Fuzzy c-means for fuzzy hierarchical clustering

Vicenç Torra
IIIA-CSIC
Campus UAB s/n
08193 Bellaterra (Catalonia, Spain)
E-mail: vtorra@iiia.csic.es

***Abstract*— This paper describes an algorithm for building fuzzy hierarchies. This is, hierarchies where the elements can have fuzzy membership to the nodes. The paper presents an approach that mainly follows a bottom-up strategy, and describes the functions needed to operate with fuzzy attributes. An example of the application of the approach is also presented.**

I. Introduction

Clustering algorithms have been studied for a long time, and are used nowadays for a large number of applications. They are sucessfully in use in systems for information access, data mining or computer vision. Different algorithms have been developed using different approaches and considering different underlying assumptions on the data and on the final set of clusters. c-means, fuzzy c-means, self-organizing maps are some of the well known clustering algorithms. Existing algorithms can be classified according to several dimensions. Some of them are described below:

One of such dimensions is the *direction* of the clustering process. In this case, methods are divided into agglomerative ones and partitive ones. Agglomerative algorithms build clusters gathering together those records that are similar. This situation corresponds to a bottom-up strategy (or a bottom-up *direction*) *i.e.* from individual records to the set that contains all records. Partitive algorithms, instead, follow a top-down strategy. This is, clusters are defined by partitioning larger sets of records.

Another dimension corresponds to the membership of records to clusters. In this case, we can distinguish among crisp, fuzzy and probabilistic clusters. In crisp clusters, membership of a record into a cluster is boolean. This is, the record either belongs or not to the cluster. Instead, in the case of fuzzy clusters, membership is a matter of degree (in $[0, 1]$). At the same time, individual records can belong to several clusters. In the case of probabilistic clusters, membership is boolean but there is a distribution of probability of belonging to clusters.

A third dimension is the structure of the clusters. In short, this is whether the clusters themselves define a structure and, if so, which is the structure they define. The simplest case is when no structure is defined. Each cluster is understood as an independent object. Alternatively, clusters can define hierarchies or other complex structures.

Such dimensions can be used to classify clustering methods. For example, agglomerative clustering methods are bottom-up (agglomerative) crisp methods that naturally lead to hierarchical cluster structures. c-means is a top-down (partitive) crisp method where clusters do not have any particular relation. Fuzzy c-means is also a top-down (partitive) algorithm that lead to fuzzy clusters (fuzzy memberships of elements to clusters). Self-Organizing Maps (SOM) is also a partitive crisp algorithm but in this case, a grid structure is established among clusters. Several variations of SOM exist, corresponding to different structures established among clusters, being in most of the cases two-dimensional structures of clusters.

In this paper we study an approach to hierarchical clustering. In particular, we propose a method to build hierarchies of clusters where membership to clusters is fuzzy.

The structure of the paper is as follows. In the rest of this section, we motivate our work and relate it with existing approaches. Then, in Section II, we review fuzzy c-means. In Section III we consider distance functions for fuzzy sets. This is needed to compute the distance between pairs of fuzzy sets when building fuzzy clusters from fuzzy clusters. In Section IV, our new approach is described. Section V describes some experiments. The paper finishes with some conclusions and directions for future work.

A. Motivation and previous work

The motivation of this research is rooted in some research we have done applying fuzzy clustering methods to information access and privacy preserving data mining. In our GAMBAL system for information access and document visualization (partially described in [11]), users navigate through sets of textual documents through a hierarchical structure of clusters. The extension of GAMBAL using fuzzy clustering, and fuzzy hierarchies, permits that the user access documents through different points of the hierarchy. This increases the flexibility of the system. With respect to privacy preserving data mining, we have developed algorithms for database protection. Here, we understand database protection as the process applied to a database so that confidentiality is assured for the individuals represented in the database. A microaggregation method based on fuzzy clustering was developed. Hierarchical fuzzy clustering [10] is used to ensure that all clusters are small enough for assuring a low information loss, but not too small to avoid disclosure risk.

To work in such environments, we developed a system for computing a hierarchical structure of fuzzy clusters that used a top-down strategy. The approach started constructing the hierarchy applying a partitive fuzzy clustering techique (as fuzzy c-means) to a crisp set of elements. Then, once the hierarchy was already defined, an iterative process was applied to partition those clusters that were *too large*. In each iteration, one *large* fuzzy cluster was selected and a partitive fuzzy clustering technique was applied to it. In this way, new subclusters were obtained and the hierarchy was enlarged to include them. This is detailed in Algorithm 1.

