fuzzified version of the k-means algorithm and is based on the minimization of an objective function called *c-means functional*:

$$J(\mathbf{X};\mathbf{U},\mathbf{V}) = \sum_{i=1}^{c} \sum_{k=1}^{N} (\mu_{ik})^m ||\mathbf{x}_k - \mathbf{v}_i||_{\mathbf{A}}^2 \tag{2}$$

where $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_c]$, $\mathbf{v}_i \in \mathbf{R}^n$ is a vector of *cluster prototypes* (centers), which have to be determined, $D^2_{ikA} = \|\mathbf{x}_k - \mathbf{v}_i\|^2_A = (\mathbf{x}_k - \mathbf{v}_i)^T \mathbf{A} (\mathbf{x}_k - \mathbf{v}_i)$ is a squared inner-product distance norm, and the $N \times c$ matrix $\mathbf{U} = [\mu_{ik}]$ represents the fuzzy partitions, where μ_{ik} denotes the membership degree that the ith data point belongs to the kth cluster. Its conditions are given by:

$$\mu_{ij} \in [0,1], \ \forall i,k, \ \sum_{k=1}^{c} \mu_{ik} = 1, \ \forall i, \ \ 0 < \sum_{i=1}^{N} \mu_{ik} < N, \ \forall k. \tag{3}$$

FCM algorithm can find only clusters with the same shape and size because the distance norm A is not adaptive and it is often Euclidean norm (spherical clusters). The solution can be given by Lagrange multiplier method.

Gustafson-Kessel algorithm (GK) is the extended version of the standard fuzzy c-means algorithm by employing an adaptive distance norm, in order to detect clusters of different geometrical shapes in one data set. Each cluster has its own norm-inducing matrix $\mathbf{A}_i$. The objective function cannot be directly minimized with respect to $\mathbf{A}_i$, since it is linear in $\mathbf{A}_i$. This means that J can be made as small as desired by simply making $\mathbf{A}_i$ less positive definite. To obtain a feasible solution, $\mathbf{A}_i$ must be constrained in some way. The usual way of accomplishing this is to constrain the determinant of $\mathbf{A}_i$. Allowing the matrix $\mathbf{A}_i$ to vary with its determinant fixed corresponds to optimizing the cluster's shape while its volume remains constant so GK algorithm can find clusters with different shape but with the same size [10].

Gath-Geva algorithm (GG) is based on the fuzzy maximum likelihood estimation and it is able to detect clusters of varying shapes, sizes and densities. The cluster covariance matrix is used in conjunction with an ”exponential” distance, and the clusters are not constrained in volume. However, this algorithm is less robust in the sense that it needs a good initialization, since due to the exponential distance norm, it converges to a near local optimum [11].

2.2 Validation

Cluster validity refers to the problem whether a given fuzzy 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:

- Starting with a sufficiently large number of clusters, and successively reducing this number by merging clusters that are similar (compatible) with respect to some predefined criteria. This approach is called *compatible cluster merging* [12].
- Clustering data for different values of c, and using *validity measures* to assess the goodness of the obtained partitions.

Different scalar validity measures have been proposed in the literature, none of them is perfect by oneself, therefore we used several indexes in our Toolbox. Detailed description about the applied indexes can be found in the literature so we just make mention of them in this section: *Partition Coefficient (PC)*, *Classification Entropy (CE)*, *Partition Index (SC)*, *Separation Index (S)*, *Xie and Beni's Index (XB)*,*Dunn's Index (DI)* and *Alternative Dunn Index (ADI)*.

Note, that the only difference of *SC, S* and *XB* is the approach of the separation of clusters. In the case of overlapped clusters the values of *DI* and *ADI* are not really reliable because of re-partitioning the results with the hard partition method.

2.3 Visualization

The clustering-based data mining tools are getting popular, since they are able to ”learn” the mapping of functions and systems or explore structures and classes in the data. There are often high-dimensional data in practice and it can be practical if we can see the results of the clustering (e.g. for checking the the results or finding out the underlying structure of the data). For this purpose several methods can be used.

