On High Dimensional Searching Spaces and Learning Methods

Hossein Yazdani, Daniel Ortiz-Arroyo, Kazimierz Choroś and Halina Kwasnicka

Abstract In data science, there are important parameters that affect the accuracy of the algorithms used. Some of these parameters are: the type of data objects, the membership assignments, and distance or similarity functions. In this chapter we describe different data types, membership functions, and similarity functions and discuss the pros and cons of using each of them. Conventional similarity functions evaluate objects in the vector space. Contrarily, Weighted Feature Distance (WFD) functions compare data objects in both feature and vector spaces, preventing the system from being affected by some dominant features. Traditional membership functions assign membership values to data objects but impose some restrictions. Bounded Fuzzy Possibilistic Method (BFPM) makes possible for data objects to participate fully or partially in several clusters or even in all clusters. BFPM introduces intervals for the upper and lower boundaries for data objects with respect to each cluster. BFPM facilitates algorithms to converge and also inherits the abilities of conventional fuzzy and possibilistic methods. In Big Data applications knowing the exact type of data objects and selecting the most accurate similarity [1] and membership assignments is crucial in decreasing computing costs and obtaining the best performance. This chapter provides data types taxonomies to assist data miners in selecting the right

H. Yazdani (✉) · D. Ortiz-Arroyo
Department of Energy Technology, Aalborg University Esbjerg, Esbjerg, Denmark
e-mail: yazdanihossein@yahoo.com

D. Ortiz-Arroyo
e-mail: doa@et.aau.dk

H. Yazdani
Faculty of Electronics, Wroclaw University
of Science and Technology, Wroclaw, Poland

H. Yazdani · K. Choroś
Faculty of Computer Science and Management, Wroclaw University
of Science and Technology, Wroclaw, Poland
e-mail: kazimierz.choros@pwr.edu.pl

H. Yazdani · H. Kwasnicka
Department of Computational Intelligence, Wroclaw University
of Science and Technology, Wroclaw, Poland
e-mail: halina.kwasnicka@pwr.wroc.pl

W. Pedrycz and S.-M. Chen (eds.), *Data Science and Big Data: An Environment of Computational Intelligence*, Studies in Big Data 24,
DOI 10.1007/978-3-319-53474-9_2

learning method on each selected data set. Examples illustrate how to evaluate the accuracy and performance of the proposed algorithms. Experimental results show why these parameters are important.

Keywords Bounded fuzzy-possibilistic method · Membership function · Distance function · Supervised learning · Unsupervised learning · Clustering · Data type · Critical objects · Outstanding objects · Weighted feature distance

1 Introduction

The growth of data in recent years has created the need for the use of more sophisticated algorithms in data science. Most of these algorithms make use of well known techniques such as sampling, data condensation, density-based approaches, grid-based approaches, divide and conquer, incremental learning, and distributed computing to process big data [2, 3]. In spite of the availability of new frameworks for Big Data such as Spark or Hadoop, working with large amounts of data is still a challenge that requires new approaches.

1.1 Classification and Clustering

Classification is a form of supervised learning that is performed in a two-step process [4, 5]. In the training step, a classifier is built from a training data set with class labels. In the second step, the classifier is used to classify the rest of the data objects in the testing data set.

Clustering is a form of unsupervised learning that splits data into different groups or clusters by calculating the similarity between the objects contained in a data set [6–8]. More formally, assume that we have a set of n objects represented by $O = \{o_1, o_2, \ldots, o_n\}$ in which each object is typically described by numerical *feature − vector* data that has the form $X = \{x_1, \ldots, x_m\} \subset R^d$, where d is the dimension of the search space or the number of features. In classification, the data set is divided into two parts: learning set $O_L = \{o_1, o_2, \ldots, o_l\}$ and testing set $O_T = \{o_{l+1}, o_{l+2}, \ldots, o_n\}$. In these kinds of problems, classes are classified based on a class label x_l. A cluster or a class is a set of c values $\{u_{ij}\}$, where u represents a membership value, i is the ith object in the data set and j is the jth class. A partition matrix is often represented as a $c \times n$ matrix $U = [u_{ij}]$ [6, 7]. The procedure for membership assignment in classification and clustering problems is very similar [9], and for convenience in the rest of the paper we will refer only to clustering.

The rest of the chapter is organized as follow. Section 2 describes the conventional membership functions. The issues with learning methods in membership assignments are discussed in this section. Similarity functions and the challenges on conventional distance functions are described in Sect. 3. Data types and their behaviour

are analysed in Sect. 4. Outstanding and critical objects and areas are discussed in this section. Experimental results on several data sets are presented in Sect. 5. Discussion and conclusion are presented in Sect. 6.

2 Membership Function

A partition or membership matrix is often represented as a $c \times n$ matrix $U = [u_{ij}]$, where u represents a membership value, i is the ith object in the data set and j is the jth class. Crisp, fuzzy or probability, possibilistic, bounded fuzzy possibilistic are different types of partitioning methods [6, 10–15]. Crisp clusters are non-empty, mutually-disjoint subsets of O:

$$M_{hcn} = \left\{ U \in \Re^{c \times n} | \; u_{ij} \in \{0, 1\}, \quad \forall j, i; \right.$$

$$\left. 0 < \sum_{i=1}^{n} u_{ij} < n, \quad \forall j; \quad \sum_{j=1}^{c} u_{ij} = 1, \quad \forall i \right\} \tag{1}$$

where u_{ij} is the membership of the object o_i in cluster j. If the object o_i is a member of cluster j, then $u_{ij} = 1$; otherwise, $u_{ij} = 0$. Fuzzy clustering is similar to crisp clustering, but each object can have partial membership in more than one cluster [16–20]. This condition is stated in (2), where data objects may have partial nonzero membership in several clusters, but only full membership in one cluster.

$$M_{fcn} = \left\{ U \in \Re^{c \times n} | \; u_{ij} \in [0, 1], \quad \forall j, i; \right.$$

$$\left. 0 < \sum_{i=1}^{n} u_{ij} < n, \quad \forall j; \quad \sum_{j=1}^{c} u_{ij} = 1, \quad \forall i \right\} \tag{2}$$

An alternative partitioning approach is *possibilistic clustering* [8, 18, 21]. In (3) the condition $\sum_{j=1}^{c} u_{ij} = 1$ is relaxed by substituting it with $\sum_{j=1}^{c} u_{ij} > 0$.

$$M_{pcn} = \left\{ U \in \Re^{c \times n} | \; u_{ij} \in [0, 1], \quad \forall j, i; \right.$$

$$\left. 0 < \sum_{i=1}^{n} u_{ij} < n, \; \forall j; \quad \sum_{j=1}^{c} u_{ij} > 0, \quad \forall i \right\} \tag{3}$$

Based on (1), (2) and (3), it is easy to see that all crisp partitions are subsets of fuzzy partitions, and a fuzzy partition is a subset of a possibilistic partition, i.e., $M_{hcn} \subset M_{fcn} \subset M_{pcn}$ [8].