Algorithm 1 Top-down algorithm

```
Algorithm TopDown (X: data) returns fuzzyHierarchy
begin
  P := Partition (X);
  while exists p ∈ P such that |p| is large do
      select p with large |p|;
      p' := Partition(p);
      include p' in P and link p' with p;
  end while
  return(P);
end
```

As indicated, when a fuzzy set μ was selected for being too large, it was partitioned again using a fuzzy clustering method into fuzzy clusters μ_i. At this point, as we used standard fuzzy c-means the algorithm required to start with a crisp partition. Such partition was defined from the support of the fuzzy cluster, selecting those elements that had membership above a certain threshold α. This is, we defined $X' := \{x | x \in X, \mu(x) \geq \alpha\}$. The partition of X' lead to a value $\mu'_i(x)$ for each subcluster i. Once the domain X was partitioned into fuzzy sets μ'_i, the new clusters were added to the tree and the membership of x to each subcluster in the tree was defined as a combination of the membership of x to μ and the membership of x to subclusters μ_i. In short, membership of x to each subclusters was defined as the product of the membership to the original cluster $\mu(x)$ and to the new one. This is, $\mu_i(x) = \mu(x) \cdot \mu'_i(x)$. In this way, when an element x had a low membership to μ it should also have a low membership to each of the subclusters. An important consequence of this construction is that the tree defines at any point of the iteration process a fuzzy partition. This is, the memberships of x into all leaves of the hierarchy add to one.

This approach for building fuzzy hierarchies implies that elements x with a low membership to X' have the same influence on the partition of x than elements x with large influence. Also, depending on the threshold α we can have large sets of elements (X') where almost all sets have low memberships. This situation is even more noticeable when partitions are partitioned again, as memberships are reduced on everystep due to their combination via the product operator.

In this paper we develop an alternative approach for building a fuzzy hierarchy of elements. The new approach mainly follows a bottom-up strategy, using fuzzy c-means in this process.

The work developed here has some similarities with some recent work on clustering. For example, with the work in [3] were fuzzy clusters are computed using fuzzy c-means and then the similarity between clusters is established to build a dendrogram. The approach developed here is different as our ultimate goal is to build a fuzzy hierarchical structure. Some other differences can be found in relation to the way distance / similarity among fuzzy clusters is computed. Other approaches on building hierarchies of clusters (as in hierarchical SOM or hierarchical clustering methods) are also different for the same reason.

II. Fuzzy c-means

This section reviews the fuzzy c-means, one of the existing clustering methods for building fuzzy partitions. This method will be used in this paper as the basic tool for building a fuzzy hierarchy.

In the description, we will use the following notation: $x_k \in \mathbb{R}^p$ for $k = 1, \ldots, n$ stands for the elements (in a given p dimensional space) that are to be clustered and u_{ik} is the membership of element x_k to the i-th cluster.

Fuzzy c-means is a fuzzy clustering method that generalizes c-means (also known by k-means). While c-means builds a crisp partition with c clusters, fuzzy c-means builds a fuzzy one (also with c clusters). Due to this fuzzy nature, in this latter case elements are allowed to belong to more than one cluster. As said, u_{ik} is used to formalize the membership of element x_k to the i-cluster. The crisp case corresponds to have u_{ik} as either 0 or 1 (boolean membership) while the fuzzy case corresponds to have u_{ik} in $[0, 1]$. In this latter case, $u_{ik} = 0$ corresponds to non-membership and $u_{ik} = 1$ corresponds to full membership to cluster i. Values in-between correspond to partial membership (the largest the value, the greatest the membership).

Formally speaking, fuzzy c-means is defined as the solution of the following minimization problem (see [1] or [8] for details):

$$J_{FCM}(U, V) = \{\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^m ||x_k - v_i||^2\} \qquad (1)$$

subject to the constraints $u_{ik} \in [0, 1]$ and $\sum_{i=1}^{c} u_{ik} = 1$ for all k. We will denote the values u that satisfy these constraints by M.

In this formulation, v_i corresponds to the centroid (cluster center/cluster representative) of the i-th cluster and m is a parameter ($m \geq 1$) that plays a central role. With values of m near to 1, solutions tends to be crisp (with the particular case that $m = 1$ corresponds to the crisp c-means). Instead, larger values of m yield to clusters with increasing fuzziness in their boundaries.