The **Principal Component Analysis** maps the data points into a lower dimensional space, which is useful in the analysis and visualization of the correlated high-dimensional data. This mapping is based on the eigenvector-eigenvalues decomposition of F covariance matrix and uses only the first few nonzero eigenvalues and the corresponding eigenvectors.[1]

The **Sammon mapping** method can be used for the visualization of the clustering results, which preserves interpattern distances. This mapping method finds N points in a q-dimensional data space, where the original data are from a higher n-dimensional space. The interpoint distances measured in the n-dimensional space approximate the corresponding interpoint distances in the q-dimensional space. This is achieved by minimizing an error criterion called Sammon's stress using e.g. gradient-descent method. [13].

To avoid the high computational of Sammon mapping, a modified Sammon mapping algorithm is used in this work. The **fuzzy Sammon mapping** method uses the basic properties of fuzzy clustering algorithms where only the distance between the data points and the cluster centers are considered to be important [9]. The modified algorithm takes into account only $N \times c$ distances, where c represents the number of clusters, weighted by the membership values. This means, in the projected two dimensional space every cluster is represented by a single point, independently to the form of the original cluster prototype.

3 Application of the Toolbox

3.1 Comparing the Clustering Results

Using the validity measures mentioned in Section 2.2 the partitioning methods can be easily compared. For illustration, a synthetic data set was used shown in Fig. 1, Fig. 2 so the index-values are better demarcated at each type of clustering. These validity measures are collected in Table 1.

First of all it must be mentioned, that all these algorithms use random initialization, so different runs issue in different partition results, i.e. values of the validation measures. On the other hand the results hardly depend from the structure of the data, and no validity index is perfect by itself for a clustering problem. Several experiment and evaluation are needed that are not the proposition of this work.

	PC	CE	SC	S	XB	DI	ADI
K-means	1	NaN	0.095	0.0001	3987.4	0.0139	0.0004
K-medoid	1	NaN	0.2434	0.0003	Inf	0.0037	0.0036
FCM	0.8282	0.3470	0.9221	0.0008	19.6663	0.0175	0.0119
GK	0.8315	0.3275	0.8697	0.0009	32.1243	0.0081	0.0104
GG	0.9834	0.0285	2.2451	0.0020	2.5983	0.0160	0.0084

Table 1. The numerical values of validity measures

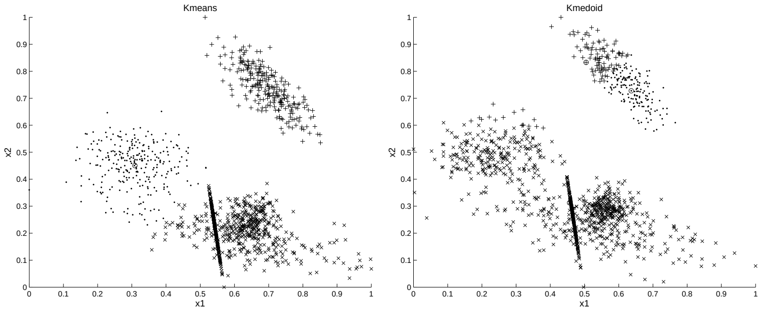

Fig. 1. Result of k-means and k-medoid algorithms by the synthetic overlapping data with normalization.

Fig. 1 shows that hard clustering methods also can find a good solution for the clustering problem, when it is compared with the figures of fuzzy clustering algorithms. On the contrary in Fig. 1 one can see a typical example for the initialization problem of hard clustering. This caused the differences between the validity index values in Table 1, e.g. the Xie and Beni's index is infinity (in "normal case" the k-medoid returns with almost the same results as K-means). The only difference

