STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **XLIX**, Number 2, 2004

DATA ANALYSIS WITH FUZZY SETS: A SHORT SURVEY

HORIA F. POP

Abstract. This paper develops a short survey of the fuzzy sets theory and how it can contribute to better, more robust data analysis methods. We briefly cover the major fuzzy sets definitions, the rough sets alternative, fuzzy clustering, fuzzy classification, fuzzy regression, data visualisation and projection. We conclude that the fuzzy sets represent a very important advance for intelligent data analysis.

1. Introduction

The fuzzy sets represent a mathematical theory suitable for modelling imprecision and unclearness. Generally, unclearness is associated to the difficulty of making precise statements with respect to a certain topic. On the other side, in the *Fuzzy Sets Theory*, the hard alternative yes - no is indefinitely nuanceable. From this point of view, the fuzzy sets theory is not only a theory dealing with ambiguity; it is also a theory of fuzzy reasoning.

The interpretation of *Fuzzy Logic* is twofold. In a narrow sense, fuzzy logic is a logical system that may be viewed as an extension and a generalization of classical logic. In a wider sense, fuzzy logic is almost synonymous with the theory of fuzzy sets, encompassing the 'strict' fuzzy logic.

The fundamental fact that lies behind fuzzy logic is that any field and any theory may be fuzzified by replacing the concept of crisp set with the concept of fuzzy set. Thus have appeared theoretic fields such as fuzzy arithmetic, fuzzy topology, fuzzy graph theory, fuzzy probability theory, 'strict' fuzzy logic, a.o. Similarily, applied fields that suffered generalizations are fuzzy neural network theory, fuzzy pattern recognition, fuzzy mathematical programming, a.o.

Received by the editors: December 10, 2004.

2000 *Mathematics Subject Classification.* 03E72, 62-07, 62Hxx.

1998 *CR Categories and Descriptors.* I.5.1 [**Computing Methodologies**] : Pattern Recognition – *Models – Fuzzy set*; I.5.2 [**Computing Methodologies**] : Pattern Recognition – *Design methodology – Classifier design and evaluation, Feature evaluation and selection, Pattern analysis*; H.3.1 [**Information Systems**] : Information Storage and Retrieval – *Content analysis and indexing* .

What is gained through fuzzyfication is *greater generality*, *higher expressivity*, an *enhanced ability to model* real-world problems, and a methodology for exploiting the *tolerance for imprecision* [6].

2. Fuzzy sets

Fuzzy sets were introduced as generalization of the classical crisp sets, as a means of representing and manipulating imprecise data. However, not all the properties that are valid for operations on crisp set are valid for fuzzy sets, and the inability to deal with this may result in improper use of fuzzy sets [13].

Professor Lotfi A. Zadeh used the following words to describe the importance of fuzzy sets:

> "The fuzzy set was conceived as a result of an attempt to come to grips with the problem of pattern recognition in the context of imprecisely defined categories. In such cases, the belonging of an object to a class is a matter of degree, as is the question of whether or not a group of objects form a cluster."

We recall here the basic definitions of fuzzy sets.

Let X be a non-empty crisp set. The *fuzzy set* A in X is characterised by its membership function, $A : X \to [0, 1]$, where $A(x)$ is interpreted as the membership degree of element $x \in X$ in the fuzzy set A.

The *universal fuzzy set* in X, denoted by X, is defined by $X(x) = 1, \forall x \in X$.

The *empty fuzzy set* in X, denoted $\emptyset$, is defined by $\emptyset(x) = 0, \forall x \in X$.

The fuzzy sets A and B in X are said *equal*, and denoted $A = B$, if $A(x) = B(x), \forall x \in X$.

The fuzzy set A in X is a *subset* of the fuzzy set B in X, denoted $A \subset B$, if $A(x) \leq B(x), \forall x \in X$.

The *complement* of a fuzzy set A in X, denoted $\bar{A}$ is defined by $\bar{A}(x) = 1 - A(x), \forall x \in X$.

The *intersection* of fuzzy sets A and B in X, denoted $A \cap B$, is the fuzzy set defined as $(A \cap B)(x) = T(A(x), B(x)), \forall x \in X$, where T is a triangular norm (i.e. commutative, associative, non-decreasing in each argument, and $T(a, 1) = a, \forall a \in [0, 1]$).