2.1 Challenges on Learning Methods

Regarding the membership functions presented above we look at the pros and cons of using each of these functions. In crisp memberships, if the object o_i is a member of cluster j, then $u_{ij} = 1$; otherwise, $u_{ij} = 0$. In such a membership function, members are not able to participate in other clusters and therefore it cannot be used in some applications such as in applying hierarchical algorithms [22]. In fuzzy methods (2), each column of the partition matrix must sum to 1 ($\sum_{j=1}^{c} u_{ij} = 1$) [6]. Thus, a property of fuzzy clustering is that, as c becomes larger, the u_{ij} values must become smaller.

Possibilistic methods have also some drawbacks such as offering trivial null solutions [8, 23] and lack of upper and lower boundaries with respect to each cluster [24]. Possibilistic methods do not have this constraint that fuzzy method have, but fuzzy methods are restricted by the constraint ($\sum_{j=1}^{c} u_{ij} = 1$).

2.2 Bounded Fuzzy Possibilistic Method (BFPM)

Bounded Fuzzy Possibilistic Method (BFPM) makes it possible for data objects to have full membership in several or even in all clusters. This method also does not have the drawbacks of fuzzy and possibilistic clustering methods. BFPM in (4), has the normalizing condition $1/c \sum_{j=1}^{c} u_{ij}$. Unlike Possibilistic method ($u_{ij} > 0$) there is no boundary in the membership functions. BFPM employs defined intervals $[0, 1]$ for each data object with respect to each cluster. Another advantage of BFPM is that its implementation is relatively easy and that it tends to converge quickly.

$$M_{bfpm} = \Big\{ U \in \Re^{c\times n} |\, u_{ij} \in [0,1], \quad \forall j, i;$$

$$0 < \sum_{i=1}^{n} u_{ij} < n, \quad \forall j; \quad 0 < 1/c \sum_{j=1}^{c} u_{ij} \leq 1, \quad \forall i \Big\} \tag{4}$$

BFPM avoids the problem of decreasing the membership degrees of objects, as the number of clusters increases [25, 26].

2.3 Numerical Example

Assume $U = \{u_{ij}(x) | x_i \in L_j\}$ is a function that assigns a membership degree for each point x_i to a line L_j, where a line represents a cluster. Now consider the following equation which describes n lines crossing at the origin:

$$AX = 0 \tag{5}$$

where matrix A is a $n \times m$ coefficient matrix, and X is an $m \times 1$ matrix, in which n is the number of lines and m is the number of dimensions. From a geometrical point of view, each line containing the origin is a subspace of R^m. Equation (5) describes n with its different lines as a subspace. Without the origin, each of those lines is not a subspace, since the definition of a subspace comprises the existence of the null vector as a condition, in addition to other properties [27].

When trying to design a probability/fuzzy-based clustering method that could create clusters using all the points in all lines, it should be noted that removing or even decreasing the membership value of the origin ruins the subspace. For instance, $x = 0$, $y = 0$, $x = y$, and $x = -y$ are equations representing some of those lines with some data objects (points) on them as shown in the following equation. Note that all lines contain point (0, 0).

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & -1 \end{bmatrix} \times \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Assume that we have two of those lines $L_1 : \{y = 0\}$ and $L_2 : \{x = 0\}$ with five points on each, including the origin, as shown in the following definitions:

$$L_1 = \{p_{11}, p_{12}, p_{13}, p_{14}, p_{15}\} = \{(-1, 0), (-2, 0), (0, 0), (1, 0), (2, 0)\}$$

$$L_2 = \{p_{21}, p_{22}, p_{23}, p_{24}, p_{25}\} = \{(0, -1), (0, -2), (0, 0), (0, 1), (0, 2)\}$$

where $p_{ij} = (x, y)$. As mentioned, the origin is part of all lines, but for convenience, we have given it different names such as p_{13} and p_{23} in each line above.

The point distances with respect to each line and Euclidean $||X||_2$ norm $\left(d_k(x, y) = \left(\sum_{i=1}^{d} | x_i - y_i |^2\right)^{(1/2)}\right)$ are shown in the (2×5) matrices below, where 2 is the number of clusters and 5 is the number of objects.

$$D_1 = \begin{bmatrix} 0.0 , 0.0 , 0.0 , 0.0 , 0.0 \\ 2.0 , 1.0 , 0.0 , 1.0 , 2.0 \end{bmatrix} \qquad D_2 = \begin{bmatrix} 2.0 , 1.0 , 0.0 , 1.0 , 2.0 \\ 0.0 , 0.0 , 0.0 , 0.0 , 0.0 \end{bmatrix}$$

A zero value in the first matrix in the first row indicates that the object is on the first line. For example in D_1, the first row shows that all the members of set X_1 are on the first line. The second row shows how far each one of the points on the line are from the second cluster. Likewise the matrix $D2$ shows the data points on the second line. We assigned membership values to each point, using crisp and fuzzy logic as shown in the matrices below by using the following membership function (6) for crisp and fuzzy methods and also the conditions for these methods described in (1) and (2).

$$U_{ij} = \begin{cases} 1 & \text{if } d_{p_{ij}} = 0 \\ 1 - \frac{d_{p_{ij}}}{d_\delta} & \text{if } 0 < d_{p_{ij}} \leqslant d_\delta \\ 0 & \text{if } d_{p_{ij}} > d_\delta \end{cases} \tag{6}$$

where $d_{p_{ij}}$ is the Euclidean distance of object x_i from cluster j, and d_δ is a constant that we use to normalize the values. In our example we used $d_\delta = 2$.

$$U_{crisp}(L_1) = \begin{bmatrix} 1.0 , 1.0 , \mathbf{1.0} , 1.0 , 1.0 \\ 0.0 , 0.0 , \mathbf{0.0} , 0.0 , 0.0 \end{bmatrix} U_{crisp}(L_2) = \begin{bmatrix} 0.0 , 0.0 , \mathbf{0.0} , 0.0 , 0.0 \\ 1.0 , 1.0 , \mathbf{0.0} , 1.0 , 1.0 \end{bmatrix}$$

or

$$U_{crisp}(L_1) = \begin{bmatrix} 1.0 , 1.0 , \mathbf{0.0} , 1.0 , 1.0 \\ 0.0 , 0.0 , \mathbf{0.0} , 0.0 , 0.0 \end{bmatrix} U_{crisp}(L_2) = \begin{bmatrix} 0.0 , 0.0 , \mathbf{0.0} , 0.0 , 0.0 \\ 1.0 , 1.0 , \mathbf{1.0} , 1.0 , 1.0 \end{bmatrix}$$

$$U_{Fuzzy}(L_1) = \begin{bmatrix} 1.0 , 0.5 , \mathbf{0.5} , 0.5 , 1.0 \\ 0.0 , 0.5 , \mathbf{0.5} , 0.5 , 0.0 \end{bmatrix} U_{Fuzzy}(L_2) = \begin{bmatrix} 0.0 , 0.5 , \mathbf{0.5} , 0.5 , 0.0 \\ 1.0 , 0.5 , \mathbf{0.5} , 0.5 , 1.0 \end{bmatrix}$$

In crisp methods, the origin can be a member of just one line or cluster. Therefore, the other lines without the origin can not be subspaces [27]. In other words, the example "crossing lines at origin" can not be represented by crisp methods.