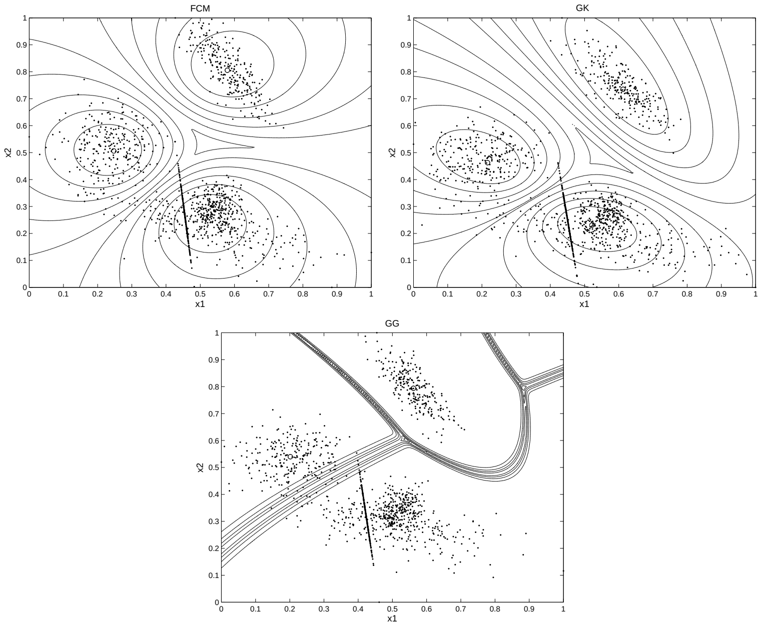

Fig. 2. Result of FCM, GK and GG algorithms by the synthetic overlapping data with normalization.

between the results of FCM and GK Fig. 2 stands in the shape of the clusters, while the Gustafson-Kessel algorithm can find the elongated clusters better. Fig. 2 shows that the Gath–Geva algorithm returned with a result of three subspaces.

As one can see in Table 1, PC and CE are not applicable for K-means and K-medoid, while they are hard clustering methods. But that is the reason for the best results in S, DI (and ADI), which are useful to validate crisp and well separated clusters. On the score of the values of the two ”most popular and used” indexes for fuzzy clustering (Partition Coefficient and Xie and Beni’s Index) the Gath-Geva clustering has the very best results for this data set.

3.2 Visualization Results

In order to examine the performance of the proposed clustering methods a well-known multidimensional classification benchmark problem is presented in this section: wine data. This data set comes from the UCI Repository of Machine Learning Databases. Cause of the too many data points there is no use to show the partition matrixes in tables, so the results of the *n*-dimensional clustering was projected into 2-dimension, and the 2-D results were plotted. Considering that projected figures are only approximations of the real partitioning results, the difference between the original and the projected partition matrix is also represented, and on the other hand one can observe the difference between the PCA, Sammon’s mapping and the Modified Sammon Mapping too, when these values are comprehended.

The detailed projection methods are based the results of a clustering algorithm. Using the proposed toolbox the best clustering algorithm can be chosen easily for this purpose. In the case of the wine data set the fuzzy c-means clustering has the best stable results correspond to the misclassified objects, so its resulting figures are shown in the following.

The Wine data contains the chemical analysis of 178 wines grown in the same region in Italy but derived from three different cultivars (marked with '.','x' and '+'). The problem is to distinguish the three different types based on 13 continuous attributes derived from chemical analysis.

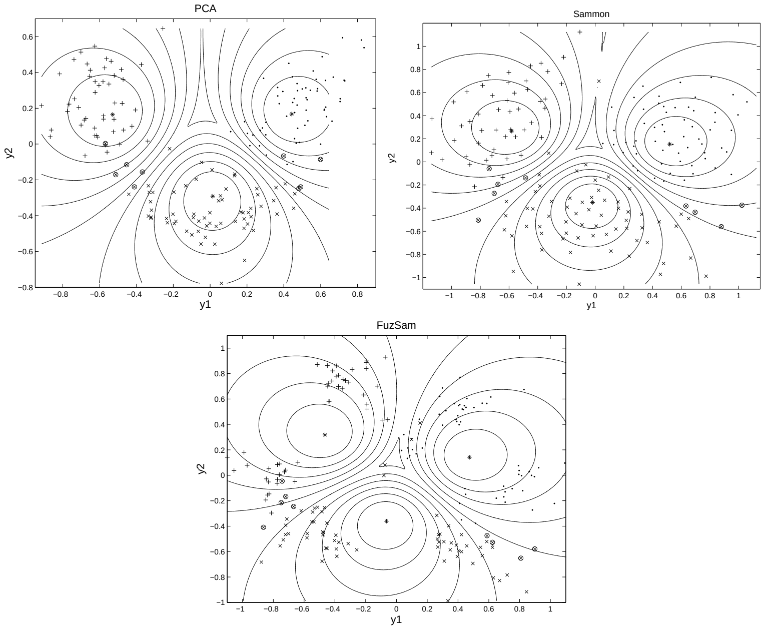

Fig. 3. Result of PCA, Sammon's Mapping and Fuzzy Sammon Mapping projection by the Wine data set.

