Available online at www.sciencedirect.com
ScienceDirect
Fuzzy Sets and Systems 158 (2007) 2057–2077
FUZZY sets and systems
www.elsevier.com/locate/fss

Extraction of fuzzy rules from support vector machines

J.L. Castro[a], L.D. Flores-Hidalgo[b], C.J. Mantas[a,*], J.M. Puche[a]

[a] *Department of Computer Science and A.I., University of Granada, Spain*
[b] *School of Mathematics, Central University of Venezuela, Venezuela*

Received 29 November 2005; received in revised form 1 December 2006; accepted 17 April 2007
Available online 21 April 2007

Abstract

The relationship between support vector machines (SVMs) and Takagi–Sugeno–Kang (TSK) fuzzy systems is shown. An exact representation of SVMs as TSK fuzzy systems is given for every used kernel function. Restricted methods to extract rules from SVMs have been previously published. Their limitations are surpassed with the presented extraction method. The behavior of SVMs is explained by means of fuzzy logic and the interpretability of the system is improved by introducing the λ-fuzzy rule-based system (λ-FRBS). The λ-FRBS exactly approximates the SVM's decision boundary and its rules and membership functions are very simple, aggregating the antecedents with uninorms as compensation operators. The rules of the λ-FRBS are limited to two and the number of fuzzy propositions in each rule only depends on the cardinality of the set of support vectors. For that reason, the λ-FRBS overcomes the course of dimensionality and problems with high-dimensional data sets are easily solved with the λ-FRBS.

Keywords: Support vector machines; Fuzzy rule-based systems; Uninorms

1. Introduction

Support vector machines (SVMs) are learning systems which solve two-class classification problems. They are based on statistical learning theory and have attracted increasing attention because of their optimal applications in speaker verification and identification [38], face detection [30] or text categorization [16] among others.

A problem of the SVMs is their limitation of being Black Boxes. It cannot be explained, in a comprehensible way, how a SVM works. A similar issue happens with the artificial neural networks. Many papers have been published about bringing that problem off in the neural networks case, a review can be found in [2,14,21,28,37]. However, only a few papers have been presented in this matter concerning SVMs [3,11,12,29].

In the field of neural networks, the *Black Box* problem has been solved by finding a comprehensible representation for the network, usually a rule-based system. We mention two approaches:

(a) Obtaining an understandable system which approximates the neural network behavior. Examples of this approach can be found in [1,18,35].

* Corresponding author.
E-mail addresses: castro@decsai.ugr.es (J.L. Castro), lflorhid@yahoo.es (L.D. Flores-Hidalgo), cmantas@decsai.ugr.es (C.J. Mantas), puche@decsai.ugr.es (J.M. Puche).

0165-0114/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.fss.2007.04.014

(b) Describing the ANN action as comprehensible as possible. Usually, these systems use fuzzy rules to achieve this equivalence. The methods presented in [4,9,25] are examples of this orientation.

Papers conceived to solve the *Black Box* problem of the SVMs, also present these two approaches:

(a) Some of them show a comprehensible system with an input–output mapping different from the mapping produced by the SVM [3,29].
In [29], a clustering algorithm and the output decision function of the SVM are used for determining prototype vectors of each class. Then, these vectors are combined with the support vectors (SVs) to define ellipsoids in the input space, which are then mapped to if-then rules.
In [3], the classes of the data set are labeled by using a trained SVM. Then, that data set is used with a machine learning technique with explanation capability. In this paper, a standard decision tree is used.

(b) Other papers [11,12] obtain a fuzzy rule-based system (FRBS) with an input–output mapping equivalent to the decision function of the SVM. They obtain an additive fuzzy system [27] from a SVM. However, the fuzzy system is only conceived for translation invariant kernel functions, for example, *Gaussian*. In the present paper, this constraint is avoided.

Our proposal solves the Black Box problem of the SVMs by considering each SVM as a fuzzy rule-based model. The current issue has been previously faced with another Black Box models [22,31]. The presented method can be considered a method of interpretability improvement in fuzzy modeling [7]. It will be shown that it is possible to obtain a λ-FRBS provided any SVM. This λ-FRBS will have the following properties:

- The equivalence between SVM and λ-FRBS is theoretically demonstrated.
- For every SVM there is a λ-FRBS, it is not restricted by the type of kernel. The extraction method can be used for every widely used kernel: hyperbolic tangent, polynomial and Gaussian.
- It avoids the "curse of dimensionality'', which shows up when high-dimensional data problem is intended to be solved by using a standard fuzzy system. In this work, the number of rules in the λ-FRBS is limited (just two rules) and the number of fuzzy propositions in each rule only depends on the cardinality of the set of SVs.

The paper is structured as follows: in the first sections several useful concepts necessary to describe the λ-FRBS are introduced: SVMs (Section 2), Takagi–Sugeno–Kang (TSK) FRBSs (Section 3) and the uninorms (Section 4). The equivalence between SVMs and λ-FRBSs is shown in Section 5 and the extraction of comprehensible fuzzy rules from SVMs is presented as well. Finally, some examples are studied and proofs of theorems are found in the Appendix.

2. Support vector machines

Let us consider a two-class classification task with the training data set $(\vec{x}_i, y_i)$ $i = 1, \ldots, m$, where $\vec{x}_i \in \Re^n$ and $y_i = \{-1, +1\}$. Let the decision function be

$$f(\vec{x}) = \text{sign}(\langle \vec{x}, \vec{w} \rangle + b).$$

A good generalization is achieved by maximizing the margin between the separating hyperplane

$$\langle \vec{x}, \vec{w} \rangle + b = 0$$

and the closest data points in the input space. This optimal hyperplane can be determined as follows:

Minimize $\langle \vec{w}, \vec{w} \rangle$

Subject to $y_i(\langle \vec{w}, \vec{x}_i \rangle + b) \geqslant 1 \quad \forall i.$

Introducing Lagrange multipliers to solve this optimization problem and making some substitutions, we arrive to the Wolfe dual of the optimization problem:

Maximize

$$Q(\alpha) = \sum_{i,k=1}^{m} \alpha_i - \frac{1}{2} \alpha_i y_i \alpha_k y_k \langle \vec{x}_i \vec{x}_k \rangle$$

subject to

$$C \geqslant \alpha_i \geqslant 0 \ \forall i, \quad \sum_{i=1}^{m} \alpha_i y_i = 0.$$

The hyperplane decision function can thus be written as

$$f(\vec{x}) = \operatorname{sign}\left(\sum_{i=1}^{m} \alpha_i y_i \langle \vec{x}, \vec{x}_i \rangle + b\right).$$

In order to use the method to produce nonlinear decision functions, the input space is projected to a higher-dimensional inner product space F, called feature space, using a nonlinear map

$$\phi(\vec{x}) : \Re^{\mathrm{n}} \to \Re^{\mathrm{d}}.$$

In the feature space the optimal hyperplane is derived. Nevertheless, by using kernels which satisfy the Mercer's theorem, it is possible to make all the necessary operations in the input space by using

$$\langle \phi(\vec{x}_i), \phi(\vec{x}_j) \rangle = K(\vec{x}_i, \vec{x}_j),$$

as $K(\vec{x}_i,\vec{x}_j)$ is an inner product in the feature space. The decision function can be written in terms of these kernels:

$$f(\vec{x}) = \operatorname{sign}\left(\sum_{i=1}^{m} \alpha_i y_i K(\vec{x}, \vec{x}_i) + b\right).$$

Just a few of the training patterns have a weight α_i non-zero in the previous equation. These elements are the closest to the boundary and they are known as SVs. These SVs optimize the procedure in the classification task. We recommend [6,32] for an extensive explanation of learning (training) for SVMs.

