Fuzzy Partition-based Distance Practical Use and Implementation

Serge Guillaume
Irstea, UMR ITAP
BP 5095
34196 Montpellier, France
Email: serge.guillaume@irstea.fr

Brigitte Charnomordic
INRA/SupAgro, UMR MISTEA
2 Place Viala
34060 Montpellier, France
Email: bch@supagro.inra.fr

***Abstract*—This work discusses the implementation of a semi-distance based on fuzzy partitions, that allows to introduce expert knowledge into distance computations done on numerical data. It can be used in various kinds of statistical clustering or other applications. The semi-distance univariate behaviour is first studied, then a multivariate clustering case study is presented, that includes an optimization procedure. All calculations are done using open source softwares: *FisPro* for fuzzy modelling and *R* for clustering. Emphasis is given to a number of software functionalities, which are targeted at the practical use of the semi-distance.**

Keywords*—*fuzzy set, semi supervised, expert knowledge, metric, clustering

I. Introduction

Many algorithms, for instance clustering or classification techniques, are based on a dissimilarity function d, which will determine how the dissimilarity of two elements is calculated: dissimilarity between individuals, between individual and group or between groups. Usually these algorithms are very sensitive to the choice of the dissimilarity function [1], [16]. The dissimilarity function must have the good properties of non-negativity and symmetry and satisfy the triangle inequality. It can be proper (identity of indiscernibles), or semi-proper. In the first case, it is a metric or distance, in the second case it is a pseudo metric or semi-distance.

The most common metrics for numerical values are defined by the L^p norms:

$$\|x\|_p = (|x_1|^p + \ldots + |x_M|^p)^{\frac{1}{p}}$$

where x is a multidimensional vector $(x_1, x_2 \ldots x_M)$. The classical case of the Euclidean norm is obtained for $p = 2$.

Other metrics can be used in multivariate problems when variables are not independent, for instance the Mahalanobis distance or the Choquet-Mahalanobis operator [23].

Dissimilarity functions have also been defined for non ordered categorical data, e.g. material={plastic, metal, wood}. Their definition is mainly based on the presence or absence of a given attribute. In [3], several similarity measures are compared. Some of them take into account the sample size, the number of values taken by the attribute and the frequencies of these values in the data set. The Lorenz curve allows the ranking of multi-attribute categorical data [8].

To our knowledge, not much attention has been paid to similarity measures for ordered categorical data, such as: price={cheap, average, expensive}. In this case the distance between *cheap* and *expensive* has to be higher than the one between *cheap* and *average* or the one between *average* and *expensive*.

Like many other concepts, the concept of dissimilarity or distance has been generalized to fuzzy sets. In her survey [2], Bloch recalls the three types of fuzzy distances already defined: between two points in a fuzzy set, from a point to a fuzzy set and between two fuzzy sets. Many works tackle this problem, among them [7], [5], [19], [9], [4]. Bloch [2] does not mention the distance between individuals within a fuzzy partition (FP). Such a distance was little studied.

In [**?**], the fuzzy set formalism is used to define a dissimilarity function between individuals within a fuzzy partition, based on the combination of numerical and symbolic information. The proposed function is a semi-distance, denoted by FP-based, that introduces a semi supervised aspect into dissimilarity-based algorithms. The semi supervision is done by using available expert knowledge to superimpose linguistic concepts onto numerical data.

The present work aims at studying some practical uses of FP-based semi-distances, and their software implementation in the open source software *FisPro*[1], that corresponds to ten years of research and software development in the field of learning interpretable fuzzy inference systems from data. In this paper, the focus is on some new functionalities, including a specific graphical representation to show the Euclidean space distortion due to the semi-distance, some partition quality indices and the distance type selection in fuzzy partitioning or fuzzy inference system optimization.

The paper is organised as follows. Section II recalls the principles of a FP-based semi-distance, used to calculate the dissimilarities between data points. The definitions are given in the particular case of Strong Fuzzy Partitions, for the univariate and multivariate cases. Section III gives some elements about the *FisPro* implementation. Section IV first illustrates the univariate behavior, and then gives an example of use to design fuzzy partitions. A multivariate case study is addressed in Section V, where FP-based semi-distances are used to optimize a clustering quality index. Section VI gives

[1] http://www7.inra.fr/mia/M/fispro/

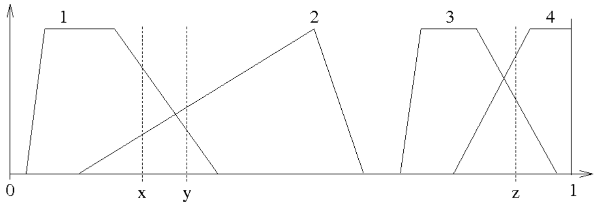


Figure 1. Example of FP used to define d_P in the general case.

some conclusions.

II. Principles of a fuzzy-partition based distance

This section recalls the principles of the FP-based semi distance, that allows the introduction of expert knowledge in any algorithm when distance calculations are involved. The FP-based semi-distance combines numerical and symbolic elements. Its numerical part allows it to handle multiple membership in overlapping areas between two fuzzy sets, while the symbolic one takes into account the granularity of the concepts associated with the fuzzy sets.

The proposal applies to data in the unit interval $U = [0, 1]$ and relies on Fuzzy Partitions (FPs). The function is called d_P.