To solve this problem, an iterative process is applied. The method interleaves two steps. One that estimates the optimal membership functions of elements to clusters (when centroids are fixed) and another that estimates the centroids for each cluster (when membership functions are fixed). This iterative process is described in Algorithm 2.

This method does not assure to find the optimal solution of the minimization problem given above but a local optimum. Different starting points can lead to different solutions.

Algorithm 2 Fuzzy c-means

Step 1: Generate an initial U and V

Step 2: Solve $min_{U \in M} J(U, V)$ computing:

$$u_{ik} = \Big(\sum_{j=1}^{c} \Big(\frac{||x_k - v_i||^2}{||x_k - v_j||^2} \Big)^{\frac{1}{m-1}} \Big)^{-1}$$

Step 3: Solve $min_V J(U, V)$ computing:

$$v_i = \frac{\sum_{k=1} n(u_{ik})^m x_k}{\sum_{k=1}^{n} (u_{ik})^m}$$

Step 4: If the solution does not converge, go to step 2; otherwise, stop

Recently an alternative fuzzy clustering method was proposed [5] (see also [6]). It is the so-called entropy-based fuzzy c-means (EFCM). While fuzzy c-means introduces fuzziness in the solution by means of the parameter m, the entropy-based fuzzy c-means uses a term based on entropy and a parameter λ ($\lambda \geq 0$) to force a fuzzy solution.

A variation of these clustering methods was introduced [7] to introduce a size variable for each cluster. Such parameter was introduced to reduce misclassification when there are clusters of different size. Otherwise, two adjacent clusters have equal membership function (equal to 0.5) in the midpoint between the two centroids. The size of the i-th cluster is represented with the parameter α_i (the largest is α_i, the largest is the proportion of elements that belong to the i-th cluster). A similar approach is given in [4].

III. Distance functions on fuzzy sets

Nowadays, several expressions have been defined for computing distances between fuzzy sets. Bloch [2] and [12] reviews some of them. We offer below a classification of such measures with respect to two dimensions. The first dimension corresponds to the type of outcome of the distance:

Real number. The distance is just a real value in a particular interval (*e.g.* the $[0, 1]$ interval).

Fuzzy number. The distance between two fuzzy sets is another fuzzy set. The definition of a fuzzy distance in terms of the extension principle is an example of this situation.

The second dimension corresponds to the type of information that is extracted from the fuzzy sets and used when computing the distance. This is, whether only membership values are used or other spatial characteristics are also considered. Particular indices can also be integrated when defining the distance. This classification is detailed below:

Distance comparing the membership values. A typical case is to compute just the difference between the membership function for two sets. For example, let δ be a distance function for real numbers, then the distance between fuzzy sets μ_1 and μ_2 is defined as:

$$d_\delta \mu_1, \mu_2 = \sum_x \delta(\mu_1(x), \mu_2(x))$$

Different instances of δ lead to different distances.
A similar approach are the distances that compare the union and the intersection of both fuzzy sets μ_1 and μ_2.
All such distances have the same drawback: when fuzzy sets are disjoint, the distance is always constant, no matter the distance between the supports of μ_1 and μ_2.

Distance considering spatial characteristics. This is the case of defining the distance between two fuzzy sets μ_1 and μ_2 in terms of the distance between the α-cuts of μ_1 and μ_2. The distance between α-cuts give information about the spatial characteristics.
Representing the α-cut of μ by μ_α this measure is defined as:

$$d(\mu, \mu') = \int_0^1 \delta(\mu_\alpha, \mu'_\alpha) d\alpha \qquad (2)$$

where δ is a distance over crisp sets.
Another approach is to compute the distance between two fuzzy sets mapping fuzzy sets in a n dimensional space into a $n+1$ space. Thus, fuzzy sets are transformed into crisp sets. This is to exted the original elements x in the original n-dimensional space into $(x|\mu(x))$. In this case, standard distances can be used for comparing the elements in the $(n+1)$-dimensional space.
Now, as distances include spatial characteristics the problem of equal distance when sets are disjoint is solved. Nevertheless, the computation cost becomes a relevant issue.

Distance based on indices. In this case, indices representing characteristics are extracted from the membership functions, and a distance is applied to such indices. When characteristics are represented as vectors, distance between two vectors can be used. The fuzzy cardinality of the set is an example of such indices.