Given the properties of the membership functions in fuzzy methods, if the number of clusters increases, the membership value assigned to each object will decrease proportionally.

Methods such as PCM, allow data objects to obtain larger values in membership assignments [8, 21]. But PCM needs a good initialization to perform clustering [23]. According to PCM condition ($u_{ij} \geq 0$), the trivial null solutions should be handled by modifying the membership assignments [8, 21, 23]. The authors in [8] did not change the membership function to solve this problem, instead they introduce an algorithm to overcome the issue of trivial null solutions by changing the objective function as:

$$J_m(U, V) = \sum_{j=1}^{c} \sum_{i=1}^{n} u_{ij}^m \, ||X_i - V_j||_A^2 + \sum_{i=1}^{C} \eta_i \sum_{j=1}^{n} (1 - u_{ij})^m \tag{7}$$

where η_i are suitable positive numbers. The authors of [23] discuss more details about (7), without considering membership functions. Implementation of such algorithm needs proper constraints and also requires good initializations, otherwise the accuracy and the results will not be reasonable [23]. U_{pcm} can obtain different values, since the implementation of PCM can be different because the boundaries for membership assignments with respect to each cluster are not completely defined.

In conclusion, crisp membership functions are not able to assign membership values to data objects participating in more than one cluster. Fuzzy membership functions reduce the membership values assigned to data objects with respect to each cluster, and possibilistic membership function is not well defined with respect to clusters. BFPM avoids the problem of reducing the membership degrees of objects when the number of clusters increases.

$$U_{bfpm}(L_1) = \begin{bmatrix} 1.0\ ,\ 1.0\ ,\ \mathbf{1.0}\ ,\ 1.0\ ,\ 1.0 \\ 0.0\ ,\ 0.5\ ,\ \mathbf{1.0}\ ,\ 0.5\ ,\ 0.0 \end{bmatrix} U_{bfpm}(L_2) = \begin{bmatrix} 0.0\ ,\ 0.5\ ,\ \mathbf{1.0}\ ,\ 0.5\ ,\ 0.0 \\ 1.0\ ,\ 1.0\ ,\ \mathbf{1.0}\ ,\ 1.0\ ,\ 1.0 \end{bmatrix}$$

BFPM allows data objects (such as the origin in the lines presented by previous example) to be members of all clusters with full membership. Additionally, BFPM may show which members can affect the algorithm if moved to other clusters. In critical systems, identifying these types of objects is a big advantage, because we may see how to encourage or prevent objects from contributing to other clusters. The method also includes those data objects that participate in just one cluster. Some of the issues on membership functions are described in [6, 24]. In [24] some other examples on different membership methods are discussed.

3 Similarity Functions

Similarity function is a fundamental part in learning algorithms [6, 28–32], as any agent, classifier, or method make use of these functions. Most of the learning methods compare a given problem with other problems to find the most suitable solution. This methodology indicates that the solution for the most similar problem can be the desired solution for the given problem [33].

Distance functions are based on the similarity between data objects or use probability measures. Tables 1, 2 and 3 show some well-known similarity functions (Eqs. 8–26) in L_1, L_2, and L_n norms [38, 39]. The taxonomy is divided into two categories: *vector* and *probabilistic* approaches. P and Q represent data objects or probability measures, in d dimensional search space, and $D(P, Q)$ presents a distance function between P and Q. Equation (13) is introduced to normalize the search space

Table 1 Distance functions or probability measures on Minkowski family

Minkowski family	Euclidean (L_2)	$D_E = \sqrt{\sum_{i=1}^{d} \lvert P_i - Q_i \rvert^2}$	(8)
	City block (L_1) [34]	$D_{CB} = \sum_{i=1}^{d} \lvert P_i - Q_i \rvert$	(9)
	Minkowski (L_p) [34]	$D_{MK} = \sqrt[p]{\sum_{i=1}^{d} \lvert P_i - Q_i \rvert^p}$	(10)
	Chebyshev (L_∞) [35]	$D_{Checb} = max_i \lvert P_i - Q_i \rvert$	(11)

Table 2 Distance functions or probability measures on Lvovich Chebyshev (L_1) family

Lvovich family	Sorensen [28]	$D_{Sor} = \frac{\sum_{i=1}^{d} \lvert P_i - Q_i \rvert}{\sum_{i=1}^{d} (P_i + Q_i)}$	(12)
	Gower [36]	$D_{Gov} = \frac{1}{d} d \sum_{i=1}^{d} \frac{\lvert P_i - Q_i \rvert}{R_i}$	(13)
		$D_{Gov} = \frac{1}{d} \sum_{i=1}^{d} \lvert P_i - Q_i \rvert$	(14)
L_1 family	Soergel [29]	$D_{Sg} = \frac{\sum_{i=1}^{d} \lvert P_i - Q_i \rvert}{\sum_{i=1}^{d} max(P_i, Q_i)}$	(15)
	Kulczynski [30]	$D_{Sg} = \frac{\sum_{i=1}^{d} \lvert P_i - Q_i \rvert}{\sum_{i=1}^{d} min(P_i, Q_i)}$	(16)
	Canberra [30]	$D_{Can} = \sum_{i=1}^{d} \frac{\lvert P_i - Q_i \rvert}{P_i + Q_i}$	(17)
	Lorentzian [30]	$D_{Lor} = \sum_{i=1}^{d} ln(1 + \lvert P_i - Q_i \rvert)$	(18)

Table 3 Distance functions or probability measures on x^2 (L_2) family

x^2 family	Squared euclidean	$D_{SE} = \sum_{i=1}^{d} (P_i - Q_i)^2$	(19)
	Pearson x^2 [1]	$D_P = \sum_{i=1}^{d} \frac{(P_i - Q_i)^2}{Q_i}$	(20)
L_2 family	Neyman x^2 [1]	$D_N = \sum_{i=1}^{d} \frac{(P_i - Q_i)^2}{P_i}$	(21)
	Squared x^2 [1]	$D_{SQ} = \sum_{i=1}^{d} \frac{(P_i - Q_i)^2}{P_i + Q_i}$	(22)
	Probabilistic x^2 [37]	$D_{PSQ} = 2 \sum_{i=1}^{d} \frac{(P_i - Q_i)^2}{P_i + Q_i}$	(23)
	Divergence [37]	$D_{Div} = 2 \sum_{i=1}^{d} \frac{(P_i - Q_i)^2}{(P_i + Q_i)^2}$	(24)
	Clark [30]	$D_{Clk} = \sqrt{\sum_{i=1}^{d} \left(\frac{(P_i - Q_i)}{(P_i + Q_i)} \right)^2}$	(25)
	Additive x^2 [30]	$D_{Ad} = \sum_{i=1}^{d} \frac{(P_i - Q_i)^2 (P_i + Q_i)}{(P_i Q_i)}$	(26)

boundaries by dividing the equation by R, the range of the population in the data set. The method scales down the search space by dividing the equation by d, the number of dimensions [36]. Asymmetric distance functions (Pearson (20), Neyman (21)) and symmetric versions of those functions $\left(\text{squared } x^2 \text{ (22)}\right)$ have been proposed, additionally to probabilistic symmetric x^2 (23) functions. There are other useful distance functions such as distance functions based on histograms, signatures, and probability density [40, 41] that we do not discuss in this paper.