The *union* of fuzzy sets A and B in X, denoted $A \cup B$, is the fuzzy set defined as $(A \cup B)(x) = S(A(x), B(x)), \forall x \in X$, where S is a triangular conorm (i.e. commutative, associative, non-decreasing in each argument, and $S(a, 0) = a, \forall a \in [0, 1]$).

For any triangular norm $T : [0, 1] \times [0, 1] \to [0, 1]$, the dual triangular conorm $S : [0, 1] \times [0, 1] \to [0, 1]$, defined by $S(x, y) = 1 - T(1 - x, 1 - y)$ is, itself, a triangular conorm.

In what follows we will analyse some of the properties of crisp sets from the point of view of applicability to fuzzy sets. We will exemplify with two widely-used t-norms and t-conorms:

Standard: $T_S(a,b) = \min\{a,b\}$ and $S_S(a,b) = \max\{a,b\}$
Lukasiewicz: $T_L(a,b) = \max\{a+b-1, 0\}$ and $S_L(a,b) = \min\{a+b, 1\}$

	Property	Crisp sets	T_S and S_S	T_L and S_L
(1)	Idempotence laws	Valid	Valid	Invalid
(2)	$A \cup \bar{A} = X$ and $A \cap \bar{A} = \emptyset$	Valid	Invalid	Valid
(3)	Distributivity laws	Valid	Valid	Invalid
(4)	De Morgan laws	Valid	Valid	Valid

Properties (1) are satisfied only by the standard t-norm and t-conorm. Properties (4) are satisfied by any t-norm and the dual t-conorm. Any t-norm and t-conorm, defined on nondegenerate fuzzy sets, that satisfy properties (2), do not satisfy properties (3). For any t-norm and t-conorm that verify the properties (2) and (3), the fuzzy sets have only crisp values, i.e. they reduce to crisp sets.

The properties above make the standard definition suitable for use in 'strict' fuzzy logic, and the Lukasiewicz definition suitable for use in fuzzy clustering.

3. Rough sets

A different generalisation of the concept of crisp set is the *rough set*. But while a fuzzy set comes as a membership function, the idea behind rough sets is to approximate sets using collections of sets [8].

Let us consider a collection of sets $C = \{C_1, C_2, \ldots\}$, and a set D. We define the *lower approximation of D by C*, denoted D_L, the set

$$D_L = \cup C_i \text{ such that } C_i \cap D = C_i.$$

We define the *upper approximation of D by C*, denoted D^U, the set

$$D^U = \cup C_i \text{ such that } C_i \cap D \neq \emptyset.$$

We define the *boundary of D by C*, denoted $D_L^U = D^U - D_L$.

A set D is said to be *rough* if it has a non-empty boundary when approximated by C. Otherwise, the set D is called *crisp*.

Fuzzy sets model the inherent vagueness in the data. As a difference, rough sets model ambiguity due to lack of information.

4. Fuzzy clustering

The subject of *Cluster Analysis* is the classification of objects into categories. Since most categories we use have vague boundaries, and may even overlap, the necessity of introducing fuzzy sets is obvious. The main issues approached by research in Fuzzy Clustering refer to the following [10]:

Data representation: Input data is obtained by measurements on the objects that are to be recognized. Each object is represented as a vector of measured values $x = (x_1, x_2, \ldots, x_s)$, where x_i is a particular characteristic of the object.

Feature extraction: Due to the large number of characteristics, there is a need to extract the most relevant characteristics from the input data, so that the amount of information lost in this way is minimal, and the classification realised with the projected data set is relevant with respect to the original data. In order to achieve this feature extraction, different statistical techniques, as well as the fuzzy clustering algorithms outlined here, may be used.

Clusters shape: Pattern recognition techniques based on fuzzy objective functions minimizations use objective functions which are particular to different clusters shapes. Ways to approach the problem of correctly identifying the clusters shape are the use of adaptive distances in a second run to change the shapes of the produced clusters so that all are unit spheres, and adaptive algorithms which dynamically change the local metrics during the iterative procedure in the original run, without the need of a second run.

Cluster validity: Another problem of such algorithms is that of determining the optimal number of classes that correspond to the cluster substructure of the data set. There are two approaches: the use of validity functionals which is a post-factum method, and the use of hierarchical algorithms, which produce not only the optimal number of classes (based on the needed granularity), but also a binary hierarchy that show the existing relationships between the classes.

