Class Segmentation to Improve Fuzzy Prototype Construction: Visualization and Characterization of Non Homogeneous Classes

Jason Forest, Maria Rifqi, Bernadette Bouchon-Meunier

2006 IEEE International Conference on Fuzzy Systems
Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada
July 16-21, 2006

Class Segmentation to Improve Fuzzy Prototype Construction: Visualization and Characterization of Non Homogeneous Classes

Jason Forest, Maria Rifqi and Bernadette Bouchon-Meunier

Abstract— **In this paper, we present a new method to construct fuzzy prototypes of heterogeneous classes, in a supervised learning context. Heterogeneous classes are classes where the coexistence of far behaviours can be observed. Our approach consists in two stages. The first one enables to discover, in an original method, the different behaviours within a class by decomposing it in subclasses. In the second stage, we construct a fuzzy prototype for each subclass by using typicality degrees. Thanks to this decomposition of a class and to this characterization of typical behaviours, we propose an intuitive summarization of a class. We illustrate the advantages of our method on both artificial and real dataset.**

I. Introduction

In the context of supervised learning the aim is to discover a function enabling to label correctly an unseen example. This function can take different forms: decision tree, neural network, set of rules,... Depending on the particular form, the discovered function can give an explicit summarization of the data by giving important descriptors explaining the class.

In this paper, we are particularly interested in the case where classes are not homogeneous, i.e. where the coexistence of far behaviours can be observed. For example, it can be observed that pupils who succeed can be those who work a lot as well as those who do not need to provide so much effort. In this case, the success of a pupil can be explained in two different ways regarding the provided quantity of work. We propose to refine the description of a class by discovering automatically its different subclasses and by characterizing each one through a fuzzy prototype.

We mostly pay attention to provide an understandable summarization of a complex class. In order to keep this property, fuzzy prototype based approach ([1], [2]), is prioritized. Indeed, the concept of prototype, as it has been defined by Rosch [3], is natural for a classification task or for a summarization task.

After presenting related methods to supervised clustering as well as fuzzy prototypes, we describe our two stages approach: the first one enables to discover, in an original method, the different behaviours within a class by decomposing it in subclasses; in the second stage, we construct a fuzzy prototype for each subclass by using [1]'s approach.

Jason Forest is with Arvem France S.A., 6 Esplanade de la Gare, F-95510 Sannois and the Universite Pierre et Marie Curie - Paris6, CNRS UMR 7606, DAPA, LIP6 8, rue du Capitaine Scott, Paris, F-75015, France (Email: jason.forest@lip6.fr).

Maria Rifqi is with the Universite Pierre et Marie Curie - Paris6, CNRS UMR 7606, DAPA, LIP6 8, rue du Capitaine Scott, Paris, F-75015, France (Email: maria.rifqi@lip6.fr).

Bernadette Bouchon-Meunier is with the Universite Pierre et Marie Curie - Paris6, CNRS UMR 7606, DAPA, LIP6 8, rue du Capitaine Scott, Paris, F-75015, France (Email: bernadette.bouchon-meunier@lip6.fr).

II. Related Methods

A. Supervised Clustering Approaches

In supervised learning, a class is not often represented by a single group. However, supervised learning is based on the hypothesis that a link between the class membership and the spatial distribution exists. So, the idea of trying to find the subgroups of a class in order to make classifiers' work easier, has been proposed by several authors [4], [5].

In [4], the author proposes a method providing a class segmentation of a database. A k-means algorithm is used in order to obtain the segmentation. As this algorithm is randomly initialised and the parameter k must be given, the author makes two experiments: the first one with the parameter $k = 5$ and five trees classifiers learned on five different initialisations of the algorithm and a second one with $k = 3$ and, also, five classifiers. There is an improvement of the classification rate (four of five real databases).

In [5], a similar approach is applied to explain and improve the Naive Bayes classifiers. The authors use an Expectation Maximisation (EM) algorithm on each class to make a class segmentation. The number of clusters could be given in parameter but it could also be estimated by using a criteria like BIC or Akaike. According to this new segmentation, the authors construct Naive Bayes classifiers for 26 real datasets. For 15 datasets, the classification rate is improved and for three datasets the performances are worse.