3.1 Challenges on Similarity Functions

Assume there are two objects in a three dimensional search space, such as $O_1 = (2, 2, 5)$ and $O_2 = (1, 1, 1)$, and a prototype $P = (2, 2, 2)$. Now if we use a distance function such as Euclidean distance, object O_2 seems overall more similar to the prototype, but from a features' perspective, O_1 is more similar to the prototype when compared with O_2 given that they share two out of three features. This example motivates the following distance functions. These functions can be applied in high dimensional search spaces ($d^{'} \gg d$) typical of big data applications [16, 42, 43] where $d^{'}$ is a very large number. Let us consider:

$$O_1^{'} = (2, 2, 2, ..., x), O_2^{'} = (1, 1, 1, ..., 1), P^{'} = (2, 2, 2, ..., 2)$$

where

$$O_{1,1}^{'} = O_{1,2}^{'} = O_{1,3}^{'} = \cdots = O_{1,d'-1}^{'} = 2 \quad and \quad O_{1,d'}^{'} = x = \sqrt{d'} + 2$$

$$O_{2,1}^{'} = O_{2,2}^{'} = O_{2,3}^{'} = \cdots = O_{2,d'}^{'} = 1$$

$$P_1^{'} = P_2^{'} = P_3^{'} = \cdots = P_{d'}^{'} = 2$$

According to all similarity functions presented in Tables 1, 2 and 3, we see how these functions may have some dominant features ($x > \sqrt{d'} + 2$) that may cause algorithms to misclassify data objects.
We should evaluate the data objects' features from different perspectives, not just using the same scale. This is because each feature has its own effect on the similarity function and a single feature should not have a large impact on the final result.

3.2 Weighted Feature Distances

Assume a set of n objects represented by $O = \{o_1, o_2, \ ... \ , o_n\}$ in which each object is typically represented by numerical $feature - vector$ data, with the same priority in features, that has the form $X = \{x_1, ... \ , x_m\} \subset R^d$, where d is the dimension of the search space or the number of features. We introduce Weighted Feature Distance (WFD) that overcome some of the issues with distance function that we have described.
$\boldsymbol{WFD_{(L_1)}}$: Weighted feature distance (WFD_{L_1}) for L_1 norm is:

$$WFD_{(L_1)} = \left(|W_iO_i - W_jO_j|\right) =$$

$$= \sum_{k=1}^{d} \left(|w_k x_{ik} - w_k^{'} x_{jk}|\right) \tag{27}$$

***WFD*$_{(L_2)}$**: Weighted feature distance (WFD_{L_2}) for L_2 norm is:

$$WFD_{L_2} = \sqrt{\left(W_iO_i - W_jO_j\right)^2} =$$

$$= \left(\sum_{k=1}^{d} \left(|w_k x_{ik} - w_k^{'} x_{jk}|^2\right) \right)^{(\frac{1}{2})} \tag{28}$$

where d is the number of variables, or dimensions for numerical data objects. w_k and $w_k^{'}$ are the weights assigned to features of the first and the second objects respectively. We make ($w_k = w_k^{'}$) if both objects are in the same scale.
We can also obtain the Euclidean distance function from (28) by assigning the same values to w_k as:

$$w_1 = w_2 = \cdots = w_d = 1$$

***WFD*$_{(L_p)}$**: Weighted feature distance (WFD_{L_p}) for L_p norm is:

$$WFD_{(L_1)} = \left(|W_iO_i - W_jO_j|^p\right)^{(\frac{1}{r})} =$$

$$= \sum_{k=1}^{d} \left(|w_k x_{ik} - w_k^{'} x_{jk}|^p\right)^{(\frac{1}{r})} \tag{29}$$

where d is the number of variables, or dimensions for numerical data objects. p and r are coefficients that allow us to use different metrics but p and r can be equal. w_k and $w_k^{'}$ are the weights assigned to features of the first and the second objects respectively. ($w_k = w_k^{'}$), if both objects are in the same scale.

4 Data Types

Data mining techniques extract knowledge from data objects. To obtain the most accurate results, we need to consider the data types in our mining algorithms. Each type of object has its own characteristic and behaviour in data sets. The type of objects discussed in this paper help to avoid the cost of redoing mining techniques caused by treating objects in a wrong way.

4.1 Data Objects Taxonomies

Data mining methods evaluate data objects based on their (*descriptive* and *predictive*) patterns. The type of data objects should be considered, as each type of data object has different effect on the final results [44]. For instance, data objects known as *outlier(s)* are interesting objects in anomaly detection. On the other hand, outliers do not play any role in other applications since they are considered noise. Since each type of data object has different effects on the final result of an algorithm, we aim to look at different types of data from different perspectives. We start with the simplest definition of data objects and categorize them into single variable or with two or more variables [45].

- ***Univariate Data Object***:
  Observations on a single variable on data sets $X = \{x_1, x_2, ..., x_n\}$, where n is the number of single variable observations (x_i). Univariate Data Object can be categorized into two groups:

  1. Categorical or qualitative [31], that can be represented by *frequency distributions* and *bar charts*.
  2. Numerical or quantitative, which can be discrete or continuous data. *Dotplots* can be used to represent this type of variables.

- ***Multivariate Data Object***:
  Observations on a set of variables on data sets or populations presented as $X = \{X_1, X_2, ..., X_n\}$, where $X_i = \{x_1, x_2, ..., x_d\}$, n is the number of observations, and d is the number of variables or dimensions. Each variable can be a member of the above mentioned categories.

4.2 Complex and Advanced Objects

The growth of data in various types prevents data taxonomies for classifying data objects into above mentioned categories. Methods dealing with data objects need to distinguish their type to create more efficient methodologies and data mining algorithms. *Complex* and *Advanced* categories are two main topics for sophisticated data objects.

These objects have sophisticated structures, and also need advanced techniques for storage, representation, retrieval and analysis. Table 4 shows these data objects without the details. Further information can be found in [24]. An advantage of sophisticated objects is in allowing miners to reduce the cost of using similarity functions on these type of objects instead of comparing the data objects individually. For example two networks can be compared at once instead of being compared individually.