Defuzzification of final fuzzy partition: Since humans need for their analysis crisp partitions, such procedures should be able to produce, together with the final fuzzy partition, a crisp version thereof. There are a number of techniques, which differ on their ability to produce a non-degenerate crisp partition (i.e. a crisp partition with all the member crisp sets non-empty), and on their ability to produce the crisp partition closest to the original fuzzy partition.

Method: Essentially, the method is based on defining a dissimilarity function between the data set and the prototypes (not necessarily vectors in the same space) of the fuzzy classes. A fuzzy objective function is defined based on this dissimilarity function. In order to minimize the fuzzy objective function, a two step iterative procedure is used: for certain prototypes, the optimal fuzzy partition is determined; reciprocally, for a certain fuzzy partition, the optimal prototypes are determined. This procedure continually decreases the value of the objective function.

In what follows we will recall the Fuzzy Clustering generic algorithm [1].

Let us consider a set of data items $X = \{x^1, \ldots, x^n\} \subset \mathbb{R}^s$, characterised in such a way that we may define a measure of their (dis)similarity. Our aim is to find a fuzzy partition $P = \{A_1, \ldots, A_c\}$ that best represents the cluster substructure of the data set X., i.e. data items of the same class should be as similar as possible, and data items of different classes should be as dissimilar as possible.

Let us suppose that the fuzzy sets A_i are represented by some prototypes L_i, and that we can define a function $D(x^j, L_i)$ that represents the dissimilarity between a certain data item x^j and the prototype L_i.

At this point, we do not specify the exact shape of the prototypes L_i. Of course, when the prototypes L_i are fully described, the dissimilarity $D(x^j, L_i)$ will need to be fully described, as well.

We define the representation inadequacy of the fuzzy partition P by the set of prototypes L as the function

$$J(P, L) = \sum_{i=1}^{c} \sum_{j=1}^{n} A_i(x^j)^m \cdot D(x^j, L_i)$$

where $m > 1$ is a constant defining the weight the fuzziness is taken into account with.

Let us observe that J is an objective function of the tipe of square errors sum. Our problem is to determine the fuzzy partition P and its prototype representation L that minimizes the function J. Because an algorithm to obtain an exact solution to this problem is not known, we will use an approximate method in order to determine a local solution. The minimum problem will be solved by using an iterative method where J is successively minimized with respect to P and L.

The *Fuzzy Clustering generic algorithm* is, thus, the following:

(1) Given: c, n, m, and $x^j, j = 1, \ldots, n$; $l = 0$;
(2) Initialize fuzzy partition $P^{(0)} = \{A_1, \ldots, A_c\}$;
(3) Compute prototypes L_i that minimize $J(P^{(l)}, \cdot)$; this depends on the definition of D, and may be costly;
(4) Compute fuzzy partition $P^{(l+1)}$ that minimizes $J(\cdot, L)$:

$$A_i^{(l+1)}(x^j) = \frac{1}{\sum_{k=1}^{c} \left(\frac{D(x^j, L_i)}{D(x^j, L_k)} \right)^{\frac{1}{m-1}}}$$

(5) Compare fuzzy partitions $P^{(l+1)}$ with $P^{(l)}$. If close enough, then STOP, else increase l by 1 and GOTO STEP 3.

4.1. **Fuzzy clustering variants and improvements.** The first algorithm from this class, developed by Dunn [3], was the *Fuzzy c-Means* algorithm and used spherical prototypes. For an up-to-date discussion of the different variants and improvements of the Fuzzy c-Means algorithm, see [5].