A fuzzy partition carries semantics and knowledge about the variable behavior. To respect the semantics attached to the partition-related concepts, the distance between two non distinguishable elements must be equal to zero. This is the case for all elements within a given fuzzy set kernel. Furthermore, elements belonging to different concepts must always have a distance higher than elements belonging to the same concept. Overall the progressivity of the distance between any two elements must be respected: the farther apart the data points, the greater their distance should be.

The function d_P has been defined in [**?**] for general FPs, as the one shown in Figure 1, and has been proven to fulfill the properties of a semi-distance. In this paper, we will use for illustration the case of Strong Fuzzy Partitions (SFPs). Indeed, in *FisPro*, all of the partitions automatically generated from data are of this kind, that has the advantage of a clear semantics. Let us give the necessary definitions.

A. Strong Fuzzy Partition

A Strong Fuzzy Partition (SFP) described by f membership functions (MFs) on the universe U fulfills the following condition:

$$\forall x \in U, \qquad \sum_{i=1}^{f} \mu_i(x) = 1 \tag{1}$$

$\mu_i(x)$ is the x membership degree to the ith fuzzy set, labelled i to simplify further notations. An example of SFP is shown in Figure 2. Let us note that the formal definition of SFPs is independent of the MF shape.

B. Monodimensional FP-based semi-distance

In the case of SFPs, the expression of d_P becomes quite simple.

Let $X_i = [\underline{K_i}, \underline{K_{i+1}}[$, for $0 \leq i \leq f$, where $\underline{K_i}$ is the lower bound of the ith fuzzy set in the partition.

We denote by $I(x)$ the function such that:

$$\forall i \in [1, f], x \in X_i \Leftrightarrow I(x) = i$$

Let us introduce the function P:

$$P(x) = I(x) - \mu_{I(x)}(x) \tag{2}$$

P is a positive non-decreasing function of x and is increasing in overlapping zones.

$d_P(x, y)$ is defined as:

$$d_P(x, y) = \frac{|P(x) - P(y)|}{f - 1} \tag{3}$$

1) Comparison with the Euclidean distance: The FP used to define the semi-distance distorts the numerical space, so that the FP-based semi-distance does behave differently than the Euclidean distance.

Its behavior essentially depends on two main characteristics of the FP: the number of linguistic concepts, which affects the symbolic component, and the size of the different kernels. The wider the kernels, the more there are indistinguishable values. Then the slope of the fuzzy sets also has an impact on d_P.

Figure 2 shows an example of rank inversion of the FP-based distance results compared with the Euclidean distance ones. With d_P, x and y are further apart than y and z, while they would be closer than y and z, were the Euclidean distance used. This rank inversion is due to the fact that all elements within a given fuzzy set kernel have a null distance.

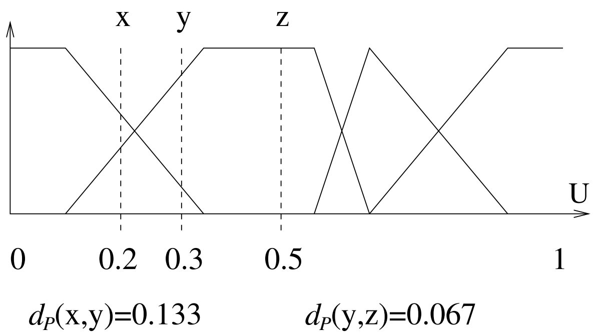


Figure 2. Example of fuzzy partition based distance (d_P) behavior.

2) Particular case of regular SFPs: A regular SFP is composed of triangular membership functions, with equidistributed kernel centers $K_1 \dots K_f$, with $K_1 = 0$ and $K_f = 1$.

For this particular case, it was shown in [?] that the proposed function d_P is a distance and that it is equivalent to the Euclidean distance, regardless of the number of terms in the partition. The reason is that d_P distorts the Euclidean space according to the symbolic distance between concepts and the indistinguishability of the kernel elements. In the case of a regular SFP, all kernels are reduced to single points and are equidistant, so there is no distortion.

C. Multidimensional FP-based distance

A simple and efficient way to obtain a multidimensional pseudo-metric is to perform a Minkowski-like combination of the univariate pseudo-metrics. Let two multidimensional points $x = (x_1, \dots, x_M)$ and $y = (y_1, \dots, y_M)$ with $x_i, y_i \in [0, 1],\ \forall i \in 1, \dots, M$.

We have the following definition for the multi-dimensional distance, which is also a pseudo-metric:

$$\forall x, y \quad d(x, y) = \left[\sum_{j=1}^{M} (d_j(x_j, y_j))^k \right]^{\frac{1}{k}} \tag{4}$$

where k is a scalar positive value, corresponding to the Minkowski exponent. The advantage of this definition is that one can use different sub-distances in the various dimensions, for instance a FP-based semi-distance in dimension a if expert knowledge is available for the corresponding feature, and on the contrary, the Euclidean one in dimension b.

III. FisPro implementation

FisPro [12] is an open source toolbox to design and optimize fuzzy inference systems (FIS). Among fuzzy software products, *FisPro* stands out because of the interpretability of fuzzy systems automatically learnt from data. Interpretability is guaranteed in each step of the FIS design with *FisPro*: variable partitioning, rule induction, optimization. *FisPro* includes several modules: fuzzy partitioning, rule and partition learning, inference and FIS optimization.

In this paper, we will not use FIS, but many functionalities implemented in *FisPro* are very useful to design fuzzy partitions, to view them and to define FP-based semi-distances.

