Detecting and Verifying Dissimilar Patterns in Unlabelled Data

Manolis Wallace[1,2,⋆], Phivos Mylonas[3], and Stefanos Kollias[4]

[1] National Technical University of Athens: Image, Video and Multimedia Systems Laboratory wallace@image.ntua.gr
[2] University of Indianapolis: Athens Campus
[3] National Technical University of Athens: Image, Video and Multimedia Systems Laboratory fmylonas@image.ntua.gr
[4] National Technical University of Athens: Image, Video and Multimedia Systems Laboratory stefanos@cs.ntua.gr

Clustering of unlabelled data is a difficult problem with numerous applications in various fields. When input space dimensions are many, the number of distinct patterns in the data is not known a priori, and feature scales are different, then the problem becomes much harder. In this paper we deal with such a problem. Our approach is based on an extension to hierarchical clustering that makes it suitable for data sets with numerous independent features. The results of this initial clustering are refined via a reclassification step. The issue of evaluation of hierarchical clustering methods is also discussed. The performance of the proposed methodology is demonstrated through the application to a synthetic data set and verified through application to a variety of well known machine learning data sets.

1 Introduction

Clustering of data is an problem that is related to numerous scientific and applied fields [6]. Although researchers in the field of *data mining* have worked in this direction for long, and numerous related texts exist in the literature, it is still considered an open issue, as it is difficult to handle in the cases that the data is characterized by numerous measurable features. This is often referred to as the *dimensionality curse*.

Works in the field of *classification* focus in the usage of labelled (characterized) data, also known as *training data*, for the automatic generation of systems that are able to classify (characterize) future data. This classification relies on the similarity of incoming data to the training data. The main aim is

⋆ Corresponding author

to automatically generate systems that are able to correctly classify incoming data [6].

Typically, in order to pursue such a task, one first needs to detect the *patterns* that underly in the data, and then study the way these patterns relate to meaningful classes. Even when using self - training systems, such as *resource allocating neural networks*, that are able to adapt themselves to the training data, good results may only be achieved when the patterns are known before hand, so that they may be used for proper initialization [5].

Although the tasks of classification and clustering are closely related, an important difference exists among them. While in the task of classification the most important part is the distinction between classes, i.e. the detection of class boundaries, in the task of clustering the most important part is the identification of cluster characteristics. The latter is usually tackled via the selection of cluster representatives and cluster centroids, or via the extraction of (fuzzy) rules [6].

Efficient solutions have been proposed in the literature for both tasks, for the case in which a unique similarity or dissimilarity measure is defined among input data elements [11]. When, on the other hand, multiple independent features characterize data, and thus more than one meaningful similarity or dissimilarity measures can be defined, both tasks become more difficult to handle. A common approach to the problem is the lowering of input dimensions. This may be accomplished by ignoring some of the available features (*feature selection*) [7], or by applying some space transformation [3].

In the case when input features are not independent from each other, a decrease of dimensions is very helpful. On the other hand, when input features are independent, or when the relation among them is not known a priori, which is often the case with real data, a decrease of space dimensions cannot be accomplished without loss of information. Therefore, if the relation among features is not known before hand, and the aim is to detect the patterns that exist in the data, the decrease of dimensions is not possible. Moreover, the differences in measurement scale among different features also tend to disrupt the process of clustering. The difficult problem of initial analysis of data as to properly re-scale features and select which ones to use in the process of clustering is known as *data pre-processing*.

In this work we attempt to tackle detection of patterns in multi – dimensional data that have not been pre-processed, when the count of distinct patterns in the data and the relation among input features are unknown. The proposed algorithm is an extension of agglomerative clustering and is based on a soft selection of features to consider when comparing data. The results of this initial clustering are refined via a reclassification step; this step, although unsupervised, is based on the principles of the Bayes classifier. This step also contributes to the experimental evaluation of the method's efficiency.

The structure of the paper is as follows: in section 2, after a short introduction to agglomerative clustering, we present the main problems that are related to our task. In section 3, we present the proposed method for initial

clustering and in section 4 we explain how a Bayes – based classifier can be used to refine, as well as to experimentally verify the efficiency of the algorithm. Finally, in section 5, we present experimental results for the proposed algorithm and in section 6, we present our concluding remarks.