Table 4 Data types (Complex and advanced data objects)

Complex objects	Advanced objects
Structured data object [46]	Sequential patterns [47, 48]
Semi-structured data object [49]	Graph and sub-graph patterns [50, 51]
Unstructured data object [52]	Objects in interconnected networks [53, 54]
Spatial data object [55]	Data stream or stream data [56]
Hypertext [57]	Time series [58]
Multimedia [59]	

4.3 Outlier and Outstanding Objects

Data objects from each of the categories previously presented, can be considered as normal data objects that do fit the data model and obey the discovered data patterns. Now we introduce some data objects known as *Outlier* and *Outstanding*, that cannot be considered as normal data objects. These data objects affect the results obtained from knowledge extracted from data sets. Data objects from these categories can be any data object from above mentioned categorizes (complex, advanced, univariate, and multivariate data objects). Outliers and outstanding objects are important since they have potential ability to change the results produced by the learning algorithms.
Outlier: A data set may contain objects that do not fit the model of the data, and do not obey the discovered patterns [60, 61]. These data objects are called '*outliers*' [17, 24]. Outliers are important because they might change the behaviour of the model, as they are far from the discovered patterns and are mostly known as noise or exceptions. Outliers are useful in some applications such as fraud and anomaly detection [62], as these rare cases are more interesting than the normal cases. Outlier analysis is used in a data mining technique known as *outlier mining*.
Outstanding Objects Unlike outliers, a data set may contain objects that do fit the model of the data and obey the discovered patterns fully, even in all models or clusters. These data objects are important because they do not change the behaviour of the model, as they are in the discovered patterns and are known as full members. These critical objects named as "*outstanding*" objects cannot be removed from any cluster that they participate in [25]. The another important property of outstanding objects is that they may easily move from one cluster to another by small changes in even one dimension [24]. The crossing lines at origin example describes the behaviour and properties of outstanding objects. Origin should be a member of each line with full membership degree, otherwise each line without the origin can not be considered as a subspace. In such cases, we can see the importance of outstanding objects, in having full membership in several or in all objective functions [63].

In next section we describe some experimental results on clustering methods to illustrate how mining methods deal with outstanding objects, and how these data objects can affect the final results.

5 Experimental Results

Experiments are based on three scenarios. The first scenario compares the accuracy of membership functions on clustering and classification methods on some data sets shown in Table 5. In the second scenario, we check the effect of dominant features on similarity functions and consequently on final results of clustering methods. Data sets are selected based on a different number of features as we aim to check how proposed methods can be influenced by dominant features. Finally, the third scenario provides an environment to evaluate the behaviour of critical areas and objects that we have called *Outstanding*. In all scenarios in our experiments, we compare the accuracy of different fuzzy and possibilistic methods with BFPM and BFPM-WFD algorithms presented in Algorithms (1) and (2).

BFPM Algorithm This algorithm uses the conventional distance functions for membership assignments. Equations (30) and (31) show how the algorithm calculates (u_{ij}) and how the prototypes (v_j) will be updated in each iteration. The algorithm runs until the condition is false:

$$max_{1\leq k\leq c}\{||V_{k,new} - V_{k,old}||^2\} < \varepsilon$$

The value assigned to ε is a predetermined constant that varies based on the type of objects and clustering problems.

U is the $(n \times c)$ partition matrix, $V = v_1, v_2, ..., v_c$ is the vector of c cluster centers in $\mathfrak{R}^d$, m is the fuzzification constant, and $||.||_A$ is any inner product A-induced norm [6, 64], and Euclidean distance function presented by (32).

$$D_E = \sqrt{\sum_{i=1}^{d} |X_i - Y_i|^2}$$

$$= \sqrt{(X_1 - Y_1)^2 + (X_2 - Y_2)^2 + \cdots + (X_d - Y_d)^2} \tag{32}$$

Table 5 Multi dimensional data sets

Dataset	Attributes	No. objects	Clusters
Iris	4	150	3
Pima Indians	8	768	2
Yeast	8	1299	4
MAGIC	11	19200	2
Dermatology	34	358	6
Libras	90	360	15

Algorithm 1 BFPM Algorithm

Input: **X,** c, m
Output: **U, V**
Initialize V;
while $max_{1\le k\le c}\{||V_{k,new} - V_{k,old}||^2\} > \varepsilon$ **do**

$$u_{ij} = \Big[\sum_{k=1}^{c} \big(\frac{||X_i - v_j||}{||X_i - v_k||}\big)^{\frac{2}{m-1}}\Big]^{\frac{1}{m}}, \quad \forall i,j \tag{30}$$

$$V_j = \frac{\sum_{i=1}^{n}(u_{ij})^m x_i}{\sum_{i=1}^{n}(u_{ij})^m}, \quad \forall j\,; \qquad (0 < \frac{1}{c}\sum_{j=1}^{c} u_{ij} \le 1). \tag{31}$$

end while

where d is the number of features or dimensions, and X and Y are two different objects in d dimensional search space.

BFPM-WFD Since BFPM algorithm assigns (u_{ij}) based only on the total distance shown by (32), we implement algorithm BFPM-WFD (BFPM Weighted Feature Distance) not only to compare the objects based on their similarity using the distance function, but also to check the similarity between features of objects and similar features of prototypes individually.

Algorithm 2 BFPM-WFD

Input: **X,** c, m
Output: **U, V**
Initialize V;
while $max_{1\le k\le c}\{||V_{k,new} - V_{k,old}||^2\} > \varepsilon$ **do**

$$\Big\{ u_{ij} = \Big[\sum_{k=1}^{c} \big(\frac{||X_i - v_j||}{||X_i - v_k||}\big)^{\frac{2}{m-1}}\Big]^{\frac{1}{m}}, \quad \forall i,j\,;$$

$$||X_i - X_j|| = \Big(\sum_{f=1}^{d} \big(|w_f . x_{if} - w'_f . x_{jf}|^2\big)\Big)^{(\frac{1}{2})}\Big\} \tag{33}$$

$$V_j = \frac{\sum_{i=1}^{n}(u_{ij})^m x_i}{\sum_{i=1}^{n}(u_{ij})^m}, \quad \forall j\,; \qquad (0 < \frac{1}{c}\sum_{j=1}^{c} u_{ij} \le 1). \tag{34}$$

end while

w_f and w'_f are weights assigned to features (x_{if} and x_{jf}) of objects X_i and X_j respectively, presented by (28). Table 6 illustrates the compared results between BFPM and other fuzzy and possibilistic methods: Type-1 fuzzy sets (T1), Interval Type-2 fuzzy sets (IT2), General Type-2 (GT2), Quasi-T2 (QT2), FCM and PCM on four data sets

Table 6 Compared accuracy between conventional fuzzy, possibilistic, and BFPM methods

Methods	Iris	Pima Indian	Yeast	MAGIC
T1 (fuzzy) [65]	95.15	73.59	60.03	77.26
IT2 (fuzzy) [65]	94.18	74.38	54.81	75.63
QT2 (fuzzy) [65]	94.28	75.05	55.97	77.44
GT2 (fuzzy) [65]	94.76	74.40	58.22	78.14
FCM (fuzzy) [12]	88.6	74	67.4	54
PCM (possibilistic) [12]	89.4	59.8	32.8	62
BFPM	97.33	99.9	67.71	100.0
BFPM-WFD	100.0	100.0	82.3	100.0