Geometric prototypes:

- Fuzzy c-Means – Dunn (1974), Bezdek (1974)
- Fuzzy c-Varieties and Fuzzy c-Elliptotypes – Bezdek et.al. (1981)
- Fuzzy c-Ellipsoids – Lenart (1989)
- Adaptive Fuzzy Clustering – Dave (1989), Dumitrescu, Pop (1990)
- Use of fuzzy covariance matrix – Gustaffson, Kessel (1979)
- L_p Fuzzy c-Means – Miyamoto, Agusta (1998), Hathaway, Bezdek (2000)

Empty shell prototypes:

- Fuzzy c-Shells – Dave (1990), Krishnapuram, Nasraoui, Frigui (1992)
- Adaptive Fuzzy c-Shells – Dave, Bhaswan (1992)
- Fuzzy c-Ellipsoidal Shells – Frigui, Krishnapuram (1996), Gath, Hoory (1995)
- Fuzzy c-Quadric Shells – Krishnapuram, Frigui, Nasraoui (1993, 1995)
- Fuzzy c-Rectangular Shells – Höppner, Klawonn, Kruse (1997)

Fuzzy clustering with incomplete data:

- Unsupervised fuzzy competitive learning – Chung, Lee (1994)
- Whole data strategy – Hathaway, Bezdek (2001)
- Partial distance strategy – Hathaway, Bezdek (2001)
- Optimal completion strategy – Hathaway, Bezdek (2001)
- Nearest neighbour strategy – Hathaway, Bezdek (2001)

Other methods and models:

- Fuzzy divisive hierarchic clustering – Dumitrescu (1988)
- Cross-clustering – Dumitrescu, Pop (1995)
- Noise Clustering – Dave (1991), Dave, Sen (1997)
- Possibilistic, Probabilistic – Krishnapuram, Keller (1993), Gath, Geva (1989)

5. Fuzzy classification

Let us consider a set of objects, $X = \{x^1, \ldots, x^p\} \in \mathbb{R}^d$, classified with a fuzzy clustering algorithm of the type Fuzzy n-Means, and the fuzzy partition $P = \{A_1, \ldots, A_n\}$ coresponding to the cluster substructure of the set X (see [4]).

We rise the problem of including an extra-object $x^0 \notin X$ in the cluster structure of X. Of course, this would mean to determine the membership degrees of x^0 to the fuzzy sets members of the partition P. These degrees will provide sufficient information in order to classify the object x^0 with respect to the elements of X.

The algorithms that solve this kind of problems are called *fuzzy decision supervized classification algorithms* [9]. Supervised classification because the classification of the extra-object is realized using not only the data set X, but also the fuzzy partition obtained by classifying the set X. Fuzzy decision because, unlike the traditional classifiers, where the aim is to state in which classical subset the

object may be included, now we are interested in the membership degrees of the object to the fuzzy sets members in the given fuzzy partition.

This category of classifiers has to be studied in comparison with the other two important classes. On one hand, the *classical classifiers*, where the supervised information is a crisp partition and the final descision is, as well, crisp (i.e. the 'to be or not to be' kind of question). On the other hand, we have the traditional fuzzy generalisations of the crisp classifiers, the so-called *fuzzy classifiers*, where the supervised information is a fuzzy partition, but the final decision is, still, crisp.

The simplest approach for the fuzzy decision supervised classifier is the classification of the extended set $X \cup \{x^0\}$ using one of the common fuzzy clustering algorithms, and the comparative analysis of the produced partition to the partition P. Unfortunately, this method is very costly considering the neccessary execution time, because it supposes the classification of the objects of X, set that in the real applications may be quite large. Also, this is not supervised classification, because the information provided by the fuzzy partition P is not used.

The alternate approach is to keep unchanged the membership degrees of the objects in X to the sets of the fuzzy partiton P, and to determine the membership degreess of x^0 as a consequence of the minimization of an objective function similar to those used for the algorithms of the Fuzzy n-Means type.

There are, as well, a few more straightforward algorithms of this type. These algorithms are fuzzy generalizations of the well-known *k nearest neighbours* and *nearest prototype*.

6. Fuzzy regression

The aim of regression techniques is to relate, corelate or model a measure response based on the value of a given variable.

The common approach has been to work with traditional schemes as, for example, the linear Least-Square method. But these methods come with important drawbacks: they assume that data is homoscedastic – y-direction error is independent of the controlled variable.

Unfortunately, most real-world data are heteroscedastic, and presence of outliers is, as well, common. In such cases, use of robust methods is desired. But most of current robust methods do not provide best results in all situations.

We aim at developing a class of fuzzy regression methods that overcome these negative issues.

Fuzzy clustering techniques are suitable for determining the optimal cluster substructure of a data set, and they suppose that such a substructure does exist. The problem at hand is, however, to be able to determine the one fuzzy set A and its prototype L that best describes the data set. In such a case, a regular fuzzy clustering algorithm will not work.