Fuzzy partitioning in *FisPro* can be done manually, or automatically from data. Plots are available to view the data distribution jointly with the fuzzy partition. New functionalities have been added to *FisPro*, in order to implement the FP-based semi-distance presented in Section II and are now available in *FisPro* version 3.5. The *data* menu includes a *distance* option, that allows to compute a distance matrix according to Definition 4. The combination of Euclidean and FP-based for active variables is user-customized. A univariate plot is available to show the difference between Euclidean and FP-based distances.

IV. Univariate illustration

In this section, we present some illustrations with *FisPro* of the FP-based semi-distance in the univariate case, first the graphical illustration of the distortion vs the Euclidean distance, then examples of use to design fuzzy partitions from data.

A. Behavior

The effect of the fuzzy partition distortion mentioned in Section II-B1 can be viewed in *FisPro*. Let us design a fuzzy partition to describe mammals' milk fat content, such as the one shown in Figure 3. That partition will be used later for a case study in Section V. Figure 4, a screen shot of *FisPro*, illustrates the distortion of the numerical space due to the use of the FP-based partition.

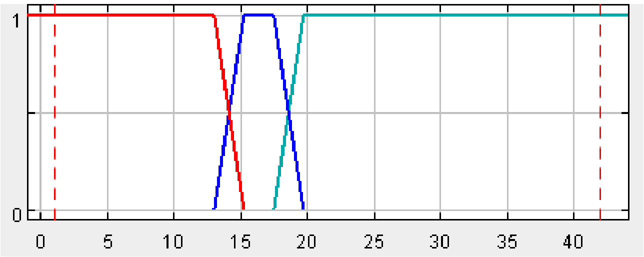


Figure 3. A fuzzy partition for the Fat variable (mammals' milk data set).

Distances are calculated on a data grid corresponding to the range of the variable. The abscissa and ordinate axes are plotted accordingly. For each pair (x, y), the calculation is done separately for the Euclidean distance and for the FP-based semi-distance. The difference $d_P(x, y) - d_E(x, y)$ is translated into a given color, according to the color scale represented on the left. The color scale is chosen so that the white color means a null difference, the red color a negative one and the blue color a positive one. The more intense the color, either blue or red, the greater the distorsion. Naturally, the figure is symmetrical with respect to the diagonal, which corresponds to $d(x, x)$.

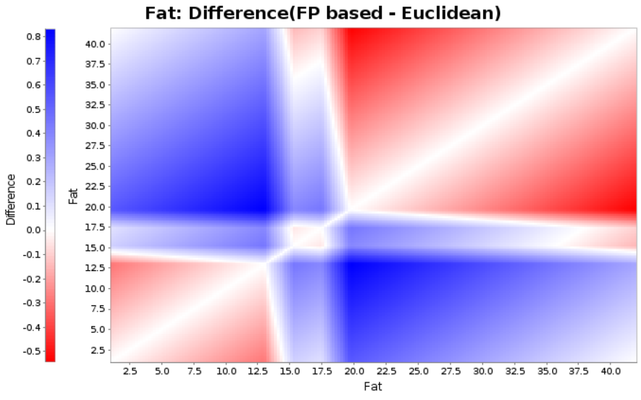


Figure 4. The distortion of the Euclidean distance, according to the fuzzy partition plotted in Figure 3.

B. Use to design fuzzy partitions

FPs can be designed from expertise or/and data [15]. In this section, we introduce a FP-based semi-distance in a method to design FPs from data, called HFP, standing for *Hierachical Fuzzy Partitioning*, that was proposed in 2004 [11].

The procedure generates a hierarchy including best partitions of all sizes from M to 2 fuzzy sets. The maximum size is determined according to the data distribution and corresponds to the finest resolution level. An ascending method is used for which a merging criterion is needed. This criterion minimizes the sum of pairwise distances over all the data points. The underlying idea is to maintain as far as possible the homogeneity of the structure built at the previous stage. To build the initial partition, each distinct value of the variable distribution is represented as a triangular MF kernel.

Similar collections of fuzzy partitions can also be generated using the *k-means* algorithm. In this case, the partitions are independent one from each other.

The generated partitions can be compared to regular partitions of same size, which do not take into account the data distribution but only the data range.

Several indices have been defined to characterize fuzzy partitions. The partition coefficient (PC) and the partition entropy (PE), both proposed by Bezdek [1], are implemented in *FisPro*. Let N be the data set size, c the number of fuzzy sets and $\mu_i(k)$ the membership degree of the ith item in the kth group, the indices are:

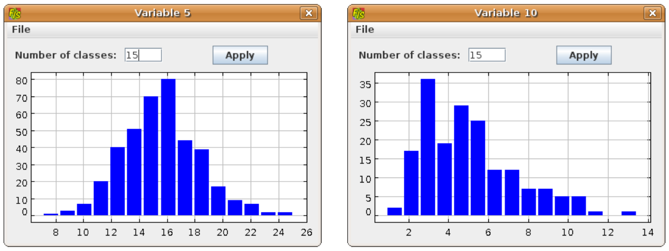
$$PC = \frac{1}{N} \sum_{k=1}^{N} \sum_{i=1}^{c} \mu_i^2(k)$$

$$PE = -\frac{1}{N} \left\{ \sum_{k=1}^{N} \sum_{i=1}^{c} [\mu_i(k) \log_a(\mu_i(k))] \right\}$$

According to these criteria a good partition should minimize entropy and maximize the coefficient partition.

