Design of Strong Fuzzy Partitions from Cuts

Corrado Mencar[1], Marco Lucarelli[1], Ciro Castiello[1], Anna M. Fanelli[2]

Dept. of Informatics, University of Bari, Italy
[1]{corrado.mencar, marco.lucarelli, ciro.castiello}@uniba.it
[2]fanelli@di.uniba.it

Abstract

The adoption of triangular fuzzy sets to define Strong Fuzzy Partitions (SFPs) is a common practice in the research community: due to their inherent simplicity, triangular fuzzy sets can be easily derived from data by applying suitable clustering algorithms. However, the choice of triangular fuzzy sets may be limiting for the modeling process. In this paper we focus on SFPs built up starting from cuts (points of separation between cluster projections on data dimensions), showing that a SFP based on cuts can always be defined by trapezoidal fuzzy sets. Different mechanisms to derive SFPs from cuts are presented and compared by employing DC*, an algorithm for extracting fuzzy information granules from classified data.

Keywords: Strong fuzzy partitions, α-cut, triangular fuzzy sets, trapezoidal fuzzy sets

1. Introduction

The key factor for the success of fuzzy logic is its ability of modeling perceptions rather than measurements. In many cases, perceptions can be expressed in natural language terms: this makes knowledge expressed in fuzzy logic highly co-intensive with linguistic concepts; hence, it is easily interpretable by users. Nevertheless, interpretability does not come with fuzzy logic *ipso facto*: it must be ensured by a number of structural and semantic constraints. More specifically, while designing an interpretable fuzzy model the data domain is represented through linguistic variables (usually one for each data feature); given a linguistic variable, the fuzzy sets associated to each linguistic term form a *fuzzy partition* of the data feature. To ensure interpretability, a number of constraints are imposed on the fuzzy sets of each fuzzy partition, like distinguishability, coverage, special elements, and so on [1].

The fulfillment of many interpretability constraints is guaranteed if Strong Fuzzy Partitions (SFPs) are adopted. Actually, SFPs are not strictly necessary for satisfying the above mentioned interpretability constraints; however, they are widely used because they simplify the modeling process as they usually require few parameters for their definition.

Triangular SFPs (TSFPs) are widely used for modeling interpretable fuzzy systems. They are characterized by the use of triangular fuzzy sets to define a fuzzy partition. Triangular fuzzy sets have a number of desirable properties, which are useful for interpretability (they are normal, convex and continuous) as well as for modeling [2]. However, triangular fuzzy sets have some non-derivable points: this prevents their use in modeling techniques that use some gradient-based learning technique to adapt fuzzy sets to available data. In such cases, usually completely differentiable fuzzy sets are used, like Gaussian fuzzy sets [3]; however, these fuzzy sets may not preserve some interpretability constraints (in particular, the proper ordering of linguistic concepts). As a consequence, triangular fuzzy sets are preferred when the modeling process does not require any gradient-based learning algorithm.

The definition of a TSFP with n fuzzy sets is completely characterized by n values that correspond to the prototypes of each fuzzy set: this makes the design of TSFPs very simple. Usually, the prototypes are computed by some algorithm that tries to locate prototypes in order to better represent the available data. As an example, Hierarchical Fuzzy Partitioning (HFP) operates an iterative merging process of triangular fuzzy sets in order to better fit available data and, simultaneously, reduce the number of fuzzy sets in a partition [4]. The merging process of two fuzzy sets is essentially computed by a weighted mean of the prototypes of the fuzzy sets to be merged; in this way the number of fuzzy sets is dynamically determined during the design process. Other approaches fix the number of triangular fuzzy sets; then the location of prototypes is determined according to some optimization process [5] or through evolutionary algorithms [6, 7].

In some cases, fuzzy partitions are designed after a clustering analysis of multidimensional data. This approach enables the discovery of multidimensional relationships among data, which can be conveniently represented as fuzzy rules [8]. To ensure interpretability, clusters are usually projected on each input feature, where fuzzy sets are defined so as to resemble as much as possible the projected clusters [9, 10]. Often, prototype-based clustering is used (like fuzzy c-means or similar): in these cases the prototypes of multidimensional clusters are projected on each input feature and could serve as pro-

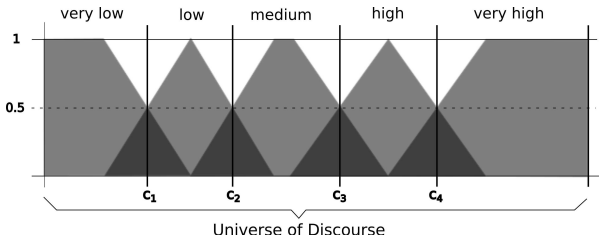


Figure 1: Example of fuzzy partition obtained from cuts c_1, c_2, c_3, c_4.

totypes of the fuzzy sets in a partition [11]. However, the simple use of multidimensional prototypes does not give enough information about the span of clusters within the data domain. For such reason, an alternative approach makes use of *cuts*, i.e. points of separation between clusters projected onto input features [12].