In our application, distance functions will be used to help in the construction of a hierarchical fuzzy structure using a bottom-up strategy. To do so, we need the distance functions to compute distances between pairs of fuzzy clusters. As our approach is integrated with a fuzzy c-means, and as in such method the spatial characteristics of the data is used, we will follow here the same approach. Thus, spatial characteristics of fuzzy sets are taken into account when computing distances between fuzzy sets.

Accordingly, we follow the approach described in Equation 2. This is, the distance between two fuzzy sets is computed from the distance between α-cuts. Nevertheless, instead of using Equation 2 straightforwardly, we will exploit the structure of the membership functions in the computation of the distance so that the computation time is reduced.

It is known that fuzzy c-means returns a fuzzy partition of the application domain where each membership function μ_i is normal in exactly one point. Formally speaking, for each μ_i there is only one $x_i \in [0,1]^N$ such that $\mu_i(x_i) = 1$. Such normal point is the centroid of the fuzzy set μ_i and corresponds to v_i using the terminology described in Section II. Also, fuzzy sets generated by Fuzzy C-Means have memberships that are decreasing with respect to the distance. Taking into account this fact, the distance between two fuzzy sets μ_1 and μ_2 are defined in terms of the distance between the α-cuts having them restricted to the convex set $\alpha v_1 + (1-\alpha)v_2$ for $\alpha \in [0,1]$, where v_1 and v_2 are the normal points of μ_1 and μ_2.

Formally speaking, we consider the following distance:

$$d(\mu, \mu') = \int_0^1 \delta(\mu_\alpha^A, \mu_\alpha'^A) d\alpha \tag{3}$$

where δ is a distance over crisp sets, $A = \{x|\alpha v_1 + (1-\alpha)v_2, \alpha \in [0,1]\}$ and μ_α^A is the α-cut of μ restricted to the set A

And, more specifically, we use as $\delta(C_1, C_2)$ as the euclidean distance between the nearest points in the crisp sets C_1 and C_2.

IV. Hierarchical fuzzy clustering

Given a set of elements X, we have applied a mixed approach to build a fuzzy hierarchical structure. The process starts building a fuzzy partition of X applying fuzzy c-means. This results into a set of fuzzy membership functions μ_i, each one built on the centroid v_i. This fuzzy partition bootstraps the process.

Then, the iterative process is applied to build the hierarchical clustering following a bottom-up strategy. At each step of the process, we start with a fuzzy partition of X represented by a set of membership functions μ_i. Such set of membership functions is partitioned using a partitive clustering method for fuzzy sets. Such partitive clustering method returns a new fuzzy partition μ_i' that is used as the starting point of the new step.

This is represented in the next algorithm:

Algorithm 3 Bottom-up algorithm

```
Algorithm BottomUp (X: data) returns fuzzyHierarchy
begin
  P := fuzzy partition (X);
  while not the root(P) do
      P' := fuzzy partition of P;
      link P' and P;
      set P := P';
  end while
  return(P);
end
```

In this algorithm, we use the fuzzy c-means algorithm for building the initial fuzzy partition. Such fuzzy partition is obtained by applying the fuzzy c-means algorithm to X. In this case, the algorithm is applied with a large number of clusters (*i.e.*, c is large). This selection of c is to have a large number of leaves in the fuzzy hierarchy.

In the iterative process, we use a fuzzy c-means based clustering method. Given a set of fuzzy sets, we build a fuzzy partition following the scheme of the fuzzy c-means described in Algorithm 2.

Differences consist on the way the distance $||x_k - v_i||$ is computed. Note that here, x_k and v_i represent fuzzy sets. More specifically, x_k stands for the k-th fuzzy set to be partitioned and v_i is one of the fuzzy sets in the new partition. Accordingly, $||x_k - v_i||$ is a distance between fuzzy sets. We use here, the distance defined in Equation 3.

Following the standard approach in fuzzy c-means, the fuzzy membership of a fuzzy set with centroid v is defined considering all other centroids v_i. In our case, the membership of the fuzzy set with centroid x_k is computed for all x taking into account all other centroids x_j as follows:

$$\mu_{x_k}(x) = \Big(\sum_{j=1}^{c} \Big(\frac{d(x_k, x)^2}{d(x_k, x_j)^2} \Big)^{\frac{1}{m-1}} \Big)^{-1}$$

Similarly, the membership of the fuzzy set with centroid v_i is computed for all x taking into account all other centroids v_j as follows:

$$\mu_{v_k}(x) = \Big(\sum_{j=1}^{c'} \Big(\frac{d(v_k, x)^2}{d(v_k, v_j)^2} \Big)^{\frac{1}{m-1}} \Big)^{-1}$$

Note that here, x_j are the centroids of the fuzzy sets being clustered and v_j are the centroids of the clusters we are constructing with the fuzzy c-means. Similarly, c is the number of centroids x_j and c' is the number of centroids in v_j.