2 Agglomerative Clustering and Related Problems

Most clustering methods belong to either of two general methods, partitioning and hierarchical. Partitioning methods create a crisp or fuzzy clustering of a given data set, but require the number of clusters as input. When the count of patterns that exist in a data set is not known beforehand, partitioning methods are inapplicable; an hierarchical clustering algorithm needs to be applied.

Hierarchical methods are divided into agglomerative and divisive. Of those, the first are the most widely studied and applied, as well as the most robust. Their general structure is as follows [9]:

1. Turn each input element into a singleton, i.e. into a cluster of a single element.
2. For each pair of clusters c_1, c_2 calculate a compatibility indicator $CI(c_1, c_2)$. The CI is also referred to as cluster similarity, or dissimilarity, measure.
3. Merge the pair of clusters that have the best CI. Depending on whether this is a similarity or a dissimilarity measure, the best indicator could be the maximum or the minimum operator, respectively.
4. Continue at step 2, until the termination criterion is satisfied. The termination criterion most commonly used is the definition of a threshold for the value of the best compatibility indicator.

The two key points that differentiate agglomerative methods from one another, and determine their efficiency, are the compatibility indicator and the termination criterion used. Major drawbacks of agglomerative methods are their high complexity and their susceptibility to errors in the initial steps, that propagate all the way to their final output.

The core of the above generic algorithm is the ability to define a unique compatibility indicator among any pair of clusters. Therefore, when the input space has more than one dimensions, an aggregating distance function, such as Euclidean distance, is typically used as the CI [14]. This, of course, is not always meaningful. Cases exist, in which the "context" can change the similarity or dissimilarity measure to be used [13]. For example, two films may be compared based on their topic, or on their directors.

In such cases, a selection of meaningful features needs to be performed, prior to calculating a CI. In the example of films, although two films may be similar to one another as far as their content is concerned, two other films may be similar as far as their cast is concerned. In other words, it may not be possible to select a single distance metric, which will apply in all cases, for a

given data set. Moreover, one feature might be more important than others, while all of the features are useful, each one to its own degree. In other words, hard (crisp) feature selection is not always possible, either.

3 Soft Feature Selection and Clustering in Multi – Dimensional Spaces

Elements are usually grouped together based on their similarity in a single or a few features. When the total number of features is high, small distances in a small subset of them barely affect the overall distance, when an aggregation of distances in all features is used. Thus, only when the correct subset of features is considered, can elements be compared correctly [10],[4].

In this paper we tackle feature selection based on the following principle: while we expect elements of a given meaningful set to have random distances from one another according to most features, we expect them to have small distances according to the features that relate them. We rely on this difference in distribution of distance values in order to identify the *context* of a set of elements, i.e. the subspace in which the set is best defined.

More formally, let c_1 and c_2 be two clusters of elements. Let also r_i, $i \in N_F$ be the metric that compares the i-th feature, and F the overall count of features (the dimension of the input space). A distance (dissimilarity) measure between the two clusters, when considering just the i-th feature, is given by

$$f_i(c_1, c_2) = \sqrt[\kappa]{\frac{\sum_{a \in c_1, b \in c_2} r_i(a_i, b_i)^\kappa}{|c_1||c_2|}} \tag{1}$$

where e_i is the i-th feature of element e, $|c|$ is the cardinality of cluster c and $\kappa \in R$ is a constant.

The context is a soft selection of features to consider when calculating an overall distance value. We can define it as a fuzzy set x defined on N_F, with a scalar cardinality of one. Then, the overall distance between c_1 and c_2 is calculated as

$$d(c_1, c_2) = \sum_{i \in N_F} x_i(c_1, c_2)^\lambda \cdot f_i(c_1, c_2) \tag{2}$$

where x_i is the degree to which i, and therefore f_i, is included in the context, $i \in N_F$ and $\lambda \in R$ is a constant.

According to the principle presented in the beginning of this paragraph, the features that relate c_1 and c_2 are the ones that produce the smallest distances f_i. Therefore, the "correct" context can be calculated through the solution of an optimization problem, as the context that produces the smallest overall distance.

When $\lambda = 1$ the solution is trivial: the feature that produces the smallest distance is the only one selected. The degree to which it is selected is 1. If more

than one features produce the best distance, then they are equally selected, as there is no information as to which should be favored.