	$\mathbf{P} = \overline{\lVert \mathbf{U} - \mathbf{U}^* \rVert}$	$\sum_{k=1}^{N} \overline{\mu_k^2}$	$\sum_{k=1}^{N} \overline{\mu_k^{2*}}$	E
PCA	0.1295	0.5033	0.7424	0.1301
Sammon	0.0874	0.5033	0.6574	0.0576
FuzSam	0.0365	0.5033	0.5170	0.0991

Table 2. Relation-indexes on Wine data set.

As Table 2 shows, Fuzzy Sammon Mapping has much better projection results by the value of **P**, which measures the difference between the original and projected membership matrices, than Principal component Analysis, and it is computationally cheaper than the original Sammon Mapping. So during the evaluation of the partition the figures created with this projection method were considered. We calculated the original Sammon's stress for all the three techniques to be able to compare them.

4 Web-based Version of The Toolbox

This section presents a solution for using the proposed toolbox without any downloading and installation. It is an user friendly way via the Internet becouse the users do not need to have MATLAB and do not need to be competent in programming languages.

The final goal of data mining is the extraction useful information and knowledge from data. Knowledge is the ability of people to learn from information and react faster and better than their competitors. Devices and methods of data acquisition, management, analyzing and forwarding to the right places can be prime importance in nowadays intensive market competition.

Corporations have to form their large databases, data warehouses and external sources to store knowledge. In the following we want to show the integration methodology of the data sources and the soft computing tools.

We store data in databases or in a data warehouses (DW), where programs with well designed GUI support fast and flexible working. If we want to work with the stored data, we need the above front-end applications to connect the DW, so clent computers needs these installed applications.

The basic idea is to use a web browser to analyze data of a complex system. Clients need only a web browser; the workflow application supplies the processing methods and the connection to the stored data.

So the work of client programs can be fullfilled by developing web based workflow applications, which can be maintained easily with system administrators, and provides the simplicity of installed client programs.

Our main goal is to provide a web – based user friendly interface to our toolbox. While Matlab Web Server is available in our Department, it is obvious to use the advantages of this technology. To develop dinamic user – friendly sites we use PHP[1], to store user data we use MySQL server. Summarizing the technologies, PHP create dinamic sites, database store user settings and Matlab Web Server deals with data processing, so the following componets are needed to create a web based workflow system with MATLAB:

- PHP interpreter (in CGI or in server modul format)
- Web server (either Apache or IIS)
- Database manager (MySQL)
- Matlab Web Server
- Web server

[1] It's free, and provides fast development

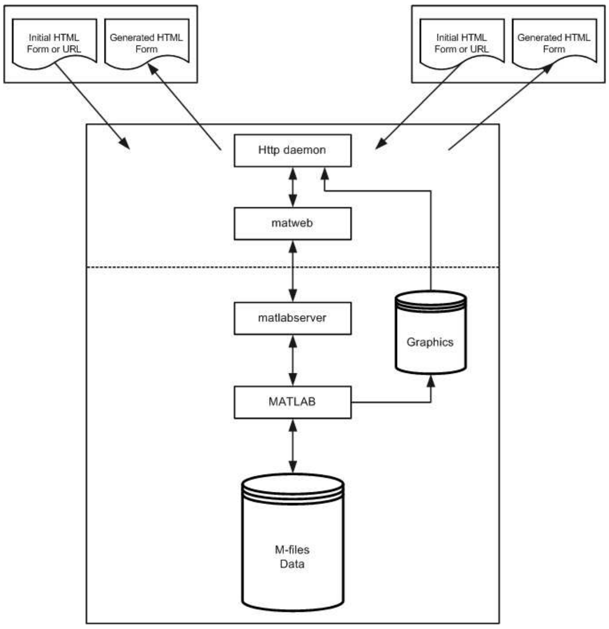

Fig. 4. MATLAB Web Server Components

Matlab Web Server[2] is a complex system to online data processing. Figure 4 shows how MATLAB operates over the Web. This structure is pieced together with the additional PHP applications, and the database storing.