Then, the distance between a fuzzy set with centroid x_k and another with centroid v_i will be computed using Equation 3 taking into account that the distance is restricted to the set $A = \{x|\alpha v_i + (1-\alpha)x_k, \alpha \in [0,1]\}$.

Another element to be taken into account is how to compute the new centroid, once the membership are known. This is, how to determine the new v_i. For fuzzy c-means, this is done as follows:

$$v_i = \frac{\sum_{k=1} n(u_{ik})^m x_k}{\sum_{k=1}^n (u_{ik})^m}$$

Here, we apply the same expression, using as x_k the centroids of the clusters being clustered and u_{ik} the corresponding membership values. Although we could use at this point the fuzzy extension to aggregate the fuzzy clusters (taking x_k as the fuzzy cluster itself instead of its centroid) we avoid such operation as that would increase the fuzziness of the whole result, and at the same time that would imply that v_i is a fuzzy centroid instead of a crisp one.

Note that this approach leads to different membership values than the one described in Section I-A. In particular, in the new approach it is possible that the membership of x to a cluster μ is smaller than the membership of x to a subcluster μ_j of μ.

This situation was not possible in our previous approach. In that case, $\mu_j(x) = \mu(x) \cdot \mu'_i(x)$ and, therefore, $\mu_i(x) \leq \mu(x)$ for all μ_i subset of μ.

V. Experiments

We have tested our approach with the IRIS database (from the UCI repository [9]). This database contains 3 sets of 50 instances (records), each set corresponding to a different class of the iris plant. Each record is described in terms of 4 numerical attributes. Instead, the class is categorical. For testing the algorithm, we have considered the records as having (4 + 1) variables, where the last one is the class (codified as 1, 2 or 3).

The algorithm has been applied to this data, starting with a partition of the original set into 30 fuzzy clusters. This corresponds to a standard application of the fuzzy c-means. Then, only one step of the iteration process has been applied, where the set of fuzzy sets is partitioned into 3 clusters. These fuzzy clusters are given in Table V. These fuzzy sets are a fuzzy partition of the fuzzy sets described in Table V. In both cases, only the centroids of the sets are given. This process has been computed with a value of $m = 1.3$.

The experiments show that only small values of m can be used in the second step. Otherwise, the partition of the set of fuzzy sets (the iterative process) tend to lead to fuzzy sets with very large fuzziness. Only for values of m near to 1, we obtain different centroids in the second step. For larger values of m, all centroids are exactly the same, thus all fuzzy sets having the same membership function in the whole domain (*i.e.*, $\mu_i(x) = 1/c$ for all x and where c is the number of clusters).

Besides of that, the experiments have confirmed that there is a neeed of large computational power to compute the fuzzy partition of the fuzzy sets. This is due to the need of computing the similarity between all pairs of fuzzy sets in each step of the iterative process within the fuzzy c-means.

VI. Conclusions

In this paper we have described our approach for building a fuzzy hierarchy of clusters and given an example of their application.

The algorithm presented in this paper permits the construction of a fuzzy hierarchy in a simpler way than our previous approach. The method requires less parameters and obtains a fuzzy partition without further manipulation of the fuzzy memberships obtained by the clustering techniques.

As future work, we include the application of this approach to information access and privacy preserving data mining. On the one hand, we plan to extend our GAMBAL system using the algorithm described here. On the other hand, we plan to test whether the approach presented here is suitable for preserving numerical databases. To do so, we will compare the results obtained with the ones presented in [10].

In the development of the applications, we will also consider alternative fuzzy clustering methods. In particular, we plan to consider variable size fuzzy clustering.