When $\lambda \neq 1$ and $\exists i \in N_F : f_i(c_1, c_2) = 0$, then the features for which $f_i(c_1, c_2) = 0$ are the ones the are (equally) selected.

When $\lambda \neq 1$ and $f_i(c_1, c_2) \neq 0 \forall i \in N_F$, then the optimization problem is not trivial and has to be solved. According to the following lemma it can be solved analytically, which means that the optimization problem does not affect the algorithmic complexity of the process:

Lemma 1. *When $\lambda \neq 1$ and $f_i(c_1, c_2) \neq 0 \forall i \in N_F$, then the best context x, and equivalently the best compatibility indicator CI, is given by:*

$$x_F(c_1, c_2) = \frac{1}{\sum_{i \in N_F} [\frac{f_F(c_1, c_2)}{f_i(c_1, c_2)}]^{\frac{1}{\lambda - 1}}} \tag{3}$$

$$x_i(c_1, c_2) = x_F(c_1, c_2) \cdot [\frac{f_F(c_1, c_2)}{f_i(c_1, c_2)}]^{\frac{1}{\lambda - 1}} \tag{4}$$

Proof. We have demanded that the scalar cardinality of the context is one; the optimization problem we have to tackle is constrained. $|x| = 1$ is equivalent to $\sum_{i \in N_F} x_i = 1$. Thus, replacing

$$x_F = 1 - \sum_{i \in N_{F-1}} x_i \tag{5}$$

the minimization of

$$d(c_1, c_2) = x_F(c_1, c_2)^\lambda \cdot f_F(c_1, c_2) + \sum_{i \in N_{F-1}} x_i(c_1, c_2)^\lambda \cdot f_i(c_1, c_2)$$

is reduced to an unconstrained optimization problem. From 5 we have

$$\frac{\partial x_F(c_1, c_2)}{\partial x_i(c_1, c_2} = -1 \forall i \in N_{F-1}$$

and thus

$$\frac{\partial \{x_F(c_1, c_2)^\lambda \cdot f_F(c_1, c_2)\}}{\partial x_i(c_1, c_2)} = -\lambda \cdot x_F(c_1, c_2)^{\lambda - 1} \forall i \in N_{F-1}$$

Easily now, demanding that

$$\frac{\partial d(c_1, c_2)}{\partial x_i(c_1, c_2)} = 0 \forall i \in N_F$$

we have

$$x_i(c_1, c_2) = x_F(c_1, c_2) \cdot [\frac{f_N(c_1, c_2)}{f_i(c_1, c_2)}]^{1 - \lambda} \forall i \in N_{F-1} \tag{6}$$

Combining 6 with 5 we also have

$$x_F(c_1, c_2) = \frac{1}{\sum_{i \in N_F} [\frac{f_F(c_1,c_2)}{f_i(c_1,c_2)}]^{\frac{1}{\lambda-1}}}$$

QED

As λ increases, pairs of clusters that are related by fewer features, and thus have greater values in their contexts, are obviously assigned smaller distances. In order for distances to be usable as compatibility indicators, they need to be unaffected by cluster direction in comparison to the axes. Thus, it is imperative that they are transformed, as to become directly comparable to each other. The following, adjusted, compatibility indicator is used:

$$CI(c_1, c_2) = \frac{d(c_1, c_2)}{x_\lambda(c_1, c_2)} \tag{7}$$

$$x_\lambda(c_1, c_2) = \sum_{i \in N_F} [x_i(c_1, c_2)]^\lambda \tag{8}$$

When features are quantized to a small set of levels, as is often the case with digital data, cases for which $f_i(c_1, c_2) = 0$ are not rare. Especially in the first steps of agglomerative clustering, when clusters are of small size, the best CIs are almost always zero. Since, as we have already mentioned, errors in the initial steps of agglomerative clustering propagate all the way to the final output, it is important to always make the best selection possible for the pair of clusters to merge. Therefore, especially for the case of CIs that are equal to zero, we introduce one more criterion: out of all the pairs for which $CI = 0$, the one that has zero distances for the most features will be selected. In other words, out of all the pairs of similar clusters, the ones that are similar according to the greatest number of features are selected.

As far as the termination criterion is concerned, a threshold on the value of CI can be used, as Lemma 2 guarantees that this is meaningful.