Table 7 Compared accuracy based on distance functions

Dataset ↓ | $\overrightarrow{Dis.Func.}$	Euclidean (L_2)	$WFD_{L_2}(w = \frac{1}{2})$	WFD (L_2) $w = \frac{1}{3}$	$WFD_{L_2}(w = \frac{1}{d})$
Irish	97.33	100	100	100
Pima	99.9	100	100	100
Yeast	67.71	77.2	77.3	**82.03**
MAGIC	100.0	100.0	100.0	100.0
Dermatology	77.4	89.5	83.0	**92.4**
Libras	57.0	**69.0**	62.5	61.4

"Iris", "Pima", "Yeast" and "MAGIC". This comparison is based on the first scenario, and as results show, BFPM performs better than the conventional fuzzy and possibilistic methods.

Table 7 compares the accuracy between WFD with different weights ($w = 1/2$, $w = 1/3, w = 1/d$) and Euclidean distance function on different data sets "Iris", "Pima", "Yeast", "Magic", "Dermatology" and "Libras", where d is the number of dimensions. According to the table, dominant features has less impact on weighted features distance functions. The table also shows that in some data sets such as "Libras" larger values for assigned weights are most desirable and in some other such as "Yeast" lower values are most suitable. The comparison between conventional similarity function and WFD was implemented with respect to the second scenario.

Table 8 Outstanding objects with ability to move from one cluster to another

Dataset	No. objects	>90%	>80%	>70%
Irish	150	25	99	99
Pima	768	677	751	751
Yeast	1299	868	1135	1264
MAGIC	19200	0	34	424
Dermatology	358	286	331	355
Libras	360	238	317	336

According to the last scenario, we aim to check the ability of outstanding objects to participating in other clusters. Table 8 demonstrates the potential ability of data objects to get membership values from the closest cluster, besides their own clusters. For example, the first row of the table shows that 25 data objects from Iris data set have the potential ability of more than 90% to participate in the closest cluster. As the table presents, some data objects are able to move to another cluster with small changes. These kinds of behavior can be beneficial or produce errors.

In some safety critical systems such as cancerous human cell detection, or fraudulent banking transactions, we need to prevent data objects to move to other clusters.

Figures 1, 2, 3 and 4 plot data objects on data sets "Iris", and "Libras" [66] obtained by Fuzzy and BFPM methods with respect to two closest clusters. By comparing the plots for BFPM and fuzzy methods [24], critical areas and objects are being shown. This comparison is being highlighted when we look at the accuracy of Fuzzy, Possibilistic, and BFPM as well as considering the ability of outstanding objects to affect performance. In fuzzy methods, data objects are mostly separated. In this situation the critical areas are not shown, but instead in BFPM method, critical areas, and also outstanding objects may be identified.