TABLE I

Centroids of fuzzy clusters

i-th	v_1	v_2	v_3	v_4	v_5
1	5.40	2.50	3.97	1.33	2.01
2	7.67	3.75	6.45	2.18	2.99
3	7.27	2.98	6.02	1.87	2.99
4	6.64	3.09	5.26	2.16	2.99
5	4.84	3.07	1.48	0.17	1.00
6	6.14	2.59	4.99	1.55	2.00
7	6.24	2.25	4.44	1.39	2.00
8	4.58	3.30	1.29	0.21	1.00
9	6.54	2.98	4.46	1.39	2.00
10	4.40	2.80	1.27	0.20	1.00
11	5.57	2.52	3.70	1.08	2.00
12	5.54	2.89	4.48	1.41	2.01
13	7.66	2.82	6.66	2.15	2.99
14	6.81	3.04	4.82	1.50	2.00
15	6.07	2.90	4.62	1.37	2.00
16	6.06	3.29	4.66	1.66	2.00
17	4.96	3.41	1.68	0.38	1.00
18	4.95	2.35	3.30	1.00	2.00
19	5.10	2.50	3.00	1.10	2.00
20	5.83	2.56	3.99	1.10	2.00
21	6.60	3.23	5.73	2.33	2.99
22	5.69	2.68	5.00	1.98	2.99
23	6.15	2.76	4.94	1.74	2.99
24	6.40	2.81	5.55	1.86	2.99
25	4.99	2.00	3.50	1.00	2.00
26	5.64	4.13	1.40	0.27	1.00
27	5.65	2.86	4.16	1.27	2.00
28	5.25	3.80	1.55	0.27	1.00
29	6.07	2.89	4.16	1.36	2.00
30	5.14	3.45	1.43	0.23	1.00

TABLE II

Centroids of the clusters obtained from Table V

i-th	v_1	v_2	v_3	v_4	v_5
1	5.09	3.34	2.07	0.56	1.21
2	6.70	3.03	5.46	1.87	2.68
3	5.42	2.90	3.12	0.92	1.70

Acknowledgment

This work was partly funded by the Spanish Ministry of Science and Technology TIC2001-0633-C03-03 ”STREAMOBILE” and also by the Spanish Ministry of Education and Science under project SEG2004-04352-C04-03 ”PROPRIETAS”.

References

[1] J. C. Bezdek, (1981), Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.

[2] I. Bloch, (1999), On fuzzy distances and their use in image processing under imprecision, Pattern Recognition, 32 1873-1895.

[3] A. Devillez, P. Billaudel, G. Villermain Lecolier, (2002), A fuzzy hybrid hierarchical clustering method with a new criterion able to find the optimal partition, Fuzzy Sets and Systems, 128 323-338.

[4] H. Ichihashi, K. Honda, N. Tani, (2000), Gaussian mixture PDF approximation and fuzzy c-means clustering with entropy regularization, Proc. of the 4th Asian Fuzzy System Symposium, May 31-June 3, Tsukuba, Japan, 217–221.

[5] S. Miyamoto, M. Mukaidono, (1997), Fuzzy c - means as a regularization and maximum entropy approach, Proc. of the 7th International Fuzzy Systems Association World Congress (IFSA'97), June 25-30, Prague, Chech, Vol.II 86–92.

[6] S. Miyamoto, (1999), Introduction to fuzzy clustering, Ed. Morikita, Japan.

[7] S. Miyamoto, K. Umayahara, (2000), Fuzzy c-means with variables for cluster sizes, 16th Fuzzy System Symposium, Akita, Sept.6-8, 537–538 (in Japanese).
[8] S. Miyamoto, K. Umayahara, (2000), Methods in Hard and Fuzzy Clustering, pp 85–129 in Z.-Q. Liu, S. Miyamoto (Eds.), Soft Computing and Human-Centered Machines, Springer-Tokyo.
[9] P. M. Murphy, D. W. Aha, (1994), UCI Repository machine learning databases, http://www.ics.uci.edu/ mlearn/MLRepository.html, Irvine, CA: University of California, Department of Information and Computer Science.
[10] V. Torra, S. Miyamoto, Evaluating fuzzy clustering algorithms for microdata protection, Privacy in Statistical Databases, 2004. (Lecture Notes in Computer Science 3050 (2004), 175-186).
[11] V. Torra, S. Miyamoto, S. Lanau, Exploration of textual databases using a fuzzy hierarchical clustering algorithm in the GAMBAL system, Information Processing and Management, Information Processing and Management, in press.
[12] C. Vertan, N. Boujemaa, (2000), Using Fuzzy Histograms and Distances for Color Image Retrieval, CIR 2000.