In all these papers, the segmentation is made thanks to a clustering algorithm performed on each class separately. There are two problems with this strategy. Firstly, the estimation of the number of clusters, which is needed for these algorithms, is a difficult task. Secondly, one class is segmented independently of the other ones. It means that all the available information is not used. If the aim is to characterize the most typical behaviours of a class, the information given by the neighbourhood can enhance what is peculiar to this class.

B. Fuzzy Prototype Construction

A prototype is an object chosen to give a simple and understandable summarization of a group (or class). The concept of prototype relies on the notion of typicality: all individuals are not equally representative of their class. Typicality has been studied by Rosch [3]. She showed that the typicality of an element for a given category depends on two

0-7803-9489-5/06/$20.00/©2006 IEEE

factors: its resemblance to the other members of the category and its differences to the members of other categories.

Rifqi [2] proposed a method to implement these principles: it computes, for each individual, its internal resemblance, i.e. its average resemblance to the other points of the group, and its external dissimilarity, i.e. its average dissimilarity to individuals belonging to other categories. The typicality degree is then the aggregation of these two quantities.

$$Typicality = Agg(Resemblance, Dissimilarity)$$

The prototype is then defined as the aggregation of the most typical data: it highlights the common points of the category members but also their discriminative features. Rifqi [2] considers the case of fuzzy data.

Lesot et al. [1], [6] have extended Rifqi's approach to numerical data i.e. vectorial data belonging to $\mathbb{R}^p$. They propose to define the prototype as the fuzzy set whose kernel contains points having typicality higher than a threshold, and its support contains points with typicality higher than a second, smaller, threshold.

However, as it has been defined in these approaches, the typicality scores for the individuals of a non homogeneous class are low. Indeed, in the case where a class is spread, the examples can be far from each other. So, the resemblance is quite low and then the typicality is also low. This case is illustrated and studied in section IV.

III. Proposed Method

Our method is in two stages: identification of subclasses and construction of fuzzy prototype.

A. Identitfication of Subclasses

At the opposite of related methods described in section II, our method of identification of subclasses takes into account the examples of other classes, in order to highlight the specificity of a class.

We consider a class separated by examples belonging to another class. This situation is shown in the figure 1A.

To perform such a task, we propose an algorithm which automatically constructs weighted graphs with the data. A graph of an example provides its *friends*, i.e. examples from the same class that are close in such a way that there is no example from another class (*enemy*) that is closer. It means that the closest enemy of an example defines the limit of its friends neighbourhood. Edges, representing the proximity, are weighted by a distance between the two nodes (examples). The union of the graphs generated by each example is splitted in non connex graphs (figure 1D). A (connex) graph is then a subclass.

To obtain these graphs, we let a metasphere grow from an example of a class until it encounters an enemy (figure 1B and 1C). All the examples in this metasphere are linked to the central example. This process (construction of a metasphere) is repeated for all the examples.

Therefore, this stage of our method allows to discover subclasses identifying the specific behaviours of the class.

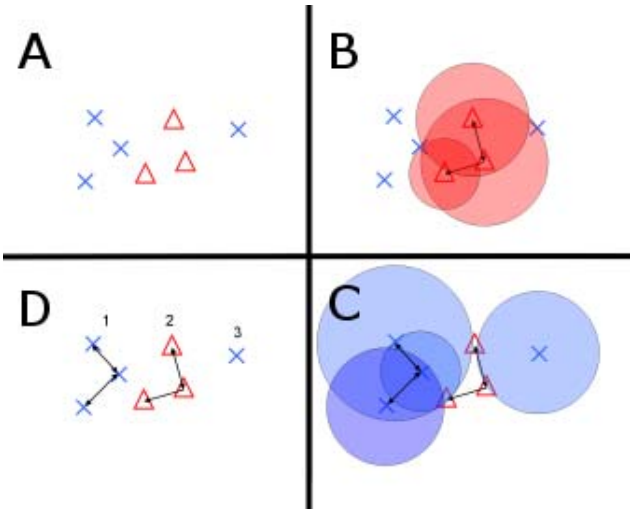

Fig. 1. Identification of subclasses: **A** *Database* **B** Metaspheres of class of red triangles **C** Metaspheres of class of blue crosses **D** Graphs of subclasses

B. Construction of prototypes

Secondly, we construct the prototypes. Thanks to the first part of the method, we have divided the classes into subclasses. By construction, our subclasses are compact groups. It is now possible to apply the construction of fuzzy prototypes describes in Lesot [1]. But instead of doing it on the class, the fuzzy prototypes are constructed for each subclass.