Cuts can be conveniently used to define the bounds of the 0.5-cuts of the fuzzy sets in a fuzzy partition[1]. More specifically, given a collection of cuts, a SFP can be defined so that the 0.5-cuts of the fuzzy sets in the partition coincide with the intervals bounded by the cuts (see fig. 1). Since the 0.5-cut of a fuzzy set is the set of elements that are most representative for the fuzzy set, then a SFP based on cuts is a robust representation of the projections of multidimensional clusters on an input feature.

In this paper we show that a SFP based on cuts *cannot* be always defined by triangular fuzzy sets. The consequences of this result impact on the flexibility of modeling approaches based on triangular fuzzy sets: imposing the use of this type of fuzzy sets restricts the possibilities of representing multidimensional relationships in an interpretable way.

In fact, the use of triangular fuzzy sets represents a further bias —which is not motivated by any interpretability requirement— to be added to the structural constraints that are already taken into account while designing a fuzzy model (as known, such constraints ultimately impose the requirement of a balance between interpretability and accuracy). In other words, the flexibility connected to a modeling process based on the employment of SFPs may be restricted by confining the choice of fuzzy sets to the triangular category.

As a consequence, interest should be shifted towards a more relevant issue concerning the possibility to define SFPs based on cuts. In this paper we show that this is feasible by resorting to trapezoidal fuzzy sets. Trapezoidal fuzzy sets are widely used for modeling interpretable fuzzy systems [7, 3, 13, 14, 15]; however, in most cases trapezoidal fuzzy sets require more parameters than triangular fuzzy sets. Such parameters need to be tuned according to some heuristic optimization process like genetic algorithms. The procedures we show in this paper do not need free parameters because trapezoidal fuzzy sets are defined given a collection of cuts only. In this way there is no need of further optimization processes beyond the clustering process that produced the cuts.

In the next Section, we provide a formal proof that triangular fuzzy sets cannot always be used to define SFPs given a set of cuts. Then we define a procedure to define SFPs based on trapezoidal fuzzy sets. In Section 3 we illustrate some examples of SFPs based on trapezoidal fuzzy sets that are derived through DC* —an algorithm for generating interpretable fuzzy partitions by using cuts— and we compare them with triangular SFPs. Some final notes are reported in Section 4.

2. Generation of Strong Fuzzy Partitions from cuts

A SFP is a collection[2] of fuzzy sets $A_1, A_2, \ldots, A_{n+1}$ defined on a Universe of Discourse $X = [m, M] \subseteq \mathbb{R}$ such that:

$$\forall x \in X : \sum_{i=1}^{n+1} A_i(x) = 1 \tag{1}$$

A triangular fuzzy set is denoted by

$$T[l, p, r]$$

where:

- l is the leftmost bound of its support;
- p is the element of its core (also called *prototype*);
- r is the rightmost bound of its support.

The membership function of a triangular fuzzy set can be conveniently defined as a case-based function:

$$T[l,p,r](x) = \begin{cases} \frac{x-l}{p-l}, & x \in]l, p] \\ \frac{x-r}{p-r} & x \in]p, r[\\ 0, & x \leq l \vee x \geq r \end{cases} \tag{2}$$

A triangular fuzzy set is *well-formed* if and only if

$$l \leq p \leq r \tag{3}$$

A Triangular Strong Fuzzy Partition (TSFP) is a SFP made with triangular fuzzy sets only[3]. A TSFP made of $n+1$ fuzzy sets is completely characterized by $n-1$ parameters p_i for $i = 2, 3, \ldots, n$. In fact, the triangular fuzzy sets of a TSFP can be defined as

$$T[p_{i-1}, p_i, p_{i+1}]$$

[1] The 0.5-cut of a fuzzy set is the (crisp) set of all elements with membership degree greater or equal to 0.5.

[2] We assume that the collection is sorted, so that it is legitimate to refer to the i-th fuzzy set in a SFP.

[3] Exceptionally, trapezoidal fuzzy sets can be defined as leftmost and rightmost fuzzy sets. However, this case can be safely ignored in the present argumentation.

for $i = 1, 2, \ldots, n+1$ with the convention that $p_0 = p_1 = m$ and $p_{n+1} = p_{n+2} = M$.

Given an element $x \in X$, at most two fuzzy sets have non-zero membership in a TSFP: these fuzzy sets are said adjacent. Furthermore, since triangular fuzzy sets are convex, their α-cuts are intervals. Given the constraint (1) of a SFP, it is immediate to verify that the 0.5-cuts of two adjacent fuzzy sets in a TSFP are also adjacent (in the sense of sharing one and only one intersection point).