Lemma 2. *The above mentioned CI is non decreasing as we move from one step to the next.*

Proof. Proof is trivial and is omitted for the sake of space.

This way, the algorithm gradually groups elements together, based on their similarities; for each cluster, a different fuzzy subset of features may be considered for the calculation of similarities. This soft feature selection may also be perceived of as a re-scaling of features, thus making up for the skipped step of data pre-processing.

The average values of features for each cluster form the centroid, i.e. a "virtual" element that is located at the center of the cluster, when all of its elements are placed in the F-dimensional space. Its position may be considered as a description of the feature values of the pattern that this cluster

corresponds to. The variances of the values for each feature indicate the importance of each feature for the definition of the cluster; this may be perceived of as an estimation of the radius of the cluster in the direction of each feature.

Assuming that the clustering has produced meaningful groups of elements, the latter may be used for the initialization of an adaptive neural classifier; the fact that clusters are described through center and variance combinations makes the output ideal for the initialization of RBF based networks [12].

4 Refinement and Verification through Bayesian Classification

As stated in section 1, the primary aim of clustering algorithms is not to correctly classify data, but rather to identify the patterns that underly in it. Therefore, 'wrong' elements in clusters may be acceptable, as long as the overall cluster correctly describes an existing and meaningful pattern. This implies that feeding labelled data to the algorithm and measuring the classification rate may not be enough to evaluate the actual efficiency of the algorithm.

In order for a clustering algorithm to be truly evaluated, the patterns that are described by the detected clusters need to be extracted and examined. In this work we examine whether detected patterns are meaningful by evaluating a classifier that is created by using them. Out of the numerous classification schemes that exist in the literature we have chosen to work with the Bayesian classifier, although others could have been chosen as well [8].

Specifically, each cluster is considered to describe a distinct class. Furthermore, we assume that all features of members of a class follow a gaussian distribution. Thus, using the centroid and standard deviations of each cluster, we may design the mixture of Gaussians that describe the class. The Bayes classification scheme calculates for each input element a the probabilities $P(p_i/a)$, $i \in N_T$, where T is the count of detected patterns, and classifies a to the pattern for which it has the greatest probability. Probabilities are easily computed by applying the transformation:

$$P(p_i/a) = P(a/p_i)P(p_i)$$

where $P(p_i)$ is equal to the relative cardinality of cluster c_i, i.e.

$$P(p_i) = \frac{|c_i|}{\sum_{j \in N_P} |c_j|}$$

and $P(a/p_i)$ is given as the value of a in the mixture of Gaussians that describe pattern i, i.e.

$$P(a/p_i) = \prod_{j \in N_F} \frac{1}{\sqrt{2\pi} s_{ij}} e^{-(\frac{a_j - m_{ij}}{2 s_{ij}})^2}$$

Table 1. The parameters for the generation of the synthetic data set

class	m_1	s_1	m_2	s_2	elements
A	2	0.5	1	0.1	100
B	1	0.9	3	0.1	100
C	1	0.1	2	0.7	100

where m_{ij} and s_{ij} are the centroid value and standard deviation for the j-th feature of pattern i and a_j is the j-th feature of element a.

Using this scheme, we may reclassify all data that were used for clustering. If the clustering was successful, i.e. if the detected patterns are meaningful, then this process will refine the classification rate by removing some of the clusters' members that were a result of errors in the initial steps. Thus, this process offers an indication of the clustering's true performance. Moreover, it makes the overall algorithm more robust, as opposed to simple hierarchical clustering, as it is more resilient to errors in the initial steps.

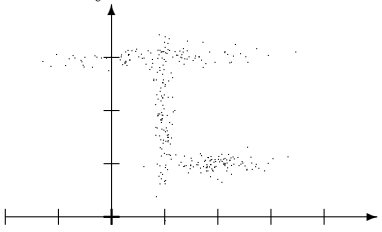
5 Experimental Results

In this section we list some indicative experimental results of the proposed methodology. In subsection 5.1 we provide an example of application to a simple synthetic data set, which facilitates the visualization of the algorithm's performance. Continuing, in subsection 5.2 we list results from application to real data sets from the machine learning databases.

5.1 Synthetic Data

In order for the visualization of the synthetic data set to be feasible, we have limited it to two dimensions. Three classes of data were created, using a gaussian random generator. The parameters of the gaussian distributions used for the generation of the data set are presented in Table 1.