In Lesot [1], the fuzzy prototype is a set of typical examples whereas in our approach we consider a fuzzy prototype as a set of typical values. So, for each value of an attribute, we compute its *internal resemblance* and its *external dissimilarity* based on an Euclidian distance d, in order to find its typicality. For an object, u we note its set of friends (examples from the same subclass) $F = f_1, ..., f_n$, its set of ennemies (the rest of examples) $E = e_1, ..., e_m$ and its value of the attribute a, V_a^u. The resemblance measure of the value V_a^u is then the mean of all the resemblances:

$$R(V_a^u) = \frac{\sum_{i=1}^{n} \frac{1}{d(V\ ,V\ )}}{n}$$

The dissimilarity measure is given by:

$$D(V_a^u) = \frac{\sum_{i=1}^{m} \frac{1}{1-d(V\ ,V\ )}}{m}$$

Then we compute the typicality for each object and each attribute by aggregating the resemblance and the dissimilarity. We choose the symmetric sum [7], [8] as the aggregation operator because as it has been shown by Lesot [1] this operator has the full reinforcement property: the aggregation of high scores can give higher score.

$$Typi(V_a^u) = \frac{R(V_a^u).D(V_a^u)}{R(V_a^u).D(V_a^u) + (1 - R(V_a^u))(1 - D(V_a^u))}$$

These concepts are described in the figure 2: the typicality of the values $V1$ is an aggregation by the symmetric sum

of internal resemblances with V4, V5, V7 and external dissimilarity with V2, V3, V6.

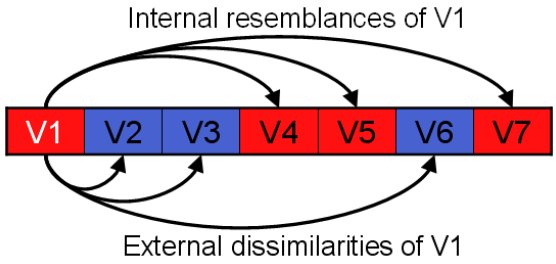

Fig. 2. The typicality for the value V1 is an aggregation of the resemblances with the values of the same category and the dissimilarity with the values of the other categories.

Finally, we draw for each attribute its typicality scores relative to its values. The power of our prototypes is the characterization and the visualization of the different attributes of the data. So, to simplify the representations, we choose to not take into account the very small subclasses. Indeed, the objects of a very small subclass are very dissimilar to the others and very similar to themselves. Consequently, the mean typicality of such subclass is very high (often maximal).

IV. Results

In this part we will show two experiments, a first one on artificial data and a second one on real data.

A. Power of the class segmentation

The goal of our first experiment is to show the improvements of the characterization of a class thanks to our method. We expect to highlight the two major behaviours of a class separated in two groups, one on each side of another class.

Without the segmentation process (figure 5), we obtain a single prototype with low degrees of typicality whereas with a segmentation we have two prototypes with significant typicality distributions.

We illustrated the advantage of our method on an artificial two dimensions database that contains 200 examples belonging to two classes. The first class (the red triangles) is composed of one group whereas the second class is made up of two distinct groups (the blue crosses) (see figure 3).

The first step is the class segmentation. The figure 4 shows the graphs obtained by this process. The blue class is divided in 19 subclasses with two of them having more than 3 examples. The red class is divided in 21 subclasses with one which has more than 3 examples. Our algorithm constructs three major subclasses for the database.

The figure 3 shows that the classes overlap. Our segmentation process eliminates these overlaps by making several small subclasses because the specificity of a class is prioritized. Besides, our algorithm can deal with noisy data. For instance, the triangles class examples which are in the middle of a subclass (the upper-right crosses group in figure 3) do not prevent the construction of a unique subclass.