Let $t_1, t_2, \ldots, t_n \in X$ a sequence of cuts, where $t_i < t_{i+1}$ for $i = 1, 2, \ldots, n-1$. In order to design a SFP based on cuts, each cut corresponds to an intersection point between two adjacent fuzzy sets in a SFP; as a consequence, n cuts correspond to the intersection points of $n+1$ fuzzy sets in a SFP. (An intersection point between two fuzzy sets is a point in X where both fuzzy sets have the same non-zero membership, see also fig. 1.)

In the following we show that it is not always possible to build a TSFP of $n+1$ fuzzy sets given an arbitrary set of n cuts. We prove this by attempting to build a TSFP and then we highlight the conditions that prevent the definition of well-formed triangular fuzzy sets. The reader can refer to fig. 2 as an illustrative example of the proof.

We suppose that a triangular fuzzy set $T[l_{i-1}, p_{i-1}, r_{i-1}]$ is defined so that

$$T[l_{i-1}, p_{i-1}, r_{i-1}](t_{i-1}) = 0.5$$

and

$$T[l_{i-1}, p_{i-1}, r_{i-1}](t_i) = 0.5$$

The membership values on t_{i-1} and t_i constrain the parameters l_{i-1} and r_{i-1}. In particular, the parameter r_{i-1} can be obtained by applying the case-based definition of a triangular fuzzy set, resulting in

$$\frac{t_i - r_{i-1}}{p_{i-1} - r_{i-1}} = 0.5 \Rightarrow r_{i-1} = 2t_i - p_{i-1}$$

The next triangular fuzzy set $T[l_i, p_i, r_i]$ must be defined so as to satisfy the constraints (1) of a SFP. The parameters of the membership function must be therefore defined as

$$l_i = p_{i-1}$$

and

$$p_i = r_{i-1} = 2t_i - p_{i-1}$$

while r_i is defined such that

$$0.5 = \frac{t_{i+1} - r_i}{p_i - r_i}$$

i.e.

$$r_i = 2\left(t_{i+1} - t_i\right) + p_{i-1}$$

In order to assure well-formedness (3), the relation

$$p_i \leq r_i$$

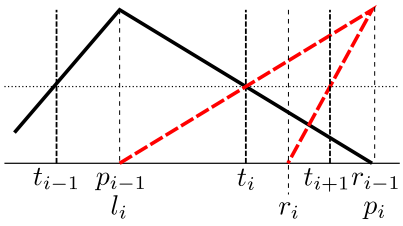


Figure 2: A sequence of cuts that prevents the generation of a well-formed triangular fuzzy set (red dashed line).

must hold. It is easy to show that this relation is true if and only if

$$t_{i+1} \geq 2t_i - p_{i-1} = r_{i-1} \tag{4}$$

Therefore, if the cuts used for partitioning do not verify (4), it is not possible to define well-formed triangular fuzzy sets.

This result has a strong impact on interpretable fuzzy modeling. In fact, if we denote by $\mathbf{T}$ the collection of all possible sets of cuts on X, and by $\mathbf{P}$ the set of all TSFPs, then relation (4) states that it is not possible to define a bijective mapping from $\mathbf{T}$ to $\mathbf{P}$. On the other hand an injective mapping from $\mathbf{P}$ to $\mathbf{T}$ is trivial: given a TSFP, the set of cuts can be defined by selecting all the intersection points between triangular fuzzy sets. Therefore, the set $\mathbf{T}$ is richer that $\mathbf{P}$, thus *any algorithm that carries out a collection of cuts is potentially more flexible and less biased than an algorithm that produces triangular SFPs.*

2.1. Generation of Trapezoidal SFPs

Is it possible to derive a SFP given a set of cuts, i.e. given an element of $\mathbf{T}$? The answer is affirmative if we resort to trapezoidal fuzzy sets instead of triangular fuzzy sets. In the following we show some procedures to derive a SFP made of trapezoidal fuzzy sets given a collection of cuts on X.

First, we recall the definition of a trapezoidal fuzzy set:

$$T\left[a, b, c, d\right](x) = \begin{cases} \frac{x-a}{b-a}, & x \in]a, b[\\ 1, & x \in [b, c] \\ \frac{x-d}{c-d} & x \in]c, d[\\ 0, & x \leq a \vee x \geq d \end{cases} \tag{5}$$

A trapezoidal fuzzy set is well formed if the relations

$$a \leq b \leq c \leq d$$

hold. Any triangular fuzzy set is a trapezoidal fuzzy set when

$$a = l \wedge b = c = p \wedge d = r$$

therefore it is possible to qualify a fuzzy set as trapezoidal even if its actual shape is triangular.