As an illustration, data distributions are considered for two variables of the *auto-mpg* and *wine* well known data sets from the UCI repository [10]. The indices are computed for partitions from two to seven terms generated from these data by the available methods. The *HFP* algorithm can be run with several types of distance. The one proposed in 2004 is called *numerical* in *FisPro* while the new one, based on d_P, is called *symbnum*. In the following, *hfp1* stands for the *HFP numerical* and *hfp2* for the *HFP symbnum*.

Two fifteen class histograms are plotted in Figure 5, with the x-axis representing the input values and the y-axis the number of data items. The histogram on the left part corresponds to the fifth variable of the *auto-mpg* data set, and the one on the right to the tenth variable of the *wine* data set. These two distributions are dissimilar in shape: a quasi Gaussian distribution for the fifth variable, and a multimodal skewed one for the tenth variable.



Number of classes: 15

Number of classes: 15

Figure 5. Data distribution for v5 from auto-mpg (left) and v10 from wine (right) data sets - v5 is the car weight, v10 is the wine color intensity.

Tables I and II show the best partition index for the corresponding variable according to the two indices. The partition size is given in parentheses following the index value.

Table I. OPTIMAL PARTITION INDEX VALUE AND SIZE FOR AUTO-V5

auto-v5	reg	km	hfp1	hfp2
PC	0.69 (3)	0.80 (2)	0.76 (2)	0.92 (2)
PE	0.47 (3)	0.30 (2)	0.36 (2)	0.13 (2)

Table II. OPTIMAL PARTITION INDEX VALUE AND SIZE FOR WINE-V10

wine-v10	reg	km	hfp1	hfp2
PC	0.67 (7)	0.82 (2)	0.75 (2)	0.93 (2)
PE	0.49 (7)	0.26 (2)	0.37 (2)	0.11 (2)

Whatever the data and the method, both indices PC and PE agree to select the same size for each partition.

For both variables, the indices are ranked in the same order. As expected, regular fuzzy partitions give the poorest results. For *wine-v10*, this method fails to identify a compact structure. This is obviously due to the histogram irregular shape. Then the three remaining methods select the same partition size for the two variables: 2 fuzzy sets. The index values are also ranked in the same order: *k-means* yields better results than *hfp1* but the proposed distance significantly improves the index value in both cases.

Let us mention that both PC and PE indices are available in *FisPro*. They are displayed in the HFP result window and in the joint plots of data and fuzzy partitions.

V. MULTIVARIATE CLUSTERING CASE STUDY

This section presents an illustration of the multidimensional FP-based d_P on a clustering case. In particular, an optimization procedure is carried out to improve a clustering quality index.

A. Data and clustering details

Data chosen to demonstrate the interest of the FP-based semi-distance are taken from [17] and have been used previously to illustrate clustering procedures [6]. The data set describes the percentages of *water*, *protein*, *fat*, *lactose* and *ash* in the milk of 25 mammals. Data are given in Table III.

Table III. INGREDIENTS OF MAMMALS MILK

	#	Water	Protein	Fat	Lactose	Ash
Bison	1	86.90	4.80	1.70	5.70	0.90
Buffalo	2	82.10	5.90	7.90	4.70	0.78
Camel	3	87.70	3.50	3.40	4.80	0.71
Cat	4	81.60	10.10	6.30	4.40	0.75
Deer	5	65.90	10.40	19.70	2.60	1.40
Dog	6	76.30	9.30	9.50	3.00	1.20
Dolphin	7	44.90	10.60	34.90	0.90	0.53
Donkey	8	90.30	1.70	1.40	6.20	0.40
Elephant	9	70.70	3.60	17.60	5.60	0.63
Fox	10	81.60	6.60	5.90	4.90	0.93
Guinea Pig	11	81.90	7.40	7.20	2.70	0.85
Hippo	12	90.40	0.60	4.50	4.40	0.10
Horse	13	90.10	2.60	1.00	6.90	0.35
Llama	14	86.50	3.90	3.20	5.60	0.80
Monkey	15	88.40	2.20	2.70	6.40	0.18
Mule	16	90.00	2.00	1.80	5.50	0.47
Orangutan	17	88.50	1.40	3.50	6.00	0.24
Pig	18	82.80	7.10	5.10	3.70	1.10
Rabbit	19	71.30	12.30	13.10	1.90	2.30
Rat	20	72.50	9.20	12.60	3.30	1.40
Reindeer	21	64.80	10.70	20.30	2.50	1.40
Seal	22	46.40	9.70	42.00	0.00	0.85
Sheep	23	82.00	5.60	6.40	4.70	0.91
Whale	24	64.80	11.10	21.20	1.60	0.85
Zebra	25	86.20	3.00	4.80	5.30	0.70

The aim of our case study is to cluster the mammals, according to these five milk components, by introducing FP-based semi-distances into a classical clustering procedure. First, a dissimilarity matrix between all pairs of items is computed. Then, the dissimilarity matrix is used as an input to the clustering algorithm. Because the standard *k-means* does not give stable results with the Euclidean distance, a robust version, called *pam* (partitioning around medoids) [18], is used. The main difference between *pam* and *k-means* is the definition of the cluster centers: in the robust version, the cluster centers are not computed as the mean but are necessarily data items, in a formation called a medoid. The R implementation [21] of *pam* is used in the experiments.