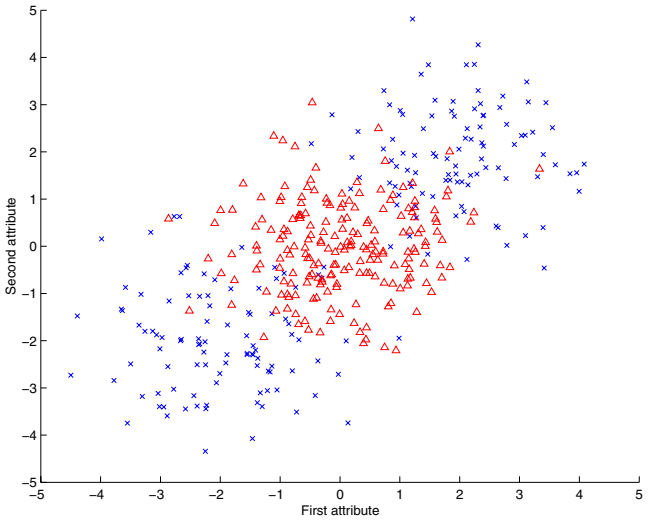

Fig. 3. Artificial two dimensions database used in experience 1. 200 examples in two classes. The blue crosses are separated in two by the red triangles.

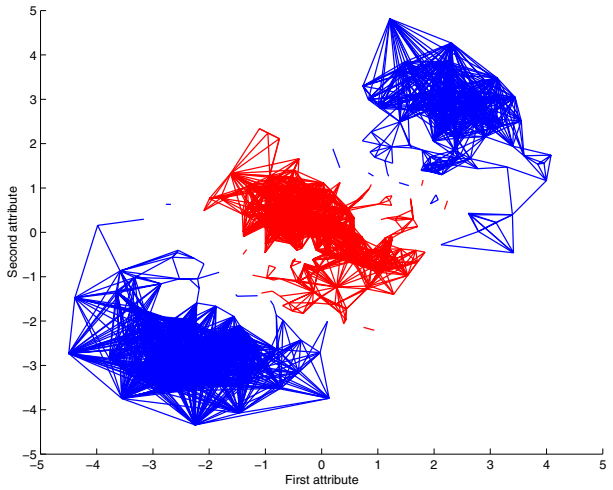

Fig. 4. Graphs obtained by the segmentation process. The blue class is divided in 19 subclasses. Two of them have more than 3 examples. The red class is divided in 21 subclasses with one which has more than 3 examples. Graphs with one node are not represented.

Another important point is that during the subclass construction, each example search in every directions how it can expand the graph. So, the shape of the subclasses has no importance.

The second stage of the experiment is the construction of prototypes. The figure 5 shows the degrees of typicality for the first attribute without segmentation step. For the crosses class, these degrees are quite low. This prototype has no typical value. So, for this class and this attribute, no information could be infered. As the crosses class is splitted in two, its examples do not resemble each other. Moreover, each group of the class of crosses is less dissimilar to the

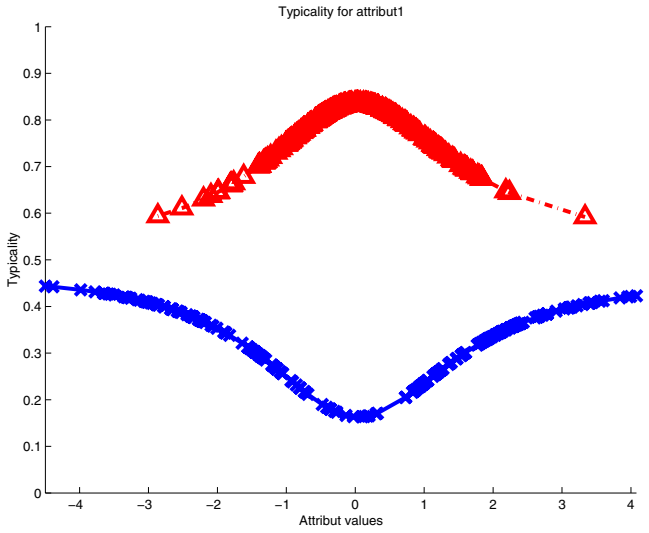

Fig. 5. Typicality degrees for the first attribute without the class segmentation. There is no typical values for the class of crosses.

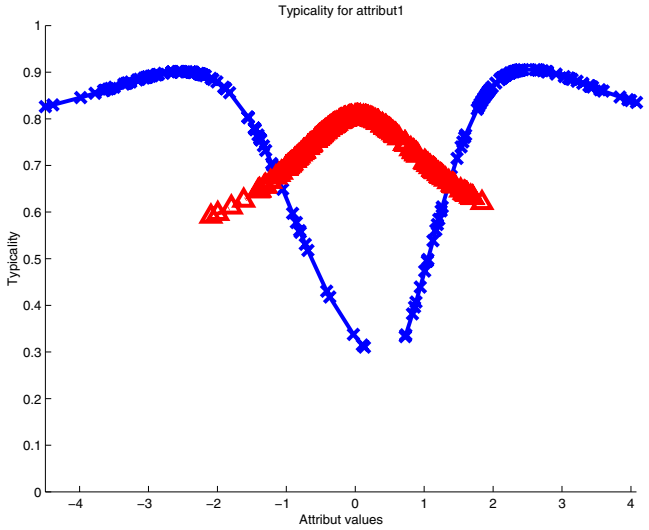

Fig. 6. Typicality degrees for the first attribute with a segmentation. Two typical values ranges appear for the class of crosses.

triangle class than the other group.