A SFP made of trapezoidal fuzzy sets $A_i = T[a_i, b_i, c_i, d_i]$, $i = 1, 2, \ldots, n+1$, requires that

$$a_{i+1} = c_i$$

and

$$b_{i+1} = d_i$$

for $i = 1, 2, \ldots, n$, as well as

$$\begin{aligned} a_1 = b_1 = m \\ c_{n+1} = d_{n+1} = M \end{aligned} \tag{6}$$

In this paper we present three approaches for designing trapezoidal SFPs. The first one (called "Constant Slope") defines trapezoidal fuzzy sets with the same slope (in absolute value). This is the simplest approach as it does not require additional knowledge for the design of a SFP. The second approach, called "Variable Fuzziness" is based on the idea that fuzzy sets with a large support are more imprecise than fuzzy sets with a small support. As a consequence, the slope of the trapezoidal fuzzy sets is defined according to the distance between two adjacent cuts. Finally, the third approach extends the second one by requiring an additional set of "Core Points", i.e. points in the domain that must belong to the core of a fuzzy set. This approach can be used when it is a-priori known that some points are representative of some concepts to be fully represented by linguistic terms.

2.1.1. Constant slope

Given a set of cuts $t_1, t_2, \ldots, t_n \in X$ it is possible to define a SFP made of trapezoidal fuzzy sets by applying the following procedure. First, the differences between cuts

$$\Delta_i = t_{i+1} - t_i$$

are computed for $i = 0, 1, \ldots, n$, with the convention that

$$t_0 = 2m - t_1$$

and

$$t_{n+1} = 2M - t_n$$

Then, the smallest difference

$$\Delta_{i_{\min}} = \min\{\Delta_i | i = 0, 1, \ldots, n\}$$

is selected with the corresponding index $i_{\min}$. (More than one index may verify this relation: in such a case the first index is selected.)

By definition, the interval $[t_{i_{\min}}, t_{i_{\min}+1}]$ is the most specific among all intervals $[t_i, t_{i+1}]$. Therefore, the most specific fuzzy set is defined, which is triangular and defined by the following parameters:

$$b_{i_{\min}} = c_{i_{\min}} = \frac{t_{i_{\min}+1} + t_{i_{\min}}}{2}$$

and

$$a_{i_{\min}} = 2t_{i_{\min}} - b_{i_{\min}} = \frac{3t_{i_{\min}} - t_{i_{\min}+1}}{2}$$

and

$$d_{i_{\min}} = 2t_{i_{\min}+1} - c_{i_{\min}} = \frac{3t_{i_{\min}+1} - t_{i_{\min}}}{2}$$

The slopes of the oblique segments in the triangular fuzzy set have the same magnitude but opposite signs. In particular, the ascending segment has slope

$$\rho^+ = \frac{1}{b_{i_{\min}} - a_{i_{\min}}} = \frac{1}{t_{i_{\min}+1} - t_{i_{\min}}}$$

while the descending segment has slope

$$\rho^- = \frac{1}{c_{i_{\min}} - d_{i_{\min}}} = \frac{1}{t_{i_{\min}} - t_{i_{\min}+1}} = -\rho^+$$

We use the slopes ρ^+ and ρ^- to define the remaining fuzzy sets. By construction, the use of these slopes assures that all trapezoidal fuzzy sets are well-formed. In fact, higher slopes (in magnitude) could be also used, while lower slopes may hamper the well-formedness of the trapezoidal fuzzy sets.

Given a cut t_i, $i = 1, 2, \ldots, n$, the following parameters are defined:

$$a_{i+1} = t_i - \frac{1}{2\rho^+}$$

$$b_{i+1} = t_i + \frac{1}{2\rho^+}$$

$$c_i = t_i + \frac{1}{2\rho^-} = a_{i+1}$$

$$d_i = t_i - \frac{1}{2\rho^-} = b_{i+1}$$

Finally, the leftmost and rightmost fuzzy sets are defined so as to be trunked at the extreme points of X. Therefore

$$a_1 = b_1 = m$$

and

$$c_{n+1} = d_{n+1} = M$$

It is easy to verify that $a_i \leq b_i$ and $c_i \leq d_i$ for each $i = 1, 2, \ldots, n+1$. Well-formedness of the trapezoidal fuzzy sets can be thus checked by verifying that $b_i \leq c_i$ for each $i = 1, 2, \ldots, n+1$. We suppose, by contradiction, that $b_i > c_i$. By construction, this means that

$$t_{i-1} + \frac{1}{2\rho^+} > t_i + \frac{1}{2\rho^-} = t_i - \frac{1}{2\rho^+}$$

which is equivalent to

$$\rho^+ < \frac{1}{t_i - t_{i-1}}$$

i.e.

$$\Delta_{i_{\min}} > \Delta_{i-1}$$

which is absurd by definition of $\Delta_{i_{\min}}$.