We first present the clustering results using the Euclidean distance in all dimensions. Then we introduce FP-based semi-distances based on the data distribution, as detailed in Section IV-B, and we show that the clustering quality, measured according to a reference silhouette index [22], can be improved by optimizing the fuzzy partitions parameters. An analysis of the results with respect to the semantics is included. Experiments have been done with various number of clusters, and the results presented here correspond to the three cluster case.

B. Clustering using the Euclidean distance

The results of the *pam* partitioning run on the multi-dimensional data are shown in Figure 6, which is a two-dimensional plot. The three clusters are labeled E_1, E_2, E_3.

All observations are represented by points in the plot, and principal components analysis (PCA) is used to reduce the dimensions to the two first axes. An ellipse is drawn around each cluster. The first two components of the PCA explain 94.91% of the variability, and we will study the cluster composition on the first plane, also called the principal plane. Of course, some individuals may be closer or farther on the other factorial planes.

The cluster composition is somewhat unexpected. *Dog* is included in the cluster of sea mammals, whereas *cat* and *pig* are assigned to a different cluster.

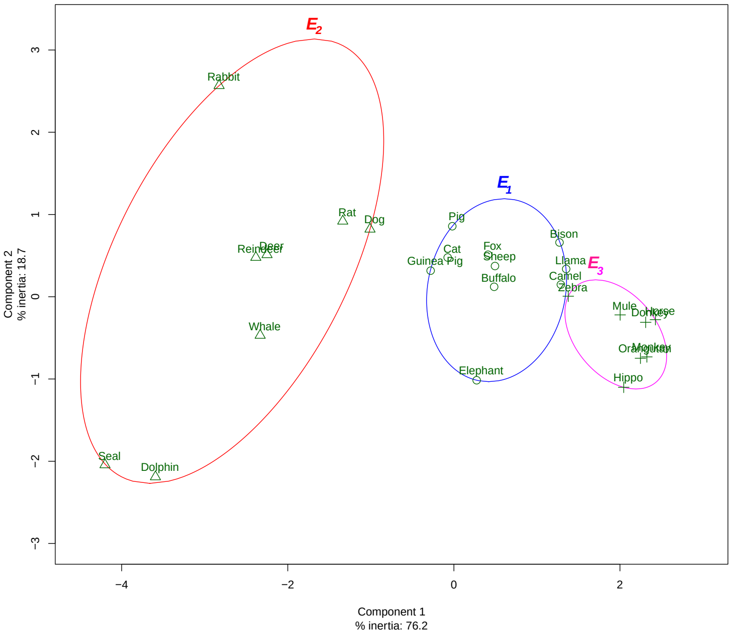


Figure 6. Clusters obtained with the Euclidean distance.

A powerful indicator can be computed using the *silhouette* index. To construct the silhouettes $S(i)$ for each item i, the following formula is used:

$$S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

where $a(i)$ is the average dissimilarity of item i to all other items in the same cluster and $b(i)$ is the minimum of average dissimilarity of item i to all items in other clusters. $b(i)$ can be seen as the dissimilarity between item i and its neighbor cluster, i.e., the nearest cluster to which it does not belong. The average silhouette S_c for each cluster is simply the average of the $S(i)$ for all items in the *cth* cluster. Similarly, the overall average silhouette $\overline{S}$ is the average of the $S(i)$ for all items in the whole data set.

Figure 7 shows the silhouette values corresponding to the clustering using the Euclidean distance. Three of the silhouette values are negative: *llama, camel* and *dog*. Others, like the *rat*-related silhouette value, are close to zero.

The silhouette index is based on cluster tightness and separation and is derived from the formula $-1 \leq S(i) \leq 1$. A value close to one indicates that the observation is correctly assigned to a group; a small value, even more so a negative one, results in a wrong assignment. The largest overall average silhouette indicates the best clustering.

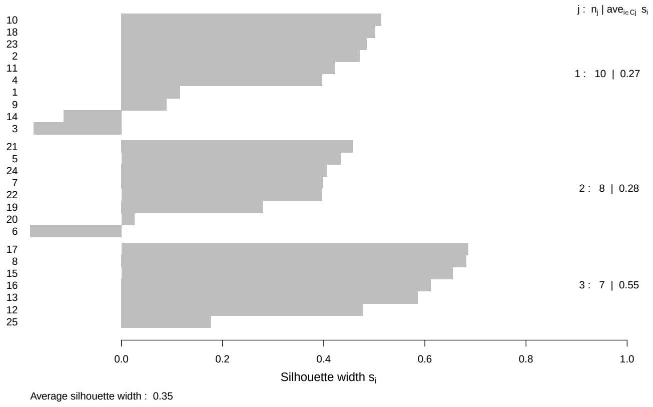


Figure 7. Silhouettes with Euclidean distance.

C. Optimizing fuzzy partitions to maximize silhouette index

The clustering can also be done using a FP-based distance. Even if it is possible to design the FP according to expert knowledge, it is also possible to build the fuzzy sets according to the data distribution. Then the fuzzy set parameters can be optimized according to a clustering quality index, chosen here as the silhouette index.

Table IV is a summary of the silhouette indices, resulting from several clustering procedures, either using the Euclidean distance or a FP-based distance.

Table IV. SILHOUETTES FOR A 3 CLUSTER CONFIGURATION FOR VARIOUS DISTANCES.

Fuzzy Partition Generation	Silhouette	Configuration
None (Euclidean distance)	0.35	
k-means	0.46	2 2 2 2 2
HFP	0.47	3 3 3 4 2

The configuration column gives the optimal number of fuzzy sets, according to the partition indices, for k-means and HFP partitions.