The equivalence between SVMs and λ-FRBSs proposed in this paper holds for all the most commonly used kernels. These functions are:

- Hyperbolic tangent kernel: $K(\vec{x}, \vec{x}_i) = \tanh(a \cdot \langle \vec{x}, \vec{x}_i \rangle + c), \ a \cdot c \leqslant 0, \ c \leqslant 0.$
- Polynomial kernel with odd exponent: $K(\vec{x}, \vec{x}_i) = (\langle \vec{x}, \vec{x}_i \rangle + s)^d$, d is odd.
- Polynomial kernel with even exponent: $K(\vec{x}, \vec{x}_i) = (\langle \vec{x}, \vec{x}_i \rangle + s)^d$, d is even.
- Gaussian kernel: $K(\vec{x}, \vec{x}_i) = \mathrm{e}^{-(\|\vec{x}-\vec{x}_i\|^2/2\sigma^2)}$.

3. TSK fuzzy rule-based systems

The rules of the TSK FRBSs [36] usually have the following form:

$$R_k\text{: If } x_1 \text{ is } A_1 * x_2 \text{ is } A_2 * \cdots * x_n \text{ is } A_n \text{ then } Y_k = p_n \cdot x_n + p_{n-1} \cdot x_{n-1} + \cdots + p_1 \cdot x_1 + p_0,$$

where x_i are the system input variables, A_i are labels with associated fuzzy sets and Y is the output variable. The output Y of a FRBS with m TSK rules is computed as the weighted average of the individual rule outputs Y_i $(i = 1, \ldots, m)$ as follows:

$$Y = \frac{\sum_{i=1}^{m} Y_i \cdot g_i}{\sum_{i=1}^{m} g_i},$$

where $g_i = *(\mu_{A_1}(x_1), \ldots, \mu_{A_n}(x_n))$ is the matching degree between the antecedent part of the rule and the current system inputs, $*$ is usually a t-norm and $\vec{x} = (x_1, x_2, \ldots, x_n)$ is the system input.

4. Uninorms

Normally, connectives used in fuzzy rules for aggregating propositions are t-norms (fuzzy intersection, *and* connective) or t-conorms (fuzzy union, *or* connective) [24,42]. However, when these operators are used, no compensation between small and large degrees of membership takes place [23,41,43].

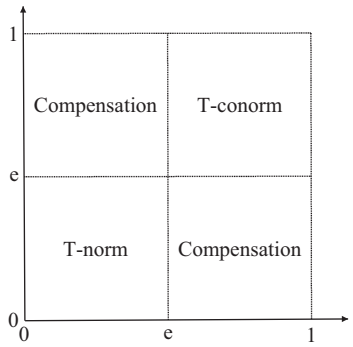

Fig. 1. Behavior of the uninorm operators.

t-Norms do not allow low values to be compensated by high values and t-conorms do not allow high values to be compensated by low values [13]. To show this problem, we can suppose the following example:

> When evaluating n features of a car (security, comfort, acceleration ...) it is obtained n values $x_i \in [0, 1]$. Each value indicates the quality of the feature evaluated (0 is bad quality—0.5 neuter quality—1 is good quality). We need to aggregate these values x_i to come to a global conclusion ($y \in [0, 1]$) about the car quality.

Let us choose the minimum operator as a t-norm (x_1 AND $x_2 \ldots$ AND x_n) to aggregate the features of the car: even though the characteristics were good (high values), only one low value will produce a low final conclusion about the car quality.

In the same way, if we take the maximum operator as a t-conorm (x_1 OR x_2 ... OR x_n): disregarding the characteristics were bad (low values), only one high value will produce a good final conclusion.

Uninorms operators were defined to solve this problem. Formally, an *uninorm* is a function

$$\mathrm{U} : [0, 1] \times [0, 1] \to [0, 1]$$

that has the following properties:

- commutability, $\mathrm{U}(x, y) = \mathrm{U}(y, x)$;
- monotonic (increasing), if $a \leqslant b$ and $c \leqslant d$ then $\mathrm{U}(a, c) \leqslant \mathrm{U}(b, d)$;
- associativity $\mathrm{U}(x, \mathrm{U}(y, z)) = \mathrm{U}(\mathrm{U}(x, y), z)$;
- it has a neuter element $e \in [0, 1]$ such as $\mathrm{U}(x, e) = x$.

The most interesting property of uninorms is its different behavior on particular sub-domains (Fig. 1):

- They behave as t-norm on the interval $([0, e] \times [0, e])$.
- They behave as t-conorm on $([e, 1] \times [e, 1])$.
- They have a compensation behavior on $([0, e) \times (e, 1] \cup (e, 1] \times [0, e))$.

In simple words, if an uninorm operator is implemented on $[0, 1] \times [0, 1]$, two small degrees of membership are scaled down, two large degrees are graded up, and some compensation takes place if small and large degrees are aggregated [23].

This behavior is coherent with some real situations. For example, if we aggregate the n values x_i about the car quality with an uninorm, the following reasoning can be carried out:

- The car problems (values $x_i \in [0, 0.5)$) are aggregated with a t-norm, obtaining only one low value $y_{\text{disadvantages}}$.
- The advantages of the car (values $x_i \in (0.5, 1]$) are aggregated with a t-conorm, obtaining only one high value $y_{\text{advantages}}$.
- The neuter features ($x_i = 0.5$) do not influence on the conclusion.
- Finally, a compensation between disadvantages ($y_{\text{disadvantages}} \in [0, 0.5)$) and advantages ($y_{\text{advantages}} \in (0.5, 1]$) of the car takes place and a final value y is obtained.

A particular uninorm is the *symmetric sum* [34] defined as follows:

$$a * b = \frac{a \cdot b}{a \cdot b + (1-a) \cdot (1-b)}.$$

Its domain is the unit square with the exception of the points (0,1) and (1,0). The neuter element of this operator is 0.5 ($e = 0.5$).

As illustrative example, the *symmetric sum* has the following behavior when combining the values $\alpha, \beta \in (0.0, 0.5)$, $\chi, \delta \in (0.5, 1.0)$ and $\varepsilon = 0.5$:

$$\alpha * \beta * \chi * \delta * \varepsilon = \alpha * \beta * \chi * \delta = \begin{matrix} (\chi \ OR \ \delta) \\ compensation \\ (\alpha \ AND \ \beta) \end{matrix} = \begin{matrix} (\chi \ OR \ \delta) \\ comp \\ (\alpha \ AND \ \beta) \end{matrix}$$

where:

- The AND operator behaves on (0,0.5) as t-norm with neuter element equal to 0.5 instead of 1.0.
- The OR operator behaves on (0.5,1.0) as t-conorm with neuter element equal to 0.5 instead of 0.0.
- The *comp* operator carries out a *compensation* between a low value $\eta = (\alpha \text{ AND } \beta)$, $\eta \in (0.0, 0.5)$ and a high value $\varphi = (\chi \text{ OR } \delta)$, $\varphi \in (0.5, 1.0)$, that is:
  - $(\eta \ comp \ \varphi)$ is lower than 0.5 if "$(0.5 - \eta) > (\varphi - 0.5)$".
  - $(\eta \ comp \ \varphi)$ is greater than 0.5 if "$(0.5 - \eta) < (\varphi - 0.5)$".
  - $(\eta \ comp \ \varphi)$ is equal to 0.5 if "$(0.5 - \eta) = (\varphi - 0.5)$".

The *symmetric sum* operator will be used for combining the fuzzy propositions in the antecedents of the rules obtained from SVMs.