2.1.2. Variable fuzziness

This approach is based on the idea that the fuzziness of a fuzzy set in a partition is dependent on the amplitude of the interval between two cuts. In particular, the smaller is such amplitude, the sharper are the related fuzzy sets. Fuzziness can be quantified through the notion of entropy measure [16]; however it is easy to verify that fuzziness is related to the slopes of the trapezoidal fuzzy sets, so that high slopes lead to sharp fuzzy sets and vice versa.

The procedure for generating the trapezoidal fuzzy sets works as follows: for each $i = 0, 1, \ldots, n-1$ the values Δ_i and Δ_{i+1} are compared and the shortest is selected. (Here, t_0 is set to m and t_{n+1} is set to M to make the selection coherent with the idea underlying this approach.)

If $\Delta_i \leq \Delta_{i+1}$ then the *descending* part of the fuzzy set A_{i+1} is defined by a membership function that is highest at the center of Δ_i and gets the value 0.5 at t_{i+1}. Formally this requires that

$$c_{i+1} = \frac{t_i + t_{i+1}}{2}$$

and

$$d_{i+1} = 2t_{i+1} - c_{i+1}$$

As a consequence, the *ascending* part of the fuzzy set A_{i+2} is defined accordingly:

$$a_{i+2} = c_{i+1}$$

and

$$b_{i+2} = d_{i+1}$$

By construction, it is verified that b_{i+2} will be smaller than the midpoint of Δ_{i+1}, thus guaranteeing well-formedness of the trapezoidal fuzzy set.

If $\Delta_i > \Delta_{i+1}$ the scheme is inverted and the ascending part of A_{i+2} is first defined by setting

$$b_{i+2} = \frac{t_{i+1} + t_{i+2}}{2}$$

and

$$a_{i+2} = 2t_{i+1} - b_{i+2}$$

Then, the descending part of A_{i+1} is defined accordingly:

$$c_{i+1} = a_{i+2}$$

and

$$d_{i+1} = b_{i+2}$$

Finally, the undefined parts of the leftmost and rightmost fuzzy set are set as in (6).

2.1.3. Core points

This approach exploits additional information to define the SFP. In particular, it is assumed that in each interval between two cuts a finite and non-empty set of points $P_i \subset [t_i, t_{i+1}]$ is available, with the constraint that such points must belong to the core of the corresponding fuzzy set in the partition[4].

The procedure for generating the trapezoidal fuzzy sets is similar to that defined for variable fuzziness. More specifically, for each P_i the minimum and maximum elements are considered, i.e.

$$p_i^{\min} = \min P_i$$

and

$$p_i^{\max} = \max P_i$$

Furthermore, the distances between such points and the cuts are considered:

$$\delta_i^{\text{left}} = t_i - p_{i-1}^{\max}$$

and

$$\delta_i^{\text{right}} = p_i^{\min} - t_i$$

for $i = 1, 2, \ldots, n$.

For each i the values of δ_i^{left} and δ_i^{right} are compared: if $\delta_i^{\text{left}} \leq \delta_i^{\text{right}}$ then

$$a_{i+1} = c_i = p_{i-1}^{\max}$$

and

$$b_{i+1} = d_i = 2t_i - c_i$$

otherwise

$$d_i = b_{i+1} = p_i^{\min}$$

and

$$c_i = a_{i+1} = 2t_i - b_{i+1}$$

3. Example: DC*

One of the algorithms that uses cuts to generate fuzzy partitions from data is DC* (Double Clustering with A*) [17, 18]. In this Section we give a brief outline of DC* and then we present some results on the use of DC* along with the different approaches for generating SFPs.

3.1. Outline of DC*

DC* (Double Clustering with A*) is an algorithm conceived for extracting interpretable fuzzy information granules from classified data. Such information granules are represented through interpretable fuzzy partitions and can be used to define a set of fuzzy classification rules.

In essence, DC* works in three consecutive steps. In the first step, a collection D of classified data in a multi-dimensional domain

$$\mathbf{X} = [m_1, M_1] \times \cdots \times [m_d, M_d]$$

is compressed through a vector quantization algorithm. (LVQ1 [19] is used in the current version of DC*.) The resulting codebook

$$\mathbf{p_1}, \mathbf{p_2}, \ldots \mathbf{p_c} \in \mathbf{X}$$

[4]The core of a fuzzy set is the (crisp) set of all elements with full membership.

consists of c multi-dimensional classified prototypes. (The parameter c is user-selected.) All the prototypes are projected onto each dimension, so that for dimension h, c classified one-dimensional prototypes

$$p_{h1}, p_{h2}, \ldots p_{hc} \in [m_h, M_h]$$

are available.