The fuzzy set that best corresponds to a data set, based on a prototype characterisation of the data, is a useful notion in the search for robust regression techniques, as well as for developing data analysis techniques where the data items are considered according to their goodness of fit (i.e. their membership degree to this fuzzy set).

We consider a binary fuzzy partition, $\{A, \bar{A}\}$, where $\bar{A}$ is a virtual class with a hypothetical prototype, characterized by the constant dissimilarity

$$D(x^j, \bar{L}) = \delta := \left(\frac{\alpha}{1-\alpha}\right)^{m-1}.$$

The optimal fuzzy set A, as defined by our problem, is determined by minimizing the following fuzzy objective function:

$$J(A, L) = \sum_{j=1}^{n} A(x^j)^m D(x^j, L) + \sum_{j=1}^{n} \bar{A}(x^j)^m \left(\frac{\alpha}{1-\alpha}\right)^{m-1}, \alpha \in (0, 1).$$

The algorithm used to solve this problem has been called the *Fuzzy Regression* generic algorithm [11, 12]:

(1) Given α; Initialize $A^{(0)}(x) = 1$, $l = 0$;
(2) Compute prototype L that minimizes $J(A^{(l)}, \cdot)$;
(3) Compute fuzzy set $A^{(l+1)}$ that minimizes $J(\cdot, L)$:

$$A^{(l+1)}(x^j) = \frac{\dfrac{\alpha}{1-\alpha}}{\dfrac{\alpha}{1-\alpha} + D(x^j, L)^{\frac{1}{m-1}}}$$

(4) Compare fuzzy sets $A^{(l+1)}$ with $A^{(l)}$. If close enough, then STOP, else increase l by 1 and GOTO STEP 2.

As an improvement, in order to assure the independence of scale, in the equation from STEP 3, we will replace the dissimilarity $D(x^j, L)$ with the relative dissimilarity

$$D_r(x^j, L) = D(x^j, L) / \max_{j=1,n} D(x^j, L).$$

This is equivalent to setting δ initially to

$$D(x^j, \bar{L}) = \delta := \left(\frac{\alpha}{1-\alpha}\right)^{m-1} \max_{j=1,n} D(x^j, L).$$

6.1. **Properties of the Fuzzy Regression algorithm.** Let us suppose that X is a data set, and A and L are the optimal fuzzy set and its prototype representation, respectively. The following properties are valid [11].

i. *(Maximal membership degree)* $A(x) = 1 \Leftrightarrow D(x, L) = 0$
ii. *(Minimal membership degree)* $A(x) = \alpha \Leftrightarrow D_r(x, L) = 1$
iii. *(Membership degree interval)* $A(x) \in [\alpha, 1]$ for all $x \in X$

iv. *(Empty fuzzy set)* $\alpha = 0 \Leftrightarrow A(x) = 0$ for all $x \in X$
v. *(Degenerate fuzzy set)* $\alpha = 1 \Leftrightarrow A(x) = 1$ for all $x \in X$
vi. *(Strict monotony)* $A(x) < A(y) \Leftrightarrow D(x, L) < D(y, L)$
vii. *(Equality)* $A(x) = A(y) \Leftrightarrow D(x, L) = D(y, L)$

The constant α is an input parameter that has the role of setting the polarization of the fuzzy partition $\{A, \bar{A}\}$. The best results appear to be obtained with $\alpha \approx 0.10$.

Being based on the Fuzzy c-Lines algorithm, the linear version of this algorithm (Fuzzy Linear Regression) uses as dissimilarity the square distance to the line, as opposed to the y-distance, used by the classical Least Squares algorithm.

Due to the use of fuzzy sets, the algorithm is efficient in all testing conditions, and is better than most methods it has been compared with [11].

The algorithm allows the detection of the type of data sets, i.e. homoscedasticity, heteroscedasticity, presence of outliers, or any combination thereof, with or without any other irregularities. This is done through repeated runs by analysing the graphical representation of the surface made by the coefficients vectors of the linear prototypes, as determined for α varied continuously in the interval $(0, 1)$. In the case of a two-dimensional data set, the curve is defined through the points (a_0, a_1), where $y = a_0 + a_1 x$ is the linear prototype of the data set defined.