5. SVMs are fuzzy rule-based systems

Let f be the decision function of a trained SVM. The set of SVs $\{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_m\}$, the parameters α_i (for $1 \leqslant i \leqslant m$) and b are fixed after the training of the SVM and we have, then, the following decision function:

$$f(\vec{x}) = \text{sign}(h(\vec{x})),$$

where

$$h(\vec{x}) = \sum_{i=1}^{m} \alpha_i y_i K(\vec{x}, \vec{x}_i) + b.$$

Theorem 1. *Every SVM with decision function* $\boldsymbol{f}$ *defined by*

$$f(\vec{x}) = \text{sign}\left(\sum_{i=1}^{m} \alpha_i y_i K(\vec{x}, \vec{x}_i) + b\right) = \text{sign}(h(\vec{x}))$$

is equivalent to the following TSK FRBS:

R_1: ***If*** $h(\vec{x})$ *is* $I_{(0,\infty)}(x)$ ***then*** $Y_1 = 1$,
R_2: ***If*** $h(\vec{x})$ *is* $I^*_{(0,\infty)}(x)$ ***then*** $Y_2 = -1$,

where

$$I_{(0,\infty)}(x) = \begin{cases} 1 & \text{if } x \in (0, \infty) \\ 0 & \text{if } x \in (-\infty, 0) \end{cases} \quad \text{and} \quad I^*_{(0,\infty)}(x) = 1 - I_{(0,\infty)}(x).$$

In order to improve interpretability, it is defined another FRBS which approximates SVMs and allows us to find a FRBS with simple propositions in the antecedents.

Theorem 2. *Every SVM with decision function* $\boldsymbol{f}$ *defined by*

$$f(\vec{x}) = \operatorname{sign}\left(\sum_{i=1}^{m} \alpha_i y_i K(\vec{x}, \vec{x}_i) + b\right) = \operatorname{sign}(h(\vec{x}))$$

is equivalent to the following FRBS:

R_1: ***If*** $h(\vec{x})$ *is* $Sigm(\lambda \cdot x)$ ***then*** $Y_1 = 1$,
R_2: ***If*** $h(\vec{x})$ *is* $Sigm^*(\lambda \cdot x)$ ***then*** $Y_2 = -1$,

when $\lambda \to \infty$, *where* $Sigm(\lambda \cdot x) = 1/(1 + \mathrm{e}^{-\lambda \cdot x})$ *and* $Sigm^*(\lambda \cdot x) = 1 - Sigm(\lambda \cdot x) = Sigm(-\lambda \cdot x)$.

Definition 1. The FRBS given in Theorem 2 is called λ-**FRBS**.

When the λ-FRBS is implemented in the experiments, the value λ is moderately high. Thus, the *sigmoid* function is not quickly saturated to one or zero according to the limited precision of the computer.

We have obtained a λ-FRBS from a SVM with only one proposition in the antecedent of the fuzzy rules. To improve its interpretability, the antecedent of each fuzzy rule will be transformed to get several simple propositions.

5.1. Several simple fuzzy propositions in the antecedents

To get several fuzzy propositions in the antecedents of the rules from a λ-FRBS, we need the following result.

Proposition 1. *Let* $*$ *be the operator*

$$a * b = \frac{a \cdot b}{a \cdot b + (1-a) \cdot (1-b)}$$

and $\beta \in \Re$. *The following equality holds*:

$$Sigm(\beta \cdot (x + y)) = Sigm(\beta \cdot x) * Sigm(\beta \cdot y).$$

The operator $*$ is the uninorm "*symmetric sum*" aforementioned in Section 4 [34,40],

$$a * b = \frac{a \cdot b}{a \cdot b + (1-a) \cdot (1-b)}.$$

This result holds to the function $Sigm^*(x)$ because of:

$$\begin{aligned} Sigm^*(\beta \cdot (x + y)) &= Sigm((-\beta) \cdot (x + y)) \\ &= Sigm((-\beta) \cdot x) * Sigm((-\beta) \cdot y) = Sigm^*(\beta \cdot x) * Sigm^*(\beta \cdot y). \end{aligned}$$

If one considers the nature of $h(\vec{x})$ in Theorem 2 we have:

R_1: **If** $\sum_{i=1}^{m} \alpha_i y_i K(\vec{x}, \vec{x}_i) + b$ is $Sigm(\lambda \cdot x)$ **then** $Y_1 = 1$,
R_2: **If** $\sum_{i=1}^{m} \alpha_i y_i K(\vec{x}, \vec{x}_i) + b$ is $Sigm^*(\lambda \cdot x)$ **then** $Y_2 = -1$.

According to Proposition 1, this λ-FRBS can be transformed into:

R_1: **If** $\alpha_1 y_1 K(\vec{x}, \vec{x}_1)$ is $Sigm(\lambda \cdot x) * \alpha_2 y_2 K(\vec{x}, \vec{x}_2)$ is $Sigm(\lambda \cdot x) * \cdots *$
$\alpha_m y_m K(\vec{x}, \vec{x}_m)$ is $Sigm(\lambda \cdot x) * b$ is $Sigm(\lambda \cdot x)$
then $Y_1 = 1$,

R_2: **If** $\alpha_1 y_1 K(\vec{x}, \vec{x}_1)$ is $Sigm^*(\lambda \cdot x) * \alpha_2 y_2 K(\vec{x}, \vec{x}_2)$ is $Sigm^*(\lambda \cdot x) * \cdots *$
$\alpha_m y_m K(\vec{x}, \vec{x}_m)$ is $Sigm^*(\lambda \cdot x) * b$ is $Sigm^*(\lambda \cdot x)$
then $Y_2 = -1$

which is equivalent to:

R_1: **If** $K(\vec{x}, \vec{x}_1)$ is $Sigm(\lambda \cdot x \cdot \alpha_1 \cdot y_1) * K(\vec{x}, \vec{x}_2)$ is $Sigm(\lambda \cdot x \cdot \alpha_2 \cdot y_2) * \cdots *$
$K(\vec{x}, \vec{x}_m)$ is $Sigm(\lambda \cdot x \cdot \alpha_m \cdot y_m) * b$ is $Sigm(\lambda \cdot x)$
then $Y_1 = 1$,

R_2: **If** $K(\vec{x}, \vec{x}_1)$ is $Sigm^*(\lambda \cdot x \cdot \alpha_1 \cdot y_1) * K(\vec{x}, \vec{x}_2)$ is $Sigm^*(\lambda \cdot x \cdot \alpha_2 \cdot y_2) * \cdots *$
$K(\vec{x}, \vec{x}_m)$ is $Sigm^*(\lambda \cdot x \cdot \alpha_m \cdot y_m) * b$ is $Sigm^*(\lambda \cdot x)$
then $Y_2 = -1$.

On the other hand, $K(\vec{x}, \vec{x}_j)$ can be seen as $K(\mathrm{O}(\vec{x}, \vec{x}_j))$ where $\mathrm{O}(\vec{x}, \vec{x}_j)$ is a function as:

- $\langle \vec{x}, \vec{x}_j \rangle$ for a polynomial or hyperbolic tangent kernel.
- $\|\vec{x} - \vec{x}_j\|$ for a Gaussian kernel.