In the second step, DC* operates a clustering process of one-dimensional prototypes in all dimensions *simultaneously*. The objective of this clustering process is to carry out a set of cuts

$$t_{h1}, t_{h2}, \ldots t_{hn} \in [m_h, M_h]$$

for each dimension so that any interval $[t_{h,i}, t_{h,i+1}]$ includes one-dimensional prototypes of the same class. Furthermore, the *hyper-boxes*

$$[t_{1,i_1}, t_{1,i_1+1}] \times [t_{2,i_2}, t_{2,i_2+1}] \times \cdots \times [t_{d,i_d}, t_{d,i_d+1}]$$

include multi-dimensional prototypes of the same class only. (The number of such prototypes can be zero.) Finally, the number of hyper-boxes containing prototypes is minimal. To achieve this complex objective, the clustering of the projections addressed in the second step is defined as a combinatorial optimization problem, which is faced by resorting to the A* search algorithm [20].

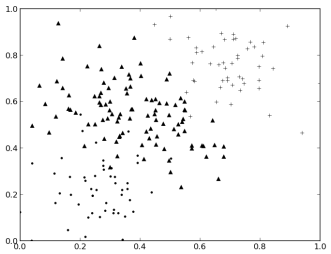
DC*, therefore, stands as a convenient approach to produce data clustering making use of *cuts*, represented by the midpoints between two adjacent projections of prototypes belonging to different classes. Furthermore, DC* produces a number of core-points, which correspond to the one-dimensional prototypes and are directly related to the compressed representation of data.

The optimal configuration of cuts identified by DC* represents the starting point for a modeling procedure devoted to define a SFP for each input feature based on trapezoidal fuzzy sets[5], which corresponds the the last step of DC*. Furthermore, the inherent working engine of DC* is oriented to produce additional pieces of information, namely the prototypes identified by the LVQ1 algorithm. Therefore, DC* represents a suitable procedure to design SFP based on cuts and core points (in line with the approach described in Section 2.1.3).

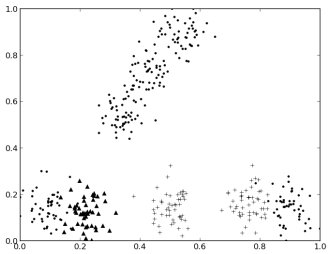
3.2. Simulation on numerical data

We recently evaluated DC* with HFP on a number of benchmark datasets for the sake of comparison; we observed that, on the average, DC* exhibits a superior behaviour in terms of accuracy/interpretability tradeoff [21]. The objective of this simulation, instead, is to evaluate the DC* behaviour when different strategies for generating SFPs are adopted. Actually, the current version of

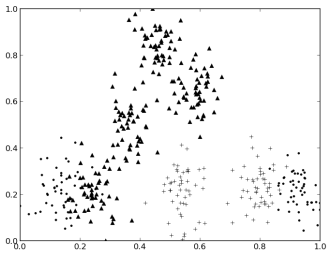
[5]The original version of DC* produced Gaussian fuzzy sets.



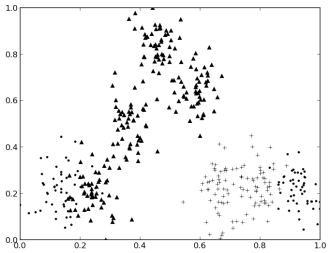
SD1



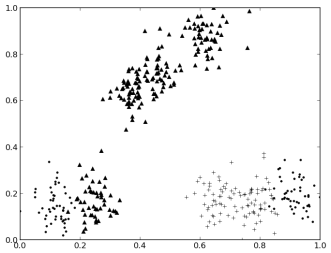
SD2



SD3



SD4



SD5

Figure 3: The synthetically generated datasets adopted for the numerical simulation.

Table 1: DC* classification error (percentage values) when different strategies are applied to generate fuzzy partitions for each of the five datasets.

	SD1	SD2	SD3	SD4	SD5
CS	17.50	11.75	16.50	13.50	8.00
VF	11.00	7.00	11.00	6.50	3.50
CP	9.00	7.00	8.75	4.75	3.00
TSFP	44.00	9.50	17.75	9.00	4.50
T0.5-cuts	20.50	16.50	18.00	19.50	18.50

DC* adopts the *variable fuzziness* approach to derive trapezoidal SFPs, but it can be easily modified to generate SFPs with all the strategies that have been presented in the previous section.

To this aim, we used a set of synthetically generated datasets: one of them (SD1) consists of 200 bi-dimensional examples, the other four datasets (SD2, SD3, SD4, SD5) consist of 400 bi-dimensional examples. In each case, the samples belong to 3 different classes. The datasets are depicted in fig. 3.