The figure 6 represents the typicality degrees for the first attribute of the data with a segmentation step. The class of crosses has two ranges of typical values. This is a useful information for a classification task or for a characterization work. Indeed, each subclass, very dissimilar, has been identified and described with a meaningful fuzzy prototype.

On this database our method has discovered two major behaviours that the Lesot's approach has not.

B. Application: characterization of real data

As we have seen in the previous part, our method isolates several behaviours for a same class. In this part we show the contributions of our approach for characterization and visualisation tasks by constructing the fuzzy prototypes for a real database. Our purpose is to propose an intuitive summarization of the data. For this application on experimental data we have chosen to work on the glasses database downloaded from the UCI data repository [9]. 214 examples of six types of glasses (the classes) are described by 9 attributes. We focus our study on the first attribute (the refractive index) because it is one of the descriptors where the prototype construction without class segmentation get the best results.

The figure 7 shows the fuzzy prototypes constructed without a class segmentation.

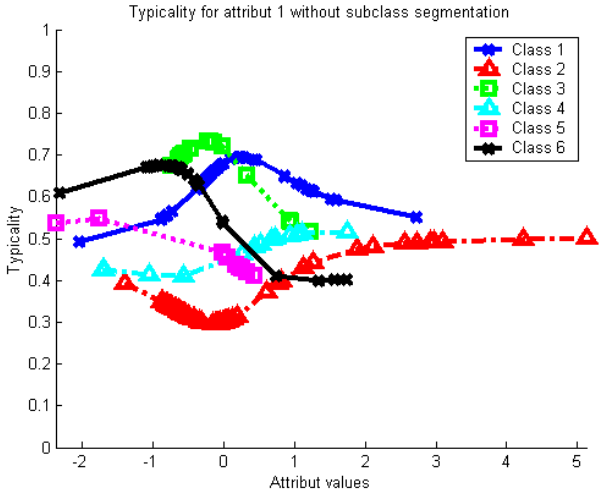

Fig. 7. Typicality degrees obtained for the first attribute of the glasses database without class segmentation.

As it was suggested in Lesot [1], we can consider that a value is significative if its typicality degree is beyond a threshold of 0.7 for example.

Without a class segmentation, we can observe two types of classes: the typicality degrees of the prototypes of the classes 1, 3 and 6 are significative whereas those of the three others (2, 4 and 5) stay below the threshold. This last results may seem unsatisfactory. However, it is important to notice that Lesot's method is appropriate to a precise context: classes coming from clustering or well defined classes (i.e. homogeneous classes). The database of figure 7 shows the limit of Lesot's approach and the need to propose a pretreatment in the case of non homogeneous classes.

Figure 8 illustrates the benefits of a class segmentation process. In this visualization all the subclasses are not represented: We have chosen to filter groups containing less than 3% of the total size of the database. We insist on the fact that the filtered subclasses are not visually represented, but they are taken into account in the process.

The blue (1) and the red (2) classes are here better characterized: both have been segmented in two major subclasses and the typicality degrees of their prototypes are high (around 0.9). Now, these classes have two ranges of typical values. It is important to notice, that the subclasses have not the same weight regarding the number of examples they have: indeed, the blue (1) class has been segmented into balanced