So, the λ-FRBS is:

R_1: **If** $K(\mathrm{O}(\vec{x}, \vec{x}_1))$ is $Sigm(\lambda \cdot x \cdot \alpha_1 \cdot y_1) * K(\mathrm{O}(\vec{x}, \vec{x}_2))$ is $Sigm(\lambda \cdot x \cdot \alpha_2 \cdot y_2) * \cdots *$
$K(\mathrm{O}(\vec{x}, \vec{x}_m))$ is $Sigm(\lambda \cdot x \cdot \alpha_m \cdot y_m) * b$ is $Sigm(\lambda \cdot x)$
then $Y_1 = 1$,

R_2: **If** $K(\mathrm{O}(\vec{x}, \vec{x}_1))$ is $Sigm^*(\lambda \cdot x \cdot \alpha_1 \cdot y_1) * K(\mathrm{O}(\vec{x}, \vec{x}_2))$ is $Sigm^*(\lambda \cdot x \cdot \alpha_2 \cdot y_2) * \cdots *$
$K(\mathrm{O}(\vec{x}, \vec{x}_m))$ is $Sigm^*(\lambda \cdot x \cdot \alpha_m \cdot y_m) * b$ is $Sigm^*(\lambda \cdot x)$
then $Y_2 = -1$

and finally we arrive to:

R_1: **If** $\mathrm{O}(\vec{x}, \vec{x}_1)$ is $Sigm(\lambda \cdot K(x) \cdot \alpha_1 \cdot y_1) * \mathrm{O}(\vec{x}, \vec{x}_2)$ is $Sigm(\lambda \cdot K(x) \cdot \alpha_2 \cdot y_2) * \cdots *$
$\mathrm{O}(\vec{x}, \vec{x}_m)$ is $Sigm(\lambda \cdot K(x) \cdot \alpha_m \cdot y_m) * b$ is $Sigm(\lambda \cdot x)$
then $Y_1 = 1$,

R_2: **If** $\mathrm{O}(\vec{x}, \vec{x}_1)$ is $Sigm^*(\lambda \cdot K(x) \cdot \alpha_1 \cdot y_1) * \mathrm{O}(\vec{x}, \vec{x}_2)$ is $Sigm^*(\lambda \cdot K(x) \cdot \alpha_2 \cdot y_2) * \cdots *$
$\mathrm{O}(\vec{x}, \vec{x}_m)$ is $Sigm^*(\lambda \cdot K(x) \cdot \alpha_m \cdot y_m) * b$ is $Sigm^*(\lambda \cdot x)$
then $Y_2 = -1$.

The function $\mathrm{O}(\vec{x},\vec{x}_j)$ can be directly considered a ***similarity measure*** between the vectors $\vec{x}$ and $\vec{x}_j$ when the Gaussian kernel is used. For other kernels, this reasoning is correct if the vectors are normalized to have unit length. Such normalization can collapse two vectors having the same direction but different magnitude. To avoid this, we can augment the vectors with a feature of magnitude 1.0, making it $(d+1)$-dimensional, and then normalize them [15].

This rule set of a λ-FRBS, which represents a SVM, offers the knowledge that we were looking for. In the following section, the linguistic interpretation of the expression $Sigm(\lambda \cdot \alpha_i \cdot y_i \cdot K(x))$ is explained for each type of kernel.

5.2. *Linguistic interpretation of* $Sigm(\lambda \cdot \alpha_i \cdot y_i \cdot K(x))$

Before studying the interpretation of the expression for each type of kernel, several comments are necessary:

(1) Once the linguistic interpretation of $Sigm(\lambda \cdot \alpha_i \cdot y_i \cdot K(x))$ is achieved, we have the interpretation for $Sigm^*(\lambda \cdot \alpha_i \cdot y_i \cdot K(x))$ because

$$Sigm^*(\lambda \cdot \alpha_i \cdot y_i \cdot K(x)) = 1 - Sigm(\lambda \cdot \alpha_i \cdot y_i \cdot K(x)) \equiv \text{“Not } [Sigm(\lambda \cdot \alpha_i \cdot y_i \cdot K(x))]\text{”}.$$

(2) The linguistic interpretation of the expression $Sigm(b)$ is “b is approximately larger than 0”.
(3) As λ and α_i are greater than zero, we can study the linguistic interpretation of the expression $Sigm(\omega_i \cdot y_i \cdot K(x))$ where $\omega_i = \lambda \cdot \alpha_i \in \mathbb{R}^+$.

In the following items we study the interpretation of $Sigm(\omega_i \cdot y_i \cdot K_0(x))$ for different types of kernel functions classified into four groups:

(a) ***Hyperbolic tangent kernel*** $(K_1(x) = \tanh(a \cdot x + c),\ a \cdot c \leqslant 0,\ c \leqslant 0)$.

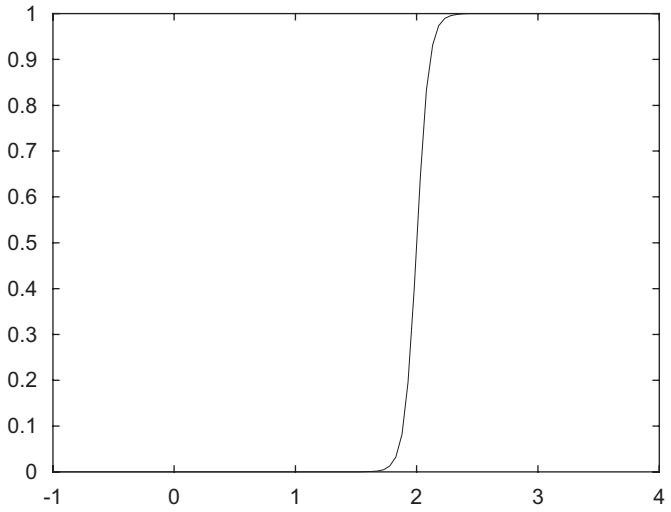

Fig. 2. $Sigm(\omega_i \cdot y_i \cdot K_1(x))$ with $\omega_i = 20$, $a = 1$, $c = (-2)$ and $y_i = (+1)$.

In this case, the expression "x is $Sigm(\omega_i \cdot y_i \cdot K_1(x))$" can be interpreted as:

- "x is approximately larger than $(-c/a)$", when $y_i = (+1)$
  (Fig. 2 with $\omega_i = 20$, $a = 1$ and $c = (-2)$).
- "x is approximately smaller (not larger) than $(-c/a)$", when $y_i = (-1)$, because

$$Sigm(-x) = 1 - Sigm(x) \equiv \text{Not } [Sigm(x)].$$

The magnitude of the slope of this fuzzy set depends on the parameters ω_i and a.

(b) ***Polynomial kernel with odd exponent*** ($K_2(x) = (x + s)^d$ with d odd).
In this case, the expression "x is $Sigm(\omega_i \cdot y_i \cdot K_2(x))$" can be interpreted as follows:

- "x is approximately larger than about $(-s)$", when $y_i = (+1)$
  (Fig. 3 with $s = (-1)$, $\omega_i = 20$ and $d = 5$).
- "x is approximately smaller than about $(-s)$", when $y_i = (-1)$.

The magnitude of the slope and the width of the term "about $(-s)$" depend on the parameter ω_i and the exponent d.

(c) ***Polynomial kernel with even exponent*** ($K_3(x) = (x + s)^d$ with d even).
In this case, the expression "x is $Sigm(\omega_i \cdot y_i \cdot K_3(x))$" returns:

- Membership Degrees in [0.5,1) for all x, when $y_i = (+1)$ (Fig. 4a with $s = (-1)$, $\omega_i = 20$ and $d = 4$).
- Membership Degrees in (0,0.5] for all x, when $y_i = (-1)$ (Fig. 4b with $s = (-1)$, $\omega_i = 20$ and $d = 4$).

On the other hand, the operator $*$ (symmetric sum) behaves as:

- t-Conorm with neuter element equal to 0.5 instead of 0, in the interval [0.5,1).
- t-Norm with neuter element equal to 0.5 instead of 1, in the interval (0,0.5].

Thus, supposing, without loss of generality:

- $y_i = (+1)$ $(i = 1, \ldots, p)$.
- $y_i = (-1)(i = (p + 1), \ldots, m)$.
- $b < 0$.