Figures 8 and 9 show the clusters and silhouettes corresponding to the HFP partition-based d_P.

The cluster composition is slightly different from the one obtained with the Euclidean distance. According to the Euclidean distance, the Camel, the Llama and the Elephant belonged to the same cluster than the Fox, the Sheep and the Buffalo while using the HFP FP-based distance they are now together with the Zebra, the Mule, the Horse and the Donkey. The Guinea Pig also moved from the Fox-group to the marine mammals group. The resulting overall silhouette index is improved.

d_P can be considered as parameterized by the fuzzy partition. As SFPs are used, there are as many parameters as fuzzy set kernel boundaries for each variable. Using the methods described in Section IV-B, these parameters have been set according to the data distribution, without any other consideration.

If the objective is to get a high overall silhouette index, this index can be used as a cost function for parameter optimization. In *FisPro*, the optimization module implementation is based upon the Solis and Wets algorithm [13]. The optimization process only modify the MF parameters, without adding or removing any of them. In order to allow the algorithm for more degrees of freedom the initial triangular MF are modeled as trapezoids with identical kernel bounds.

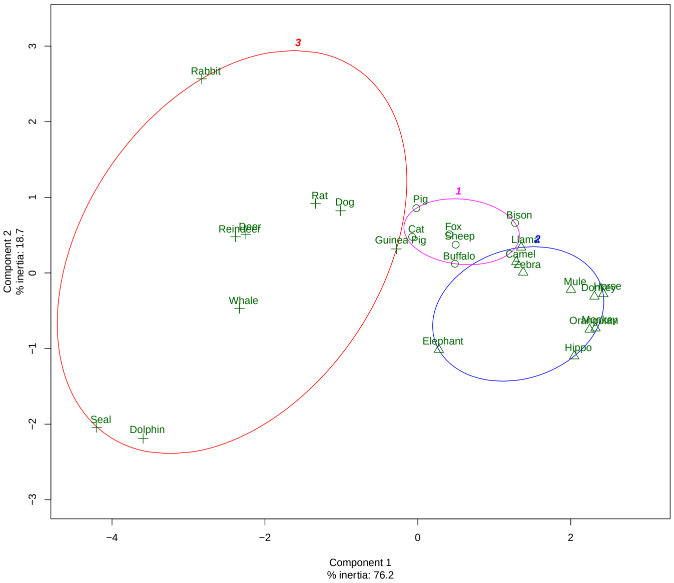


Figure 8. Initial HFP FP-based cluster representation.

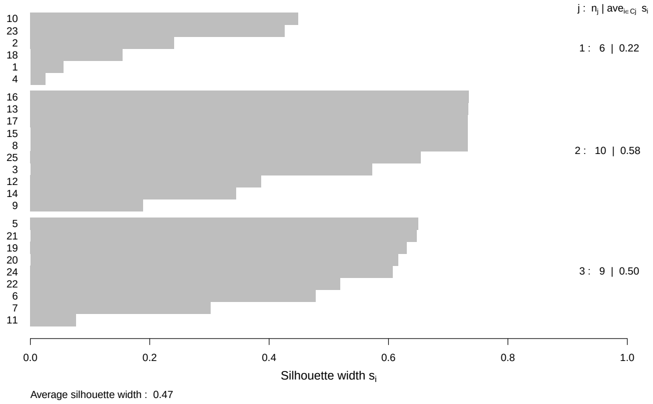
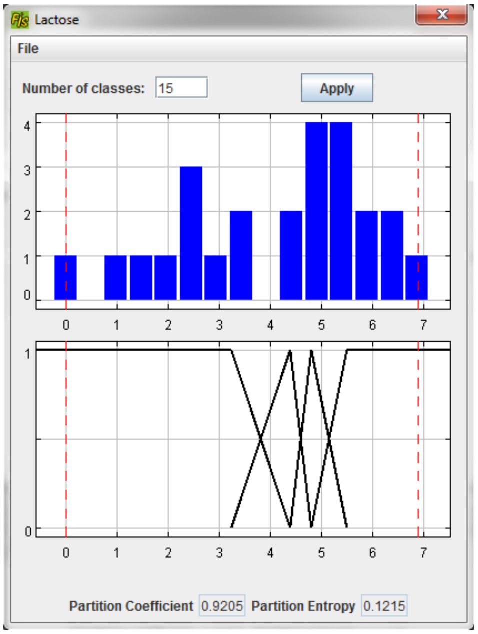


Figure 9. Initial silhouettes with HFP FP-based semi-distance.

Tests have been carried out to assess the algorithm sensitivity to its own parameters: noise level and input order optimization. When the noise level is high enough the results do not depend on its value. In the following, 0.01 is used. The order can be set according to the partition index values, but the partitions can also be optimized in the order they are stored in the file.

All of the fuzzy partitions are modified. The *Lactose* variable, the only one with four MFs is chosen as an illustration. Figure 10 shows the initial partition jointly with the data distribution. The PC and PE indices introduced in Section IV-B are displayed at the bottom.