7. Data projection and selection

7.1. **Visualisation of high-dimensional data items.** The simplest method to visualise a data set is to plot profiles: two-dimensional graphs where the dimensions are enumerated on the x axis, and the corresponding values on y. An alternative, also largely used, is to plot two-dimensional representations of pairs of two original dimensions.

There are, also, methods that produce different curves based on the data items values. The most important drawback of such methods is that they do not reduce the amount of data, and thus they cannot be used effectively with large high-dimensional data sets. However, they can be used for illustrating data summaries.

7.2. **Projection methods.** The main feature of clustering algorithms is that they reduce the amount of data items by grouping them in classes. The projection methods described below can be used for reducing the data dimensionality. The goal of these techniques is to represent the data set in a lower-dimensional space in such a way that certain properties of the data set are preserved as much as possible.

The following questions are important when discussing a method for large, high-dimensional data sets: what kind of structure the method extracts from the data set; how does it illustrate the structure; does it reduce dimensionality of data; does it reduce the number of data items.

7.3. **Principal Components Analysis with Fuzzy sets.** The aim of PCA is to achieve data dimensionality reduction by determining new, fewer variables. The new variables, called principal components, correspond to the axes of maximal elongation of data These principal components are linear combinations of the original variables.

It has been remarked that the number of principal components necessary to conserve 90% of data variance is considerably less than the size of data space. As such, it has become extremely important to determine the relevant variables. We could thus achieve not only a significant data projection, but as well a variable selection.

The use of fuzzy sets enables us to aproach the problem of isolated points with respect both to the data set and to the principal directions [2].

The PCA Algorithm. The PCA algorithm starts by computing the covariance or corelation matrix

$$\mathrm{Cov}_{ij} = \frac{1}{n-1}\sum_{k=1}^{n}(x_i^k - \bar{x}_i)\cdot(x_j^k - \bar{x}_j)$$

$$\mathrm{Cor}_{ij} = \frac{\mathrm{Cov}_{ij}}{s_i \cdot s_j}, s_i = \frac{1}{n-1}\sum_{k=1}^{n}(x_i^k - \bar{x}_i)^2$$

Then we compute the eigenvectors and eigenvalues of this matrix; these are the principal components and the scatter values. Based on these scatter values, select the necessary number of principal components.

Then we determine the values of data for the new variables (i.e. project the data set in the space of the selected principal components): $X'^T = X^T \cdot E$.

This is a very simple, but very effective algorithm indeed. Essentially, we achieve a clearer image about the data by only rotating the data space such that the new axes of coordinates coincide with the directions of maximal elongation of data.

Fuzzy PCA, first component. The major problem of the PCA algorithm rests, as always, with the isolated points. As a first possible way to handle this, we will take into account the points isolated with respect to the first principal component only.

We aim to introduce fuzzy membership degrees according to the distance to the first principal component. As such, we will use a scheme similar to the fuzzy regression algorithm, to determine the first fuzzy principal component and the corresponding fuzzy membership degrees.

We will thus replace the traditional covariance matrix by the fuzzy covariance matrix, given by

$$C_{ij} = \frac{\sum_{k=1}^{n} A(x^k)^m \cdot (x_i^k - \bar{x}_i)\cdot(x_j^k - \bar{x}_j)}{\sum_{k=1}^{n} A(x^k)^m},\ i,j = 1,\ldots,p.$$

The major advantage is that the first principal component will count the merits of each data item; as such, will consider the isolated points with less significance.

Fuzzy PCA, orthogonal. The Fuzzy PCA algorithm discussed in the preceding section fuzzifies only the first component. In order to get a most effective method, we have to deal with the problem of fuzzifying all the components.

The main idea is to use a different approach: by projecting the data in smaller-sized spaces. After the first fuzzy eigenvector is determined, all data is projected to the hyperplane rectangular on it. The eigenvectors corresponding to the projected data will be orthogonal to the eigenvector determined above. As such, the second largest eigenvector of the original data will correspond to the largest eigenvector of the projected data. The projection continues further on, etc.; finally, the eigenvectors are rebuilt in the original space.