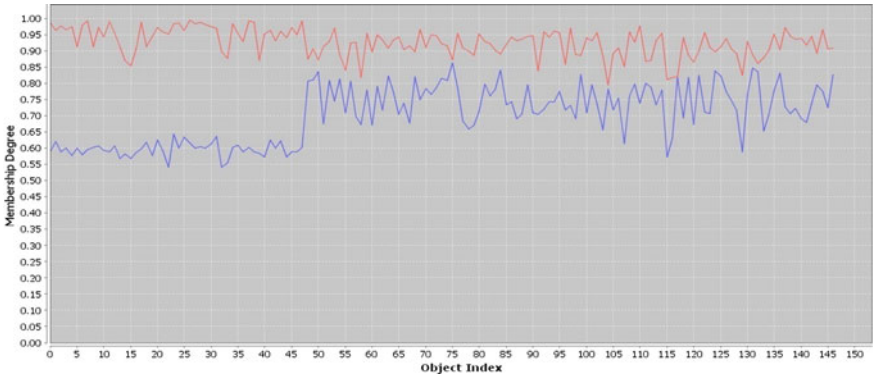

Fig. 1 Mutation plot for Iris data set, prepared by BFPM method

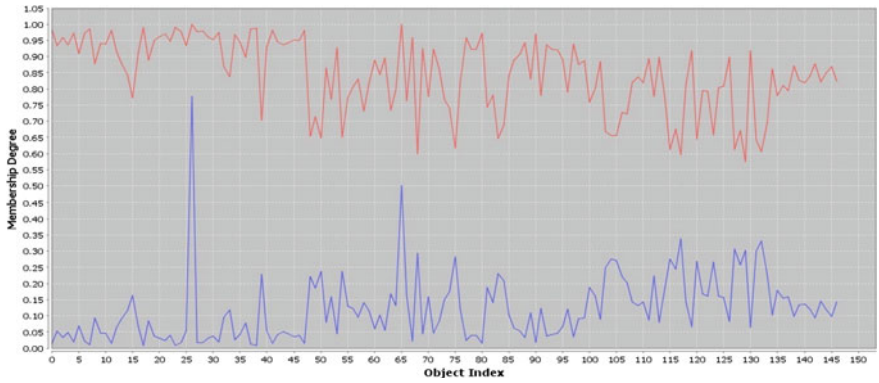

Fig. 2 Mutation plot for Iris data set, prepared by Fuzzy method

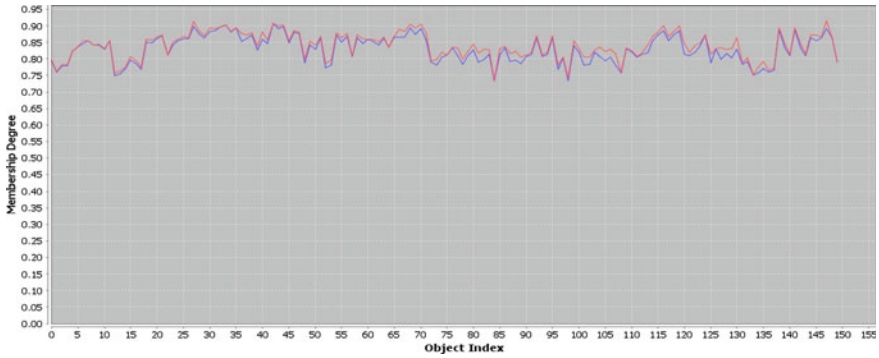

Fig. 3 Mutation plot for Libras data set, prepared by BFPM method

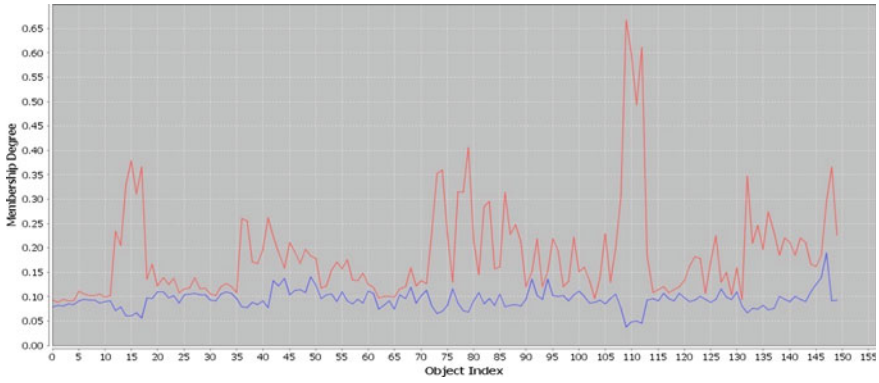

Fig. 4 Mutation plot for Libras data set, prepared by Fuzzy method

6 Conclusion

This chapter describes some of the most important parameters in learning methods for partitioning, such as similarity functions, membership assignments, and type of data objects. Additionally, we described the most used and well known membership functions. The functionality of these membership functions was compared on different scenarios. The challenges in using similarity functions that could deal correctly with dominant features is another concept studied in this chapter. The presented similarity functions were compared in different aspects.

This chapter also discusses different types of data objects and their potential effect on learning algorithm's performance. Critical objects known as outstanding were described in the context of several examples.

Our results show that BFPM performs better than other conventional fuzzy and possibilistic algorithms discussed in this chapter on the presented data sets. WFD helps learning methods to handle the impact of the dominant features in their processing steps. Outstanding objects are the most critical and many learning methods do not even consider this type of data objects. BFPM provides the most flexible environment for outstanding objects by analysing how critical objects can make the system more stable. We found that the most appropriate membership and similarity function should be selected, regarding the type of data objects considered by our model.

References

1. S.-H. Cha, "Comprehensive Survey On Distance/Similarity Measures Between Probability Density Functions," Int. J. Math. Models Methods Appl. Sci., vol. 1, no. 4, pp. 300–307, 2007.
2. P. N. Tan, M. Steinbach, V. Kumar, "Introduction to Data Mining Instructor's Solution Manual," Pearson Addison-Wesley, 2006.
3. P. N. Tan, M. Steinbach, V. Kumar, "Introduction to Data Mining," Pearson Wesley, 2006.
4. B. Taskar, E. Segal, D. Koller, "Probabilistic classification and clustering in relational data," In Proc. Int. Joint Conf. Artificial Intelligence (IJCAI01), pp. 870–878, Seattle, WA, 2001.
5. H. Yazdani, H. Kwasnicka, "Fuzzy Classiffication Method in Credit Risk," in Springer Int. Conf. Computer and Computational Intelligence, pp. 495–505, 2012.
6. H. Yazdani, D. O. Arroyo, H. Kwasnick, "New Similarity Functions", IEEE, AIPR, pp. 47–52, 2016.
7. R. Xu, D. C. Wunsch, "Recent advances in cluster analysis," Intelligent Computing and Cybernetics, 2008.
8. D. T. Anderson, J. C. Bezdek, M. Popescu, J. M. Keller. "Comparing Fuzzy, Probabilistic, and Posibilistic Partitions," IEEE Transactions On Fuzzy Systems, Vol. 18, No. 5, 2010.
9. C. Borgelt, "Prototype-based Classification and Clustering." Magdeburg, 2005.
10. L. F. S. Coletta, L. Vendramin, E. R. Hruschka, R. J. G. B. Campello, W. Pedrycz, "Collaborative Fuzzy Clustering Algorithms: Some Refinements and Design Guidelines," IEEE Transactions On Fuzzy Systems, Vol. 20, No. 3, pp. 444–462, 2012.
11. X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z. H. Zhou, M. Steinbach, D. J. Hand, D. Steinberg. "Top 10 algorithms in data mining," Springer-Verlag London, 2007.
12. N. R. Pal, K. Pal, J. M. Keller, J. C. Bezdek. "A Possibilistic Fuzzy c-Means Clustering Algorithm," IEEE Transactions On Fuzzy Systems, Vol. 13, No. 4. 2005.

13. S. Singh, A.K. Solanki, N. Trivedi, M. Kumar, “Data Mining Challenges and Knowledge Discovery in Real Life Applications,” IEEE, 978-1-4244-8679-3/11/, 2011.
14. T. Hastie, R. Tibshirani, J. Friedman, “The Elements of Statistical Learning Data Mining, Inference, and Prediction,” Springer Series in Statistics, 2005.
15. L.A.Zadeh. “Fuzzy Sets,” Information and Control, 338–353, 1965.
16. R. J. Hathawaya, J. C. Bezdekb. “Extending fuzzy and probabilistic clustering to very large data sets,” Elsevier, 2006.
17. J. Han, M. Kamber, “Data Mining Concepts and Techniques,” Elsevier, Morgan Kaufmann series, 2006.
18. L.A. Zadeh, “Fuzzy Sets As A Basis For A Theory Of Possibility” North-Holland Publishing Company Fuzzy Sets and Systems 1, 1978.
19. L.A. Zadeh. “Toward Extended Fuzzy Logic- A First Step,” Elsevier, Fuzzy Sets and Systems, 3175–3181, 2009.
20. H. C. Huang, Y. Y. Chuang, C. S. Chen, “Multiple Kernel Fuzzy Clustering,” IEEE Transactions On Fuzzy Systems, 2011.
21. R. Krishnapuram, J. M. Keller, “A Possibilistic Approach to Clustering,” IEEE, Transaction On Fuzzy Systems, Vol. 1, No. 2. 1993.
22. F. Memoli, G. Carlsson, “Characterization, Stability and Convergence of Hierarchical Clustering Methods”, Journal of Machine Learning Research, pp. 1425–1470, 2010.
23. M. Barni, V. Cappellini, A. Mecocci, “Comments on A Possibilistic Approach to Clustering,” IEEE, Transactions On Fuzzy Systems, Vol. 4, No. 3. 1996.
24. H. Yazdani, H. Kwasnicka, “Issues on Critical Objects in Mining Algorithms”, IEEE, AIPR, pp. 53–58, 2016.
25. H. Yazdani, D. Ortiz-Arroyo, K. Choros, H. Kwasnicka, “Applying Bounded Fuzzy Possibilistic Method on Critical Objects”, IEEE, CINTI, 2016.
26. H. Yazdani, “Bounded Fuzzy Possibilistic On Different Search Spaces”, IEEE, CINTI, 2016.
27. G. Strang, “Introduction to Linear Algebra”, Wellesley-Cambridge Press, 2016.
28. J. Looman, J.B. Campbell, “Adaptation of Sorensen’s K For Estimating Unit Affinities In Prairie Vegetation,” Ecology, Vol. 41, No. 3, 1960.
29. V. Monev, “Introduction to Similarity Searching in Chemistry,” Communications in Mathematical And Computer Chemistry, No. 51, 2004.
30. P. Kumar, A. Johnson, “On A Symmetric Divergence Measure And Information Inequalities,” Journal of Inequalities in Pure And Applied Mathematics, Vol. 6, Issue 3, Article 65, 2005.
31. S. Boriah, V. Chandola, V. Kumar, “Similarity Measures For Categorical Data: A Comparative Evaluation,” SIAM, 2008.
32. M. Minor, A. Tartakovski, R. Bergmann, “Representation and structure-based similarity assessment for agile workflows,” Springer, 7th International Conf. on Case-Based Reasoning and Development, pp. 224–238, 2007.
33. X. Chen, X. Li, B. Ma, P. M.B. Vitanyi, “The Similarity Metric,” IEEE Transactions On Information Theory, Vol. 50, No. 12, 2004.
34. E.F. Krause, “Taxicab Geometry An Adventure in Non-Euclidean Geometry,” Dover, 1987.
35. D. M. J. Tax, R. Duin, D. De Ridder, “Classification, Parameter Estimation and State Estimation: An Engineering Approach Using MATLAB,” John Wiley and Sons, 2004.
36. J.C. Gower, “A General Coefficient Of Similarity And Some Of Its Properties,” Biometrics, Vol. 27, No. 4, pp. 857–871, 1971.
37. M. Deza and E. Deza, “Encyclopedia of Distances,” Springer-Verlag, 2014.
38. Y. Rubner, C. Tomasi, L. J. Guibas, “A Metric For Distributions With Applications to Image Databases,” IEEE International Conference on Computer Vision, 1998.
39. D.G. Gavin, W.W. Oswald, E.R. Wahl, J.W. Williams, “A Statistical Approach To Evaluating Distance Metrics And Analog Assignments For Pollen Records,” Quaternary Research, Vol. 60, Issue 3, pp. 356–367, 2003.
40. F.D. Jou, K.C. Fan, Y.L. Chang, “Efficient Matching of Large-Size Histograms,” Elsevier, Pattern Recognition, pp. 277–286, 2004.