Then the following expression:

$$\underset{i=1}{\overset{m}{*}} (Sigm\ (\omega_i \cdot y_i \cdot K_3(x))) * Sigm\ (b)$$

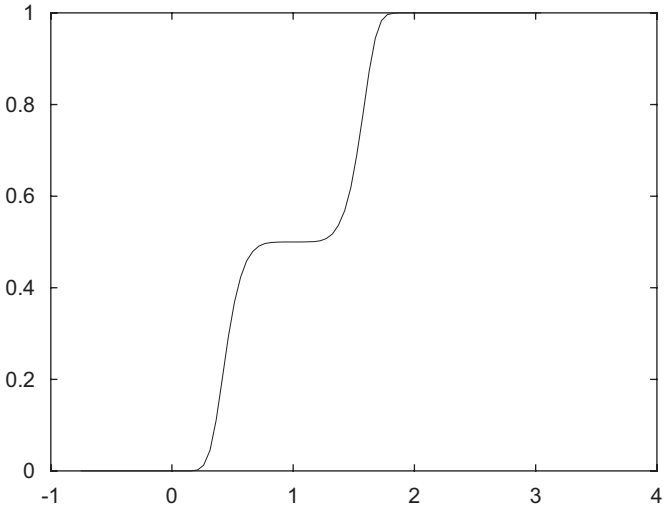

Fig. 3. $Sigm(\omega_i \cdot y_i \cdot K_2(x))$ with $s = (-1)$, $\omega_i = 20$, $d = 5$ and $y_i = (+1)$.

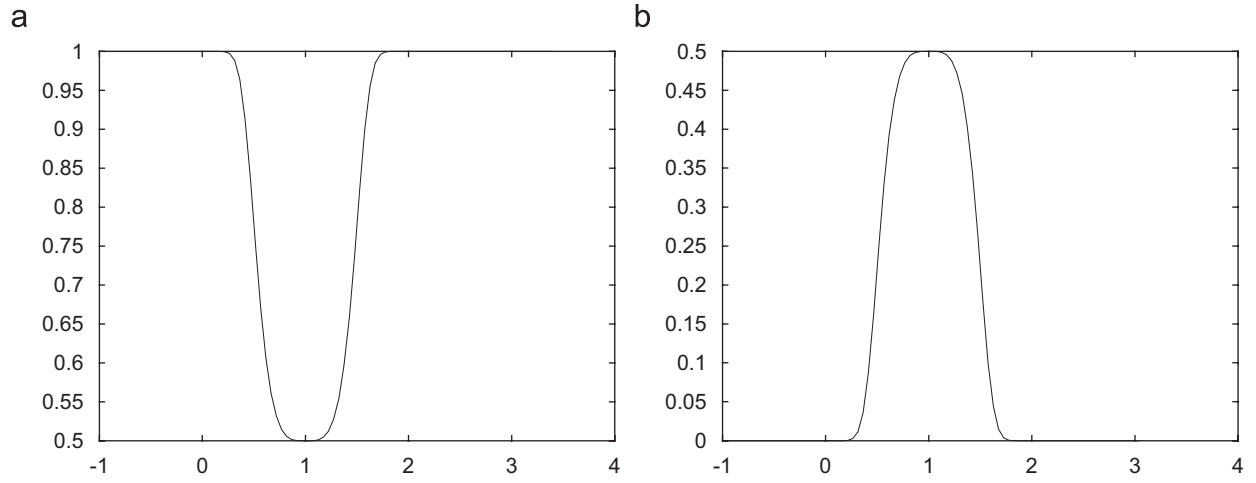

Fig. 4. (a) $Sigm(\omega_i \cdot y_i \cdot K_3(x))$ with $s = (-1)$, $\omega_i = 20$, $d = 4$ and $y_i = (+1)$. (b) $Sigm(\omega_i \cdot y_i \cdot K_3(x))$ with $s = (-1)$, $\omega_i = 20$, $d = 4$ and $y_i = (-1)$.

can be modified as

$$\left[\underset{i=1}{\overset{p}{OR}}\left(Sigm\ (\omega_i \cdot y_i \cdot K_3(x))\right)\right]\ \ comp\ \ \left[\underset{i=p+1}{\overset{m}{AND}}\left(Sigm\ (\omega_i \cdot y_i \cdot K_3(x)))\ AND\ Sigm\ (b)\right)\right].$$

It can be interpreted as

$$\left[\underset{i=1}{\overset{p}{OR}}\left(\text{“}x \text{ is not approximately}_{OR} \text{ about } (-s)\text{''}\right)\right]$$

$$comp$$

$$\left[\underset{i=p+1}{\overset{m}{AND}}\left(\text{“}x \text{ is approximately}_{AND} \text{ about } (-s)\text{''}\right)\ AND\ \text{“}b \text{ is approximately larger than } 0\text{''}\right].$$

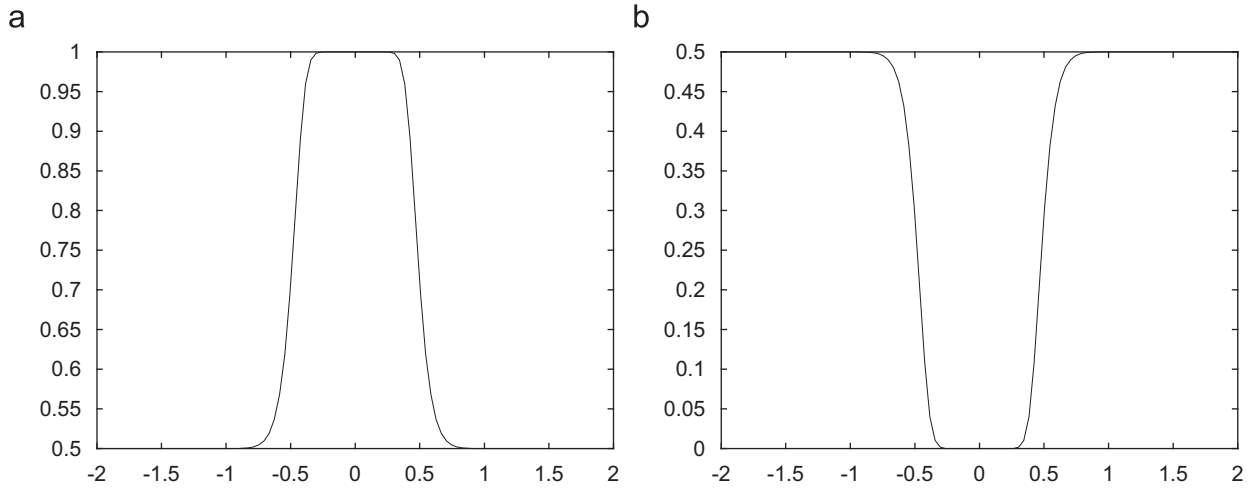

Fig. 5. (a) $Sigm(\omega_i \cdot y_i \cdot K_4(x))$ with $\omega_i = 20$, $\sigma = 0.2$ and $y_i = (+1)$. (b) $Sigm(\omega_i \cdot y_i \cdot K_4(x))$ with $\omega_i = 20$, $\sigma = 0.2$ and $y_i = (-1)$.

The magnitude of the slope of the fuzzy sets "*is* [*not*] $approximately_{[OR|AND]}$ *about* $(-s)$" and the width of the term "about $(-s)$" depends on the parameter ω_i and the exponent d.

The words ***approximately***$_{OR}$ and ***approximately***$_{AND}$ are used in this expression because they only have a valid meaning in the aggregation of fuzzy propositions with the t-conorm or t-norm provided by the symmetric sum, respectively.