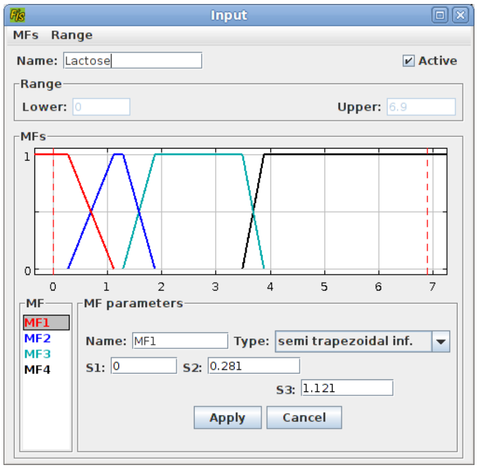


Number of classes: 15

Partition Coefficient 0.9205 Partition Entropy 0.1215

Figure 10. HFP Lactose initial partition.

The optimized partition is plotted in Figure 11.



Name: Lactose ☑ Active

Lower: 0 Upper: 6.9

Name: MF1 Type: semi trapezoidal inf.

S1: 0 S2: 0.281

S3: 1.121

Figure 11. Optimized Lactose partition.

The overall silhouette index is really improved, up to 0.66, as shown in Figure 12. This is due to some changes in cluster composition, plotted in Figure 13: the Elephant and the Guinea pig are now back with the Fox group, the Bison moved from the Fox group to the Horse group. This one keeps including the Camel and the Llama.

D. Result analysis

As the distance function is based upon multidimensional computation, the analysis of these changes is not easy. Nevertheless, a careful examination of some univariate fuzzy partitions, jointly with the data, is likely to bring useful information.

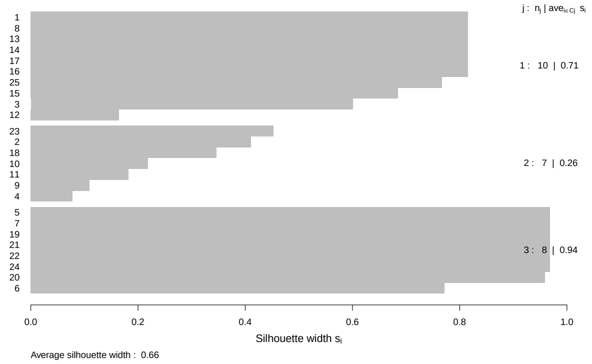


Figure 12. Silhouettes with HFP FP-based semi-distance after optimization.

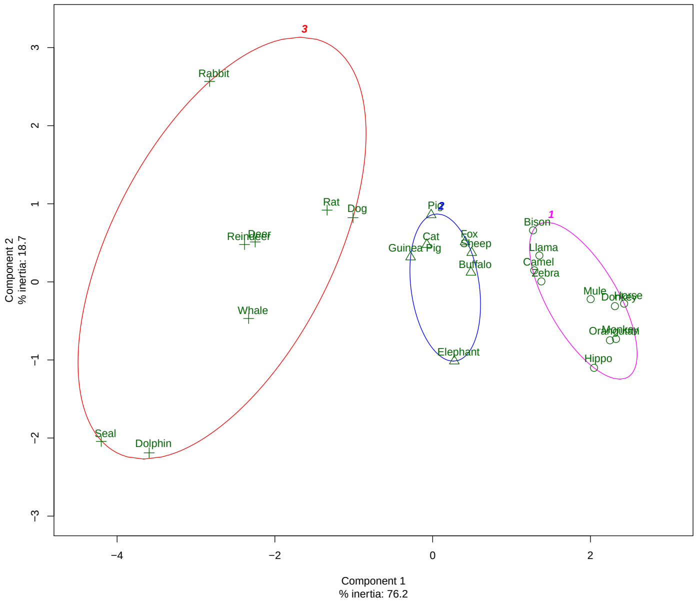


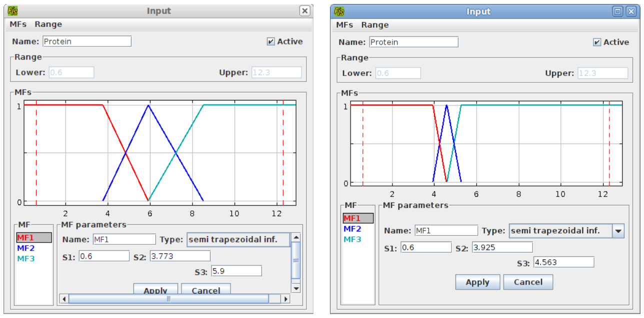
Figure 13. HFP FP-based cluster representation after optimization.

Let the Buffalo and the Mule be the representatives of the two clusters labelled 1 and 2 in Figure 13. The last cluster which includes the marine mammals is untouched. The Bison, the Llama and the Camel moved from the Buffalo to the Mule, the Elephant went back to the Buffalo after optimization.

The Protein fuzzy partition is plotted before and after optimization (Figure 14, left and right respectively).

Considering that variable, according to the partition displayed in Figure 14 (right-hand side), the Camel, Llama, but also the Elephant, only belong to the first MF, as does the Mule; the Buffalo fully belongs to MF3 and the Bison lays in the overlapping area between MF2 and MF3.

According to the Fat variable, the Bison, Camel, Llama only belong to the first MF, like the Mule; the Elephant fully belongs to the second MF, like the marine mammals, while the Buffalo partly belong to the two first ones.



Name: Protein ☑ Active
Range
Lower: 0.6 Upper: 12.3
Name: MF1 Type: semi trapezoidal inf.
S1: 0.6 S2: 3.773
S3: 5.9

Name: Protein ☑ Active
Range
Lower: 0.6 Upper: 12.3
Name: MF1 Type: semi trapezoidal inf.
S1: 0.6 S2: 3.925
S3: 4.563

Figure 14. Initial (left) and optimized (right) Protein fuzzy partitions.