The synthetic data set.



As can be seen from the Table, as well as from the diagram, the three classes are not clearly distinguished from each other, and the subspaces that best characterize each class differ to a great extent. This makes distance aggregation – based approaches inefficient; this is verified in Table 3. The initial classification step produces a classification rate of 91% (assigning each cluster to the class that dominates it), and the reclassification refines this to 95.7%, indicating that the initial step, although has a smaller rate, has correctly identified the underlying patterns.

Table 2. The clusters produced, for the synthetic data set. Format: (class 1, class 2, class 3)

Method	cluster 1	cluster 2	cluster 3	Classification rate
Euclidian clus.	(0,0,13)	(28,0,87)	(72,100,0)	66.7%
Initial clus.	(4,0,86)	(92,5,8)	(4,95,6)	91%
Bayesian reclass.	(6,0,98)	(94,5,2)	(0,95,0)	95.7%

5.2 Real Data

Iris data

The iris data set contains 150 elements, characterized by 4 features, that belong to three classes; two of these classes are not linearly separable from each other. This is a relatively easy data set, as the number of clusters in the data is equal to the number of classes. The labels of the elements were not used during clustering and reclassification; there were used, though, for evaluation purposes. Results are shown in Table 3.

The considerable refinement that a single step of Bayesian reclassification offers is indicative of the validity of the detected clusters. This observation is supported even more by the fact that recursive application of the reclassification step refines even more the results, even though this step is unsupervised (it does not use element labels).

Wisconsin Breast Cancer Database

The Wisconsin breast cancer database contains 699 elements, which are characterized by the following attributes: clump thickness, uniformity of cell size,

Table 3. Classification rates for iris data ($\kappa = \lambda = 2$)

Method	cluster 1	cluster 2	cluster 3	Classification rate
Initial clus.	(36,4,12)	(13,0,38)	(1,46,0)	80%
Bayesian reclass. 1	(33,0,2)	(17,0,48)	(0,50,0)	87.3%
Bayesian reclass. 2	(35,0,0)	(15,0,48)	(0,50,0)	90%

Table 4. Classification rates for Wisconsin data ($\kappa = \lambda = 2$)

Method	cluster 1	cluster 2	cluster 3	Classification rate
Initial clus.	(31,42)	(3,136)	(410,61)	86.1%
Bayesian reclass.	(5,18)	(2,177)	(437,44)	92.5%

Table 5. Classification rates for Wisconsin data ($\kappa = \lambda = 5$)

Method	cluster 1	cluster 2	cluster 3	Classification rate
Initial clus.	(192,56)	(3,154)	(249,29)	87.1%
Bayesian reclass.	(0,0)	(10,218)	(434,21)	95.5%

uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, mitoses. All these attributes assume integer values in $[1, 10]$. Elements are also accompanied by an id, and class information; possible classes are benign and malignant. 65.5% of the elements belong to the benign class and 34.5% to the malignant class. 16 elements are incomplete (an attribute is missing) and have been excluded from the database for the application of our algorithm.

This data set, having a greater number of features, is considered to be more difficult than the iris data set. Detailed results acquired using the proposed methodology are available in Tables 4 and 5. It is worth noting that, although the classification rate of the initial clustering procedure are not extremely high, the reclassification step refines it considerably. Furthermore, for the case where $\kappa = \lambda = 5$, the reclassification step classifies every element to one of exactly two clusters, each one almost totally dominated by one class.

This performance is not far from that of trained classification systems that utilize the same dataset; a classification rate of 97% is reported in [2]. This is indicative of the method's efficiency, considering that we are referring to the comparison of an unsupervised method to a supervised one.

Ionosphere Database

This radar data was collected by a system in Goose Bay, Labrador. The targets were free electrons in the ionosphere. "Good" radar returns are those showing evidence of some type of structure in the ionosphere. "Bad" returns are those that do not. Elements of the data set are characterized by 34 features and classified as either good or bad. Results from the application of the proposed methodology appear in Table 6.

Considering that supervised classification algorithms report a classification rate of around 90%, it is easy to conclude that the initial clustering is extremely efficient. If we also consider that unsupervised clustering methods do not exceed a classification rate of 80% for 10 clusters [1], then we might conclude that the detection of two clusters with a classification rate of 87.2% is extremely successful.