DC* has been employed to process the data. The initial clustering has been performed considering 24 multi-dimensional prototypes for SD1 and 48 multi-dimensional prototypes for SD2–SD5. (The prototypes are proportionally distributed according to the number of samples for each class.) The final fuzzy partitions have been derived by alternatively applying the previously described procedures: Constant Slope (CS), Variable Fuzziness (VF) and Core Points (CP). Additionally, two more strategies have also been tested, oriented to the generation of triangular fuzzy partitions. In the first case (TSFP), SFPs have been obtained by partially exploiting the information coming from cuts: the design of the triangular fuzzy sets is such that their core points correspond to the midpoints of the intervals defined by the cuts. In the second case (T0.5-cuts), the triangular fuzzy sets are shaped so that the membership values in $t_1, \ldots, t_n$ are set at 0.5. (As shown in section 2, the latter mechanism leaves no guarantee to derive a SFP for sure.)

Table 1 reports the performance (in terms of percentage of classification error) of DC* for each adopted strategy. It can be verified that for each dataset the best performance is attained by applying the Core Points strategy. In general, resorting to triangular fuzzy partitions means a deterioration in the classification error values.

More interestingly, fig. 4 depicts the different fuzzy partitions produced by DC* when the above mentioned strategies are applied. We show here the configurations related to the clustering processes performed over one of the synthetic datasets (namely, SD4); for the sake of conciseness, only one input feature is considered in the figures. It is important to highlight how the choice for a triangular fuzzy partition forced to express a 0.5 value at the cuts points gives rise to a configuration which does not satisfy the SFP conditions. On the other hand, the fuzzy partion provided through the CP approach gives a tangible idea on the fuzziness of the linguistic terms in accordance with the core points provided by DC*: it is apparent that fuzziness is acceptable in the right side of the Universe of Discourse, while crisper linguistic terms are required to discriminate data in the center and left side.

4. Conclusions

The definition of fuzzy partition represents a key issue for designing interpretable fuzzy models since fuzzy partitions are often required to fulfill several interpretability constraints. In this sense, Strong Fuzzy Partitions (SFPs) are commonly adopted as a reliable tool to design interpretable models, and triangular SFPs are often preferred because they can be easily derived through some clustering mechanism performed over the available data.

In this paper we considered a particular approach for defining SFPs which is based on cuts, that are points of separation between cluster projections on data dimensions. We dealt with the problem of identifying the proper shape of fuzzy sets while generating SFPs from cuts, highlighting how the choice of triangular fuzzy sets represents an additional bias for the modeling process which can be conveniently removed by resorting to trapezoidal fuzzy sets.

Through some numerical simulations that make use of DC*, a cut-based algorithm for generating fuzzy partitions, we showed that the use of trapezoidal fuzzy sets enables the derivation of highly interpretable fuzzy partitions that are more accurate than triangular fuzzy partitions in classification tasks.

References

[1] Corrado Mencar and Anna M Fanelli. Interpretability constraints for fuzzy information granulation. *Information Sciences*, 178(24):4585–4618, 2008.

[2] Witold Pedrycz. Why triangular membership functions? *Fuzzy Sets and Systems*, 64(1):21–30, May 1994.

[3] Shang-Ming Zhou and John Q. Gan. Extracting Takagi-Sugeno Fuzzy Rules with Interpretable Submodels via Regularization of Linguistic Modifiers. *IEEE Transactions on Knowledge and Data Engineering*, 21(8):1191–1204, August 2009.

[4] S. Guillaume and B. Charnomordic. Generating an Interpretable Family of Fuzzy Partitions From Data. *IEEE Transactions on Fuzzy Systems*, 12(3):324–335, June 2004.

[5] Cheng-Liang Chen, Sheng-Nan Wang, Chung-Tyan Hsieh, and Feng-Yuan Chang. Theoretical analysis of a fuzzy-logic controller with