(d) ***Gaussian kernel*** $(K_4(x) = \mathrm{e}^{-(x^2/2\sigma^2)})$.

In this case, the expression "x is $Sigm(\omega_i \cdot y_i \cdot K_4(x))$" also returns:

- Membership Degrees in [0.5,1) for all x, when $y_i = (+1)$ (Fig. 5a with $\omega_i = 20$ and $\sigma = 0.2$).
- Membership Degrees in (0,0.5] for all x, when $y_i = (-1)$ (Fig. 5b with $\omega_i = 20$ and $\sigma = 0.2$).

If the same assumptions made for $K_3(x)$ are supposed now, the following expression:

$$\underset{i=1}{\overset{m}{*}} (Sigm\ (\omega_i \cdot y_i \cdot K_4(x))) * Sigm\ (b)$$

can be modified as

$$\left[\underset{i=1}{\overset{p}{OR}} (Sigm\ (\omega_i \cdot y_i \cdot K_4(x)))\right]\ \ comp\ \ \left[\underset{i=p+1}{\overset{m}{AND}} (Sigm\ (\omega_i \cdot y_i \cdot K_4(x)))\ AND\ Sigm\ (b)\right].$$

It can be interpreted as

$$\left[\underset{i=1}{\overset{p}{OR}} \left(\text{“}x \text{ is approximately}_{OR}\ 0\text{''}\right)\right]$$

$$comp$$

$$\left[\underset{i=p+1}{\overset{m}{AND}} \left(\text{“}x \text{ is not approximately}_{AND}\ 0\text{''}\right)\ AND\ \text{“}b \text{ is approximately larger than } 0\text{''}\right].$$

The magnitude of the slope of the fuzzy sets "*is* [*not*] $approximately_{[OR|AND]}0$" is determined by the parameter ω_i. The width of these fuzzy sets is determined by the parameter σ in $K_4(x)$.

6. Examples

The main aim of the present section is to illustrate the advantages and disadvantages of the λ-FRBS with respect to the "course of dimensionality". As the reader has seen the λ-FRBS' number of fuzzy rules is fixed to only two. The

number of propositions of each rule varies depending on the set of SVs selected by the SVM's learning algorithm. For that reason, the number of atoms in each rule increases with the complexity of the problem.

Clarifying the last idea, we can suppose a very high-dimensional ($\gg 50\,000$) classification problem. If classes are linearly separable, just a few SVs will be provided by the learning algorithm and in consequence just a few antecedents will be obtained in the two rules of the λ-FRBS.

Three classification problems have been used for illustrating the presented extraction method:

1. The comprehension of the rules extracted from a SVM is shown with the X-or problem.
2. The fuzzy rules extracted from several SVMs that solve a multicategory classification problem are illustrated with Iris problem.
3. USPS problem is used for indicating that high-dimensional data sets can be modeled with λ-FRBSs.

6.1. X-Or problem

The *X-Or* problem has been selected to show the construction of a λ-FRBS that extracts knowledge from a SVM which solves the *X-Or*. This problem is two-class classification problem and it is composed by the following training examples:

$$\{[\vec{x}_1 = (-1,-1), y_1 = -1], [\vec{x}_2 = (-1,1), y_2 = 1], [\vec{x}_3 = (1,-1), y_3 = 1], [\vec{x}_4 = (1,1), y_4 = -1]\}.$$

This classification problem has been solved with SVMs by choosing four different kernels. Fuzzy rules have been extracted from these trained SVMs. The value λ has been established to 20 in all the experiments.

6.1.1. λ-FRBS from SVM with hyperbolic tangent kernel

In this case, the used kernel is $K_1(x) = \tanh(x-1)$, $b = 0$ and the values $\alpha_i \cdot y_i$ are

$$\{\alpha_1 \cdot y_1 = -0.775,\ \alpha_2 \cdot y_2 = 0.775,\ \alpha_3 \cdot y_3 = 0.775,\ \alpha_4 \cdot y_4 = -0.775\}.$$

Thus, the λ-FRBS extracted from a SVM that solves the X-Or problem is:

R_1: **If** $\langle \vec{x}, \vec{x}_1 \rangle$ is $Sigm(-15.5 \cdot K_1(x)) * \langle \vec{x}, \vec{x}_2 \rangle$ is $Sigm(15.5 \cdot K_1(x)) *$
$\langle \vec{x}, \vec{x}_3 \rangle$ is $Sigm(15.5 \cdot K_1(x)) * \langle \vec{x}, \vec{x}_4 \rangle$ is $Sigm(-15.5 \cdot K_1(x))$
then $Y_1 = 1$,

R_2: **If** $\langle \vec{x}, \vec{x}_1 \rangle$ is $Sigm^*(-15.5 \cdot K_1(x)) * \langle \vec{x}, \vec{x}_2 \rangle$ is $Sigm^*(15.5 \cdot K_1(x)) *$
$\langle \vec{x}, \vec{x}_3 \rangle$ is $Sigm^*(15.5 \cdot K_1(x)) * \langle \vec{x}, \vec{x}_4 \rangle$ is $Sigm^*(-15.5 \cdot K_1(x))$
then $Y_2 = -1$.

It can be interpreted as:

R_1: **If** $\langle \vec{x}, \vec{x}_1 \rangle$ is approximately smaller than 1 $*$
$\langle \vec{x}, \vec{x}_2 \rangle$ is approximately larger than 1 $*$
$\langle \vec{x}, \vec{x}_3 \rangle$ is approximately larger than 1 $*$
$\langle \vec{x}, \vec{x}_4 \rangle$ is approximately smaller than 1
then $Y_1 = 1$,

R_2: **If** $\langle \vec{x}, \vec{x}_1 \rangle$ is approximately larger than 1 $*$
$\langle \vec{x}, \vec{x}_2 \rangle$ is approximately smaller than 1 $*$
$\langle \vec{x}, \vec{x}_3 \rangle$ is approximately smaller than 1 $*$
$\langle \vec{x}, \vec{x}_4 \rangle$ is approximately larger than 1
then $Y_2 = -1$.

To understand this λ-FRBS, we must analyze the information provided by the fuzzy propositions. They offer us a comparison of the similarity that exists between $\vec{x}$ and each SV $\vec{x}_i$. For example, if the following propositions are true

for the example $\vec{x}$ (with $i_0 \neq j_0$):

- "$\langle \vec{x}, \vec{x}_{i_0} \rangle$ *is approximately larger than* 1".
- "$\langle \vec{x}, \vec{x}_{j_0} \rangle$ *is approximately smaller than* 1".

We can deduce that the similarity between $\vec{x}$ and $\vec{x}_{i_0}$ (*similarity* $(\vec{x}, \vec{x}_{i_0})$) is greater than *similarity* $(\vec{x}, \vec{x}_{j_0})$. Because:

- $\langle \vec{x}, \vec{x}_{i_0} \rangle = \|\vec{x}\| \cdot \|\vec{x}_{i_0}\| \cdot \cos \alpha_{i_0} \geqslant 1 \Rightarrow \cos \alpha_{i_0} \geqslant 1/\|\vec{x}\| \cdot \|\vec{x}_{i_0}\|$.
- $\langle \vec{x}, \vec{x}_{j_0} \rangle = \|\vec{x}\| \cdot \|\vec{x}_{j_0}\| \cdot \cos \alpha_{j_0} \leqslant 1 \Rightarrow \cos \alpha_{j_0} \leqslant 1/\|\vec{x}\| \cdot \|\vec{x}_{j_0}\|$.
- As $\|\vec{x}_1\| = \|\vec{x}_2\| = \|\vec{x}_3\| = \|\vec{x}_4\|$, we have $\cos \alpha_{i_0} \geqslant \cos \alpha_{j_0}$.
- As $\cos \alpha_i$ can be considered a similarity measure between $\vec{x}$ and $\vec{x}_i$, we obtain

$$\textit{similarity } (\vec{x}, \vec{x}_{i_0}) \geqslant \textit{similarity}(\vec{x}, \vec{x}_{j_0}).$$

Once the information provided by the fuzzy propositions is understood, the action of the λ-FRBS is comprehensible:

- Rule R_1: If the degree of "*similarity* $(\vec{x}, \vec{x}_2)$ or *similarity* $(\vec{x}, \vec{x}_3)$" is higher than the one of "*similarity* $(\vec{x}, \vec{x}_1)$ or *similarity* $(\vec{x}, \vec{x}_4)$" then the output is equal to 1.
- Rule R_2: Otherwise, the output is equal to (-1).