Table 6. Classification rates for Ionosphere database ($\kappa = \lambda = 2$)

Number of clusters	Initial clust.	Bayesian reclass.
2	87.2%	80%
3	87.2%	80.1%
10	87.2%	84.9%
15	87.2%	87.2%
20	87.2%	87.7%
25	87.2%	91.2%

The step of Bayesian reclassification, in addition to refining the clustering, as to reach a classification rate of 91.2% for 25 clusters, also discriminates meaningful from random output. Thus, it is easy to see that although the same classification rate is reported, a partitioning with less than 10 clusters is not able to provide efficient classifier initialization, as it does not adequately describe the underlying patterns, while a partitioning of 25 clusters would be much more effective.

6 Conclusions

In this paper we developed an algorithm for the detection of patterns in unlabelled data. The first step of the algorithm consists of an hierarchical clustering process. This process performs a soft feature selection in order to determine the subspace within which a set of elements is best defined. Thus, it is suitable for data sets that are characterized by high dimensionality. The second part of the algorithm is a Bayesian classification. This process considers initial clusters to be labels and uses this information to build a classifier, through which to reclassify all data. Thus, errors from the hierarchical algorithms initial steps are corrected. In addition to making the overall algorithm more efficient and resilient to errors, it also serves as a means for its evaluation.

The efficiency of the proposed algorithm as a whole, as well as of its distinct steps independently, has been demonstrated through application s to a variety of synthetic and real data sets. Within them was the Wisconsin breast cancer database which is a multi – dimensional data set; the algorithm performs remarkably well for it. The ionosphere database was also considered, through which it was made obvious that the evaluation of the performance of a clustering method is imperative, before its output is further used for other tasks of data processing.

The fact that the proposed methodology performs a soft feature selection makes it able to handle input data with features of different scales. Thus, it may be used to substitute the phase of data pre-processing. Moreover, the representation of clusters using a mixture of Gaussians is compatible with the internal representation of RBF nodes, which makes our method ideal for the initialization of RBF based neural networks.

References

1. Aggarwal, C.C., Yu, P.S. (2002) Redefining clustering for High–Dimensional Applications. IEEE Transactions on Knowledge and Data Engineering 14(2):210–225
2. Bagui, S.C., Bagui, S., Pal, K., Pal, N.R. (2003) Breast cancer detection using rank nearest neighbor classification rules. Pattern Recognition 36:25–34
3. Brunzell, H., Eriksson, J. (2000) Feature reduction for classi"cation of multidimensional data. Pattern Recognition 33: 1741–1748
4. Dong M., Ravi Kothari, R. (2003) Feature subset selection using a new definition of classifiability. Pattern Recognition Letters 24:1215–1225
5. Haykin, S. (1999) Neural Networks: A Comprehensive Foundation, 2nd edition. Prentice Hall
6. Hirota, K., Pedrycz, W. (1999) Fuzzy computing for data mining. Proceedings of the IEEE 87:1575–1600
7. Kohavi, R., Sommerfield, D. (1995) Feature Subset Selection Using the Wrapper Model: Overfitting and Dynamic Search Space Topology. Proceedings of KDD-95
8. Lim, T.-S., Loh, W.-Y., Shih, Y.-S. (2000) A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-three Old and New Classification Algorithms. Machine Learning 40:203–229
9. Miyamoto, S. (1990) Fuzzy Sets in Information Retrieval and Cluster Analysis. Kluwer Academic Publishers
10. Swiniarski, R.W., Skowron, A. (2003) Rough set methods in feature selection and recognition. Pattern Recognition Letters 24:833–849
11. Theodoridis, S. and Koutroumbas, K. (1998) Pattern Recognition, Academic Press
12. Tsapatsoulis, N., Wallace, M. and Kasderidis, S. (2003) Improving the Performance of Resource Allocation Networks through Hierarchical Clustering of High – Dimensional Data. Proceedings of the International Conference on Artificial Neural Networks (ICANN), Istanbul, Turkey
13. Wallace, M., Stamou, G. (2002) Towards a Context Aware Mining of User Interests for Consumption of Multimedia Documents. Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), Lausanne, Switzerland
14. Yager, R.R. (2000) Intelligent control of the hierarchical agglomerative clustering process. IEEE Transactions on Systems, Man and Cybernetics, Part B 30(6): 835–845