Using this structure there is no need to compile MATLAB algoritms with MATLAB – C compiler, which is a very handy feature, becouse the C compiler couldn't be able to deal with structures, so it wasn't able to use the resources of the Matlab programming language.

Further benefits of this architecture is that there is no need to deep re-structuring of the implemented algorimts, only the *in* and *output*s have to be well determined and the core process method is in a basic M file. Matlab Web Server do the rest, if it is invoked by a special html form. After processing the data Matlab Web Server can return the results with using a template file.

5 Conclusions

To meet the growing demands of systematizing the nascent data, a flexible, powerful tools are needed. The Fuzzy Clustering and Data Analysis Toolbox provides several approaches to cluster, classify and evaluate wether industrial or experimental data sets. The software for these operations has been developed with MATLAB, which is very powerful for matrix-based calculations. The Toolbox provides five different

[2] http://www.mathworks.com

types of clustering algorithms, which can be validated by seven validity measures. High-dimensional data sets can be also visualized with a 2-dimension projection, hence the toolbox contains three different method for visualization. The web based version of this tool does not require to have MATLAB and the users' computers will be free for other applications because the program runs on the web server and the results can be downloaded from the server.

Acknowledgement

The authors would like to acknowledge the support of the Cooperative Research Centre (VIKKK) (project 2004-I) and Hungarian Research Found (OTKA T049534). Janos Abonyi is grateful for the support of the Bolyai Research Fellowship of the Hungarian Academy of Sciences.

References

1. Vesanto, J., Himberg, J., Alhoniemi, E. and Parhankangas, J., Self-organizing map in matlab: the som toolbox, Proceedings of MATLAB DSP Conference, Espo, Finland (1999) pp. 35–40.
2. K. Murphy, The bayes net toolbox for matlab, Computing Science and Statistics.
3. Brückner, S., The Dimensional Analysis Toolbox for Matlab, in: User's Manual, Stuttgart, http://www.sbrs.net/., 2002.
4. Castellano, G., Castiello, C. and Fanelli, A.M., KERNEL: A Matlab toolbox for Knowledge Extraction and Refinement by NEural Learning, 2000.
5. R. Babuska, Fuzzy modeling and identification toolbox for matlab, Delft University of Technology: Faculty of Information Technology and Systems.
6. W. Pedrycz, Fuzzy clustering with a knowledge-based guidance, Pattern Recognition Letters 25 (2004) pp. 469–480.
7. Szeto, L.K., Liew, A.W.C., Yan, H. and Tang, S.S., Gene expression data clustering and visualization based on a binary hierarchical clustering framework, Journal of Visual Languages and Computing 14(4) (2003) pp. 341–362.
8. M. Barni, R. Gualtieri, A new possibilistic clustering algorithm for line detection in real world imagery, Pattern Recognition 32(11) (1999) pp. 1897–1909.
9. Abonyi, J., Babuska, R. and Szeifert, F., Modified Gath-Geva fuzzy clustering for identification of Takagi-Sugeno fuzzy models, IEEE Transactions on Systems, Man and Cybernetics, Part B-Cybernetics 32(5) (2002) pp. 612–621.
10. D. Gustafson, W. Kessel, Fuzzy clustering with fuzzy covariance matrix, Proceedings of the IEEE CDC, San Diego (1979) pp. 761–766.
11. I. Gath, A. Geva, Unsupervised optimal fuzzy clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 7 (1989) pp. 773–781.
12. M. Setnes, Supervised fuzzy clustering for rule extraction, Proceedings of FUZZ-IEEE'99, Seoul, Korea, (1999) pp. 1270–1274.
13. J. J. Sammon, A nonlinear mapping for data structure analysis, IEEE Transactions on Computers 18 (1969) pp. 401–409.