6.1.2. λ-FRBS from SVM with polynomial kernel with odd exponent

In this case, the used kernel is $K_2(x) = (x+1)^3$, $b = 0$ and the values $\alpha_i \cdot y_i$ are

$$\{\alpha_1 \cdot y_1 = -0.0417,\ \alpha_2 \cdot y_2 = 0.0417,\ \alpha_3 \cdot y_3 = 0.0417,\ \alpha_4 \cdot y_4 = -0.0417\}.$$

Thus, the λ-FRBS extracted from a SVM that solves the X-Or problem is:

R_1: **If** $\langle \vec{x}, \vec{x}_1 \rangle$ is $Sigm(-0.834 \cdot K_2(x)) * \langle \vec{x}, \vec{x}_2 \rangle$ is $Sigm(0.834 \cdot K_2(x)) *$
$\langle \vec{x}, \vec{x}_3 \rangle$ is $Sigm(0.834 \cdot K_2(x)) * \langle \vec{x}, \vec{x}_4 \rangle$ is $Sigm(-0.834 \cdot K_2(x))$
then $Y_1 = 1$,

R_2: **If** $\langle \vec{x}, \vec{x}_1 \rangle$ is $Sigm^*(-0.834 \cdot K_2(x)) * \langle \vec{x}, \vec{x}_2 \rangle$ is $Sigm^*(0.834 \cdot K_2(x)) *$
$\langle \vec{x}, \vec{x}_3 \rangle$ is $Sigm^*(0.834 \cdot K_2(x)) * \langle \vec{x}, \vec{x}_4 \rangle$ is $Sigm^*(-0.834 \cdot K_2(x))$
then $Y_2 = -1$.

It can be interpreted as:

R_1: **If** $\langle \vec{x}, \vec{x}_1 \rangle$ is approximately smaller than about $(-1) *$
$\langle \vec{x}, \vec{x}_2 \rangle$ is approximately larger than about $(-1) *$
$\langle \vec{x}, \vec{x}_3 \rangle$ is approximately larger than about $(-1) *$
$\langle \vec{x}, \vec{x}_4 \rangle$ is approximately smaller than about (-1)
then $Y_1 = 1$,

R_2: **If** $\langle \vec{x}, \vec{x}_1 \rangle$ is approximately larger than about $(-1) *$
$\langle \vec{x}, \vec{x}_2 \rangle$ is approximately smaller than about $(-1) *$
$\langle \vec{x}, \vec{x}_3 \rangle$ is approximately smaller than about $(-1) *$
$\langle \vec{x}, \vec{x}_4 \rangle$ is approximately larger than about (-1)
then $Y_2 = -1$.

These fuzzy propositions and the ones of the λ-FRBS obtained with the kernel $K_1(x)$ provide the same information. They offer a comparison of the similarity that exists between $\vec{x}$ and each SV $\vec{x}_i$. For example, if the following propositions are true for the example $\vec{x}$ (with $i_0 \neq j_0$):

- "$\langle \vec{x}, \vec{x}_{i_0} \rangle$ *is smaller than about* (-1)" .
- "$\langle \vec{x}, \vec{x}_{j_0} \rangle$ *larger than about* (-1)".

We can deduce that the similarity between $\vec{x}$ and $\vec{x}_{i_0}$ (*similarity* $(\vec{x}, \vec{x}_{i_0})$) is lower than *similarity* $(\vec{x}, \vec{x}_{j_0})$, because:

- $\langle \vec{x}, \vec{x}_{i_0} \rangle = \|\vec{x}\| \cdot \|\vec{x}_{i_0}\| \cdot \cos \alpha_{i_0} \leqslant (-1) \Rightarrow \cos \alpha_{i_0} \leqslant (-1)/\|\vec{x}\| \cdot \|\vec{x}_{i_0}\|$.
- $\langle \vec{x}, \vec{x}_{j_0} \rangle = \|\vec{x}\| \cdot \|\vec{x}_{j_0}\| \cdot \cos \alpha_{j_0} \geqslant (-1) \Rightarrow \cos \alpha_{j_0} \geqslant (-1)/\|\vec{x}\| \cdot \|\vec{x}_{j_0}\|$.
- As $\|\vec{x}_1\| = \|\vec{x}_2\| = \|\vec{x}_3\| = \|\vec{x}_4\|$, we have $\cos \alpha_{i_0} \leqslant \cos \alpha_{j_0}$.
- As $\cos \alpha_i$ can be considered a similarity measure between $\vec{x}$ and $\vec{x}_i$, we obtain

$$\textit{similarity } (\vec{x}, \vec{x}_{i_0}) \leqslant \textit{similarity}(\vec{x}, \vec{x}_{j_0}).$$

Thus, the information of the fuzzy propositions and the action of this λ-FRBS is understood. It has the same explanation as the action of the λ-FRBS obtained with the kernel $K_1(x)$.

6.1.3. λ-FRBS from SVM with polynomial kernel with even exponent

In this case, the used kernel is $K_3(x) = (x)^2$, $b = 0$ and the values $\alpha_i \cdot y_i$ are

$$\{\alpha_1 \cdot y_1 = 0.0,\ \alpha_2 \cdot y_2 = 0.25,\ \alpha_3 \cdot y_3 = 0.0,\ \alpha_4 \cdot y_4 = -0.25\}.$$

Thus, the λ-FRBS extracted from a SVM that solves the X-Or problem is:

R_1: **If** $\langle \vec{x}, \vec{x}_2 \rangle$ is $Sigm(5 \cdot K_3(x)) * \langle \vec{x}, \vec{x}_4 \rangle$ is $Sigm(-5 \cdot K_3(x))$
then $Y_1 = 1$,

R_2: **If** $\langle \vec{x}, \vec{x}_2 \rangle$ is $Sigm^*(5 \cdot K_3(x)) * \langle \vec{x}, \vec{x}_4 \rangle$ is $Sigm^*(-5 \cdot K_3(x))$
then $Y_2 = -1$.

It can be interpreted as:

R_1: **If** $\langle \vec{x}, \vec{x}_2 \rangle$ is not approximately$_{\text{OR}}$ about 0
comp
$\langle \vec{x}, \vec{x}_4 \rangle$ is approximately$_{\text{AND}}$ about 0
then $Y_1 = 1$,

R_2: **If** $\langle \vec{x}, \vec{x}_4 \rangle$ is not approximately$_{\text{OR}}$ about 0
comp
$\langle \vec{x}, \vec{x}_2 \rangle$ is approximately$_{\text{AND}}$ about 0
then $Y_2 = -1$.

The fuzzy propositions of this λ-FRBS use localized fuzzy sets. This fact facilitates the comprehension of these propositions. For example, the following proposition:

"$\langle \vec{x}, \vec{x}_i \rangle$ *is approximately about* 0"

is true when $\vec{x}$ and $\vec{x}_i$ are approximately perpendicular.