41. S.H. Cha, "Taxonomy of Nominal Type Histogram Distance Measures," American Conference On Applied Mathematics, Harvard, Massachusetts, USA, pp. 24–26, 2008.
42. L. Parsons, E. Haque, H. Liu, "Subspace Clustering for High Dimensional Data: A Review," ACM. Sigkdd Explorations, Vol. 6, pp. 90–105, 2004.
43. T. C. Havens, J. C. Bezdek, C. Leckie, L. O. Hall, M. Palaniswami, "Fuzzy $c-$Mean Algorithms for Very Large Data," in IEEE Transactions on Fuzzy Information and Engineering (ICFIE), pp. 865–874, 2007.
44. R. Peck, J. L. Devore, "Statistics The Exploration and Analysis of Data," Cengage Learning, 2010.
45. W. Hardle, L. Simar, "Applied Multivariate Statistical Analysis," Springer, 2003.
46. H. Bunke, B.T.Messmer, "Similarity Measures for Structured Representations," Springer, Vol. 837, pp. 106–118, 1993.
47. J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, M.C. Hsu, "Mining Sequential Patterns by Pattern-Growth: The Prefix Span Approach," IEEE Transactions on Knowledge and Data Engineering, pp. 1424–1440, 2004.
48. S. Cong, J. Han, D. Padua, "Parallel Mining of Closed Sequential Patterns," Knowledge Discovery in Databases, pp. 562–567, 2005.
49. S. Chakrabarti, "Mining the Web: Statistical Analysis of Hypertext and Semi-Structured Data," Morgan Kaufmann, 2002.
50. P. Kefalas, P. Symeonidis, Y. Manolopoulos, "A Graph-Based Taxonomy of Recommendation Algorithms and Systems in LBSNs," IEEE Transactions On Knowledge And Data Engineering, Vol. 28, No. 3, pp. 604–622, 2016.
51. M. Kuramochi, G. Karypis, "Frequent Sub-graph Discovery," Data Mining, pp. 313–320, 2001.
52. S. Weiss, N. Indurkhya, T. Zhang, F. Damerau, "Text Mining: Predictive Methods for Analysing Unstructured Information," Springer, 2004.
53. P. J. Carrington, J. Scott, S.Wasserman, "Models and Methods in Social Network Analysis," Cambridge University Press, 2005.
54. P. Domingos, "Mining Social Networks for Viral Marketing," IEEE Intelligent Systems, pp. 80–82, 2005.
55. K. Koperski, J. Han, "Discovery of Spatial Association Rules in Geographic Information Databases," Large Spatial Databases, pp. 47–66, 1995.
56. J. Gehrke, F. Korn, D. Srivastava, "On Computing Correlated Aggregates Over Continuous Data Streams," Conf. Management of Data, pp. 13–24, 2001.
57. H. J. Oh, S. H. Myaeng, M. H. Lee, "A Practical Hypertext Categorization Method Using Links and Incrementally Available Class Information," Research and Development in Information Retrieval, pp. 264–271, 2000.
58. A. Bagnall, J. Lines, J. Hills, A. Bostrom "Time-Series Classification with COTE: The Collective of Transformation-Based Ensembles", IEEE Transactions On Knowledge And Data Engineering, Vol. 27, No. 9, pp. 2522–2535, 2015.
59. A. Hinneburg, D. A. Keim, "An Efficient Approach to Clustering in Large Multimedia Databases with Noise," Knowledge Discovery and Data Mining, pp. 58–65, 1998.
60. C. C. Aggarwal, "Outlier Analysis," Springer, 2013.
61. V. Barnett, T. Lewis, "Outliers in Statistical Data," John Wiley and Sons, 1994.
62. D. J. Weller-Fahy, B. J. Borghetti, A. A. Sodemann, "A Survey of Distance and Similarity Measures Used Within Network Intrusion Anomaly Detection", IEEE Communication Surveys and Tutorials, Vol. 17, No. 1, 2015.
63. H. Yazdani, H. Kwasnicka, D. Ortiz-Arroyo, "Multi Objective Particle Swarm Optimization Using Fuzzy Logic," in Springer Int. Conf. Computer and Computational Intelligence, pp. 224–233, 2011.
64. R. Xu, D. Wunsch, "Clustering," IEEE Press Series on Computational Intelligence, 2009.
65. O. Linda, Milos Manic, "General Type-2 Fuzzy C-Means Algorithm for Uncertain Fuzzy Clustering," in IEEE Transactions on Fuzzy Systems, Vol 20, pp. 883–897, 2012.
66. J. Zhou, C. L. P. Chen, L. Chen, H. X. Li, "A Collaborative Fuzzy Clustering Algorithm in Distributed Network Environments", IEEE, Transactions On Fuzzy Systems, Vol. 22, No. 6, pp. 1443–1456, 2014.

http://www.springer.com/978-3-319-53473-2

Data Science and Big Data: An Environment of Computational Intelligence

Pedrycz, W.; Chen, S.-M. (Eds.)
2017, VIII, 303 p. 101 illus., 80 illus. in color., Hardcover
ISBN: 978-3-319-53473-2