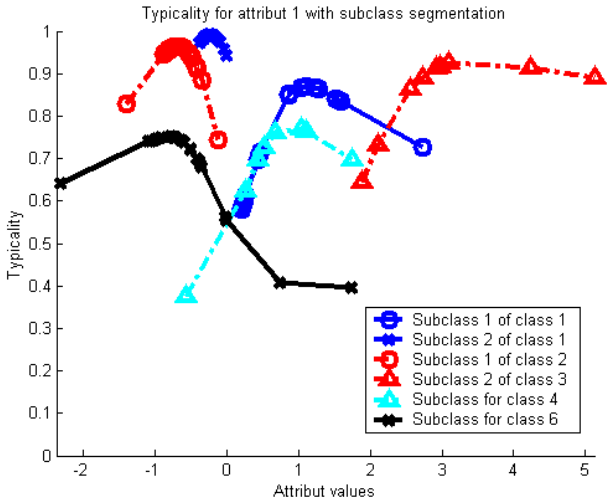

Fig. 8. Typicality degrees obtained for the first attribute of the glasses database with class segmentation and a subclass filtering (Threshold of 3%).

groups with many examples (71% of the class members in total) whereas the red (2) one is represented by a prototype with 40% of the red (2) class examples and another one with 8.5%.

The cyan (4) and the black (6) classes gives another remarkable situation. These two classes has been segmented in several subclasses with a unique subclass besides the threshold of 3%. It means that the well defined prototypes obtained without the segmentation process are still present after the segmentation process.

One can remark that the pink (5) and the green (3) classes have disapeared in figure 8. These classes has been so segmented that no subclass can exceed the threshold. We can of course lowered the threshold.

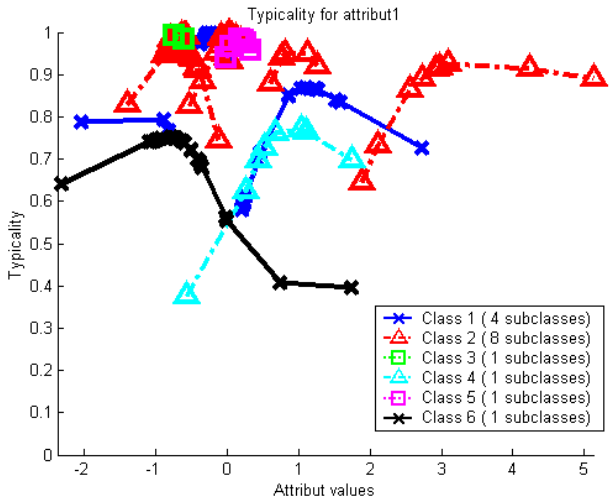

Fig. 9. Typicality degrees obtained for the first attribute of the glasses database with class segmentation and a subclass filtering (Threshold of 1%).

Figure 9 shows that with a threshold of 1% the pink (5) and the green (3) classes reappear. But, six new fuzzy prototypes for the red (2) class appear. and then the characterization is more difficult.

V. Conclusions and Further Works

In this paper, we have presented an original method based on a class segmentation process which allows the characterization, the visualization and the summarization of several behaviours of a complex class, thanks to a representation by fuzzy prototypes. Our approach enrichies the Lesot's one by better defining a class.

In a future work we will study our strategy of subclasses construction. The growth of the graph is currently stopped when an enemy is encoutered. It could be interesting to be more tolerant and to study the impact of the number of ennemies allowed in the growth step.

References

[1] M.-J. Lesot, L. Mouillet, and B. Bouchon-Meunier, "Fuzzy prototypes based on typicality degrees," *Fuzzy Days04*, 2004.

[2] M. Rifqi, "Constructing prototypes from large databases," *Information Processing and Management of Uncertainty (IPMU'96)*, 1996.

[3] E. Rosch, "Principles of categorization," *E. Rosch and B. Lloyd, editors, Cognition and categorization*, pp. 27–48, 1978.

[4] N. Japkowicz, "Supervised learning with unsupervised output separation," in *Proceedings of the IASTED International Conference on Artificial Intelligence and Soft Computing*, 2002, pp. 321–325.

[5] R. Vilalta and I. Rish, "A decomposition of classes via clustering to explain and improve naive bayes," in *In Proceedings of European Conference on Machine Learning*, 2003, pp. 444–455.

[6] M.-J. Lesot, "Similarity, typicality and fuzzy prototypes for numerical data," *6th European Congress on Systems Science, Workshop Similarity and resemblance*, 2005.

[7] M. Detyniecki, "Mathematical aggregation operators and their application to video querying," Ph.D. dissertation, Université de Paris VI, 2000.

[8] W. Silvert, "Symmetric summation: a class of operations on fuzzy sets," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, pp. 659–667, 1979.

[9] "Uci machine learning repository." [Online]. Available: http://www.ics.uci.edu/ mlearn/MLRepository.html