This method Advantage: the compotation of the other fuzzy components is reduced to the computation of the first fuzzy component of a smaller-sized matrix

7.4. **Multi-Dimensional Scaling.** Alternate projection methods aim to reduce the data dimensionality is to optimize the representation in the lower-dimension space so that the distances between points in the projected space are as similar as possible to the distances between the corresponding points in the original space [7].

We will present here a class of methods known as *multidimensional scaling (MDS)*. The aim of these methods is to project data from a pseudo-metric space (i.e. characterised by a dissimilarity measure) onto a metric space. Such metods are especially useful for preprocessing non-metric data in order to use them with algorithms only valid with metric input.

The first MDS method is the *metric MDS*, characterized by minimizing the squared error cost function:

$$E_M = \sum_{k \neq l} \left(d(k,l) - d'(k,l) \right)^2 ,$$

where, for the original items x_k and x_l, $d(k,l)$ is their dissimilarity, and $d'(k,l)$ is the distance between the corresponding vectors from the projected metric space.

If the components of the data vectors are expressed on an ordinal scale, a perfect reproduction of the Euclidean distances may not be the best goal. In such a situation, only the rank order of the distances between the vectors is meaningful. The error function is defined as

$$E_N = \frac{\sum_{k \neq l} \left(f(d(k,l)) - d'(k,l) \right)^2}{\sum_{k \neq l} \left(d'(k,l) \right)^2},$$

where f is a monotonically increasing function that acts on the original distances and always maps the distances to such values that best preserve the rank order.

Another non-linear mapping method, the *Sammon's mapping*, is closely related to the metric MDS. The only difference is that the errors in distance preservation are normalized with the distance in the original space. Thus, preservation of small distances is emphasized. The error function is defined as

$$E_S = \sum_{k \neq l} \frac{(d(k,l) - d'(k,l))^2}{d(k,l)}.$$

8. Conclusions

This paper surveyed the different intelligent data analysis aspects, with an insight from the fuzzy sets perspective. The fuzzy sets, as a natural generalisation of the classical crisp sets, bring more power and refinement to data analysis. A lot of work has been done in developping better, more robust algorithms for data analysis. The different fuzzy data analysis algorithms are not only an issue for theoretical study, but a series of effective tools used in most diverse applications, from medicine to chemistry, physics, biology and engineering.

References

[1] James C. Bezdek, *Pattern Recongnition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981
[2] Thomas R. Cundari, Costel Sârbu, Horia F. Pop, Robust Fuzzy Principal Component Analysis (FPCA). A comparative study concerning interaction of carbon-hydrogen bonds with molybdenum-oxo bonds. *J. Chem. Inf. Comput. Sci.*, 42, 6, 2002, 1363–1369
[3] J.C. Dunn, A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters, *Journal of Cybernetics* 3, 1974, 32–57
[4] D. Dumitrescu, Hierarchical pattern classification, *Fuzzy Sets Syst.*, 28, 1988, 145–162
[5] F. Höppner, F. Klawonn, R. Kruse, T. Runkler, *Fuzzy Cluster Analysis*, Wiley, New York, 1999
[6] George J. Klir, Bo Yuan, *Fuzzy Sets and Fuzzy Logic*, Prentice Hall, New Jersey, 1995
[7] J. B. Kruskal, M. Wish, *Multidimensional Scaling*, Sage Publications, Beverly Hills, CA, 1977
[8] Zdzislaw Pawlak, Rough sets, *Int. J. Comput. Inform. Sci.*, 11, 1982, 341–356
[9] Horia F. Pop, Fuzzy decision supervised classifiers, *Studia Universitatis Babes-Bolyai, Series Informatica*, 40, 3, 1995, 89–100
[10] Horia F. Pop, *Sisteme inteligente în probleme de clasificare*, Editura Mediamira, Cluj, 2004
[11] Horia F. Pop, Costel Sârbu, A New Fuzzy Regression Algorithm, *Anal. Chem.* 68, 1996, 771–778
[12] Costel Sârbu, Horia F. Pop, Fuzzy robust estimation of central location, *Talanta*, 54, 2001, 125–130
[13] Lotfi A. Zadeh, Fuzzy sets, *Inf. Control*, 8, 1965, 338–353

Department of Computer Science, Babeş-Bolyai University, 1 M. Kogălniceanu St., RO-400084 Cluj-Napoca, Romania
E-mail address: hfpop@cs.ubbcluj.ro