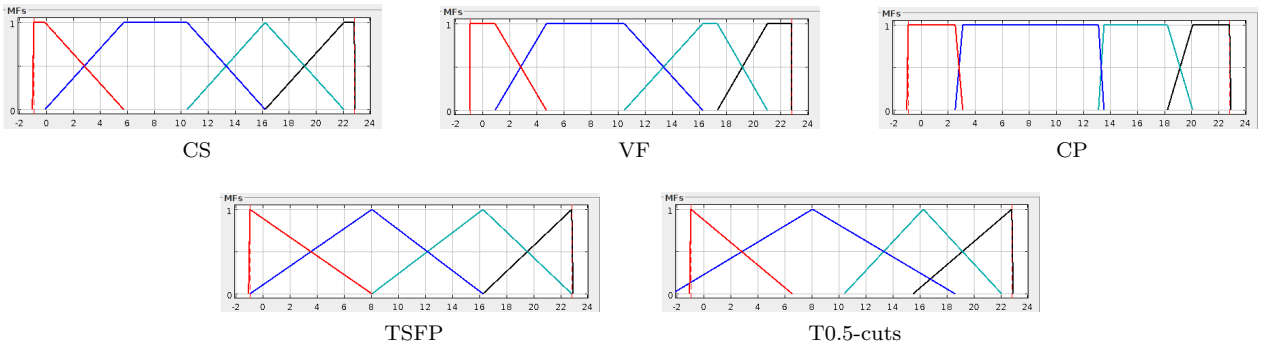


Figure 4: Fuzzy partitions obtained for a single input feature by DC* through the adoption of different strategies.

unequally spaced triangular membership functions. *Fuzzy Sets and Systems*, 101(1):87–108, January 1999.

[6] Yau-Tarng Juang, Yun-Tien Chang, and Chih-Peng Huang. Design of fuzzy PID controllers using modified triangular membership functions. *Information Sciences*, 178(5):1325–1333, March 2008.

[7] J. Casillas. Embedded genetic learning of highly interpretable fuzzy partitions. In *Proceedings of IFSA-EUSFLAT*, pages 1631–1636, Lisbon, Portugal, 2009.

[8] J. Abonyi, R. Babuska, and F. Szeifert. Modified Gath-Geva fuzzy clustering for identification of Takagi-Sugeno fuzzy models. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics*, 32(5):612–21, January 2002.

[9] F. Klawonn and A Keller. Fuzzy clustering and fuzzy rules. *Proceedings of the 7th International Fuzzy Systems Association World Congress {(IFSA'97)}*, pages 193–198, 1997.

[10] J. Abonyi, H. Roubos, R. Babuska, and F. Szeifert. Interpretable Semi-Mechanistic Fuzzy Models by Clustering, {OLS} and {FIS} Model Reduction. In J. Casillas, O. Cordon, F. Herrera, and L. Magdalena, editors, *Interpretability Issues in Fuzzy Modeling*, pages 221–248. Springer-Verlag, Heidelberg, 2003.

[11] P. Angelov. An approach for fuzzy rule-base adaptation using on-line clustering. *International Journal of Approximate Reasoning*, 35(3):275–289, March 2004.

[12] G Castellano, AM Fanelli, and Corrado Mencar. DCf: a double clustering framework for fuzzy information granulation. In *2005 IEEE International Conference on Granular Computing*, volume 2, pages 397–400. IEEE, 2005.

[13] Magne Setnes and Hans Roubos. Transparent fuzzy modeling using fuzzy clustering and GAs. In IEEE, editor, *18th International Conference of the North American Fuzzy Information Processing Society - NAFIPS*, pages 198–202, New York, 1999. IEEE.

[14] J.M. Alonso and L. Magdalena. HILK++: an interpretability-guided fuzzy modeling methodology for learning readable and comprehensible fuzzy rule-based classifiers. *Soft Computing*, 15(10):1959–1980, June 2011.

[15] S. Guillaume. Designing fuzzy inference systems from data: An interpretability-oriented review. *IEEE Transactions on Fuzzy Systems*, 9(3):426–443, June 2001.

[16] B.R. Ebanks. On measures of fuzziness and theri representations. *Journal of Mathematical Analysis and Applications*, 94:24—-37, 1983.

[17] Giovanna Castellano, Anna Maria Fanelli, Corrado Mencar, and Vito Leonardo Plantamura. Classifying data with interpretable fuzzy granulation. In *Proceedings of the 3rd International Conference on Soft Computing and Intelligent Systems and 7th International Symposium on Advanced Intelligent Systems 2006*, pages 872–877, Tokyo, Japan, 2006.

[18] Corrado Mencar, Arianna Consiglio, Giovanna Castellano, and Anna Maria Fanelli. Improving the Classification Ability of DC* Algorithm. In Francesco Masulli, Sushmita Mitra, and Gabriella Pasi, editors, *Applications of Fuzzy Sets Theory (7th International Workshop on Fuzzy Logic and Applications, WILF 2007, Proceedings)*, volume 4578, pages 145–151. Springer Berlin / Heidelberg, 2007.

[19] T. Kohonen. *Self-organizing maps*, volume 30 of *Information Sciences*. Springer Verlag, 2001.

[20] S Edelkamp and S Schrödl. *Heuristic Search: Theory and Applications*. Morgan Kaufmann, 2011.

[21] Marco Lucarelli, Ciro Castiello, Anna M. Fanelli, and Corrado Mencar. Automatic design of interpretable fuzzy partitions with variable granularity: an experimental comparison. in press on: Proceedings of ICAISC2013.