The Elephant is close to the Mule group according to Protein and Lactose, but close to the marine mammals' group according to the Fat, thus it is assigned to the intermediate group, the Buffalo's.

It is worth mentioning that the optimization procedure keeps the fuzzy partitions interpretable. Even if they are automatically built, the partitions carry a high level of semantics. As the MFs are clearly divided, they can be labelled using a linguistic term, e.g. Low, Average, High for a 3-term partition, and used for linguistic rule-based reasoning. These properties are guaranteed by the strong fuzzy partition paradigm. As only the MF shapes and locations are modified, the semantics attached to the initial fuzzy partition remains valid for the optimized partition.

VI. Conclusion

In this paper, we discussed the practical use of a semi-distance based on fuzzy partitions, and its implementation in the *FisPro* open source software. FP-based semi-distances and Euclidean distances can be combined in the multi-variate case by using a Minkowski combination. A dissimilarity matrix can be exported to other software and used in all kinds of clustering or dissimilarity-based algorithms, so allowing to introduce some supervision into these methods. To our knowledge, such a metrics has not been implemented before.

The FP-based semi-distance d_P, that was defined in the general case in [**?**], becomes a very simple function in the case of SFPs. Apart from its utility to represent knowledge, the function can also be used to optimize fuzzy partitions, as presented in the case study on multidimensional clustering.

d_P can be used in the statistical procedures that involve a proximity or dissimilarity computation (e. g. hierarchical clustering, data analysis, classification) but also in a wider range of applications that deal with neighborhood, such as the k-nearest-neighbors classifier or image processing algorithms. An application to spatial data zoning, based on a region growing algorithm [20], can be found in [14].
The FP-based semi-distance may be useful in any of these application areas where knowledge and data integration is an important issue.

References

[1] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Functions Algorithms*. Plenum Press, New York, 1981.

[2] I. Bloch, "On fuzzy distances and their use in image processing under imprecision," *Pattern Recognition*, vol. 32, pp. 1873–1895, 1999.

[3] S. Boriah, V. Chandola, and V. Kumar, "Similarity measures for categorical data: A comparative evaluation," in *SIAM Data Mining Conference*, Atlanta, GA, April 2008, pp. 243–254.

[4] C. Coppola and T. Pacelli, "Approximate distances, pointless geometry and incomplete information," *Fuzzy Sets and Systems*, vol. 157, pp. 2371–2383, 2006.

[5] P. Diamond and P. Kloeden, "Metric spaces of fuzzy sets," *Fuzzy Sets and Systems*, vol. 35, pp. 241–249, 1990.

[6] W. J. Dixon, *BMDP statistical software manual: to accompany the 1990 software release*, BDMP, 1990.

[7] D. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Applications*. Academic Press, New York, 1980.

[8] L. Egghe and R. Rousseau, "Classical retrieval and overlap measures satisfy the requirements for rankings based on a lorenz curve," *Information Processing and Management*, vol. 42, pp. 106–120, 2006.

[9] J. Fan and W. Xie, "Distance measure and induced fuzzy entropy," *Fuzzy Sets and Systems*, vol. 104, pp. 305–314, 1999.

[10] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml

[11] S. Guillaume and B. Charnomordic, "Generating an interpretable family of fuzzy partitions," *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 3, pp. 324–335, 2004.

[12] ——, "Learning interpretable fuzzy inference systems with fispro," *International Journal of Information Sciences*, vol. 181, pp. 4409–4427, 2011.

[13] ——, "Parameter optimization of a fuzzy inference system using the fispro open source software," in *IEEE International Conference on Fuzzy Systems*, I. C. N. CFP12FUZ-USB, Ed. Brisbane, Australia: IEEE, June 2012, pp. 402–409.

[14] S. Guillaume, B. Charnomordic, and B. Tisseyre, "Open source software for modelling using agro-environmental georeferenced data." in *IEEE International Conference on Fuzzy Systems*, I. C. N. CFP12FUZ-USB, Ed. Brisbane, Australia: IEEE, June 2012, pp. 1074–1081.

[15] S. Guillaume and L. Magdalena, "Expert guided integration of induced knowledge into a fuzzy knowledge base," *Soft computing*, vol. 10, no. 9, pp. 773–784, 2006.

[16] R. E. Hammah and J. H. Curran, "On distance measures for the fuzzy k-means algorithm for joint data," *Rock Mechanics and Rock Engineering*, vol. 32 (1), pp. 1–27, 1999.

[17] J. A. Hartigan, *Clustering Algorithms*. Wiley, 1975.

[18] L. Kaufman and P. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. New York: Wiley Interscience, 1990.

[19] R. Lowen and W. Peeters, "Distance between fuzzy sets representing grey level images," *Fuzzy Sets and Systems*, vol. 99, pp. 135–149, 1998.

[20] M. Pedroso, J. Taylor, B. Tisseyre, B. Charnomordic, and S. Guillaume, "A segmentation algorithm for the delineation of management zones," *Computer and Electronics in Agriculture*, vol. 70, no. 1, pp. 199–208, 2010.

[21] R Development Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2008, ISBN 3-900051-07-0. [Online]. Available: http://www.R-project.org

[22] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.

[23] V. Torra and Y. Narukawa, "On a comparison between mahalanobis distance and choquet integral: The choquet–mahalanobis operator," *Information Sciences*, vol. 190, no. 0, pp. 56 – 63, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025511006335