Therefore, we can easily deduce the action of this λ-FRBS:

- Rule R_1: If ($\vec{x}$ is not approximately perpendicular to $\vec{x}_2$) and ($\vec{x}$ is approximately perpendicular to $\vec{x}_4$) then the output is equal to 1.
  The second condition is necessary to avoid the compensation in rule R_1 that would decrease the Y_1 output degree. For example, this condition is fulfilled by $\vec{x}_2 = (-1, 1)$ and $\vec{x}_3 = (1, -1)$.
- Rule R_2: If ($\vec{x}$ is not approximately perpendicular to $\vec{x}_4$) and ($\vec{x}$ is approximately perpendicular to $\vec{x}_2$) then the output is equal to (-1).
  In this case, like above, the second condition is necessary to avoid the compensation in rule R_2 that would decrease the Y_2 output degree. For example, this condition is fulfilled by $\vec{x}_1 = (-1, -1)$ and $\vec{x}_4 = (1, 1)$.

6.1.4. λ-FRBS from SVM with Gaussian kernel

Now, the used kernel is $K_4(x) = \mathrm{e}^{-(x^2/2(0.341)^2)}$, $b = 0$ and the values $\alpha_i \cdot y_i$ are

$$\{\alpha_1 \cdot y_1 = -1.0,\ \alpha_2 \cdot y_2 = 1.0,\ \alpha_3 \cdot y_3 = 1.0,\ \alpha_4 \cdot y_4 = -1.0\}.$$

Thus, the λ-FRBS extracted from a SVM that solves the X-Or problem is:

R_1: **If** $\|\vec{x} - \vec{x}_1\|$ is $Sigm(-20 \cdot K_4(x)) * \|\vec{x} - \vec{x}_2\|$ is $Sigm(20 \cdot K_4(x)) *$
$\|\vec{x} - \vec{x}_3\|$ is $Sigm(20 \cdot K_4(x)) * \|\vec{x} - \vec{x}_4\|$ is $Sigm(-20 \cdot K_4(x))$
then $Y_1 = 1$,

R_2: **If** $\|\vec{x} - \vec{x}_1\|$ is $Sigm^*(-20 \cdot K_4(x)) * \|\vec{x} - \vec{x}_2\|$ is $Sigm^*(20 \cdot K_4(x)) *$
$\|\vec{x} - \vec{x}_3\|$ is $Sigm^*(20 \cdot K_4(x)) * \|\vec{x} - \vec{x}_4\|$ is $Sigm^*(-20 \cdot K_4(x))$
then $Y_2 = -1$.

It can be interpreted as:

R_1: **If** $\|\vec{x} - \vec{x}_2\|$ is approximately$_{\mathrm{OR}}$ 0 OR $\|\vec{x} - \vec{x}_3\|$ is approximately$_{\mathrm{OR}}$ 0
comp
$\|\vec{x} - \vec{x}_1\|$ is not approximately$_{\mathrm{AND}}$ 0 AND $\|\vec{x} - \vec{x}_4\|$ is not approximately$_{\mathrm{AND}}$ 0
then $Y_1 = 1$,

R_2: **If** $\|\vec{x} - \vec{x}_1\|$ is approximately$_{\mathrm{OR}}$ 0 OR $\|\vec{x} - \vec{x}_4\|$ is approximately$_{\mathrm{OR}}$ 0
comp
$\|\vec{x} - \vec{x}_2\|$ is not approximately$_{\mathrm{AND}}$ 0 AND $\|\vec{x} - \vec{x}_3\|$ is not approximately$_{\mathrm{AND}}$ 0
then $Y_2 = -1$.

The fuzzy propositions of this λ-FRBS use localized fuzzy sets and Euclidean distances as variables. This fact facilitates the comprehension of these propositions. For example, the following proposition:

$$\text{“}\|\vec{x} - \vec{x}_i\| \textit{ is approximately } 0\text{”}$$

is true when $\vec{x}$ and $\vec{x}_i$ are similar, according to Euclidean distance.

The action of this λ-FRBS can be easily understood:

- Rule R_1: The output is equal to 1 when ($\vec{x}$ is similar to $\vec{x}_2$ or $\vec{x}$ is similar to $\vec{x}_3$).
  This fact implies that ($\vec{x}$ is not similar to $\vec{x}_1$ and $\vec{x}$ is not similar to $\vec{x}_4$). Thus the compensation is avoided in this rule.
- Rule R_2: The output is equal to (-1) when ($\vec{x}$ is similar to $\vec{x}_1$ or $\vec{x}$ is similar to $\vec{x}_4$).
  This fact implies that ($\vec{x}$ is not similar to $\vec{x}_2 = (-1, 1)$ and $\vec{x}$ is not similar to $\vec{x}_3$). Thus the compensation is avoided in this rule.

6.2. Iris classification problem

The Iris data set of Fisher [17] is a standard workbench in the machine learning community and it can be found out in the online UCI repository [5].

The data set is the description of 150 flowers as vectors in the form (x_1, x_2, x_3, x_4, y). The input features x_is are the size (in centimeters) of the following four attributes: Petal width, Petal length, Sepal width and Sepal length, respectively. The output y means the class (Setosa, Versicolor or Virginica). Each class has 50 instances. The first one is linearly separable from the other two, which are not linearly separable from each other.

We extract fuzzy rules from a trained SVM that uses a RBF kernel. The parameter selection process of our model is similar to the one used in [19]. The steps have been:

1. We have scaled the data in the interval $[-1, 1]$.
2. We have conducted directly a fivefold cross validation to obtain the best (C, σ) model parameters using all available data. This space is explored on a two-dimensional grid with the following values $C = \{2^{-5}, 2^{-3}, \ldots, 2^{15}\}$ and $\gamma = \{2^3, 2^1, \ldots, 2^{-15}\}$ where $\gamma = 1/(2 \cdot \sigma^2)$.

3. We have trained SVMs with the best obtained (C, σ) parameters by the fivefold cross validation.
4. Finally, we extract fuzzy rules by means of our method from a SVM.

Three two-class SVMs have been designed in this experiment:

- SVM_{12}: To separate classes 1 and 2.

  Class 1 (Setosa) $\Rightarrow y = (+1)$, Class 2 (Versicolor) $\Rightarrow y = (-1)$.

- SVM_{13}: To distinguish between classes 1 and 3.

  Class 1 (Setosa) $\Rightarrow y = (+1)$, Class 3 (Virginica) $\Rightarrow y = (-1)$.

- SVM_{23}: To classify between classes 2 and 3.

  Class 2 (Versicolor) $\Rightarrow y = (+1)$, Class 3 (Virginica) $\Rightarrow y = (-1)$.

To solve the multicategory problem that appears when the output of these two-class SVMs is added, we use the OvO standard method. The final class for each input is obtained using the standard technique called Vote Count or Winner-Take-All [26].

Experiments have been ran on a Pentium IV 3.20 GHz with 1 GB of main memory. It runs with Fedora Core 5 operating system. We use the `LIBSVM` [10] version 2.82 to train the SVMs.

The resulting fivefold cross validation rate was 97.3%. We obtained 17 SVs. The best parameter set was $C = 2048$ and $\sigma = 8$. So, the Gaussian kernel used in these experiments was $K(x) = \mathrm{e}^{-(x^2/128)}$.

Next, the λ-FRBS extracted from each two-class SVM is described where the value λ has been established to 20.

SVM_{12}: In this case, $b = (-0.308)$ and the values $(\alpha_i \cdot y)$ together with the SVs are:

$$(\alpha_1 \cdot y) = 96.121, \quad \vec{x}_1 = [x_{11} = (-0.55),\ x_{12} = 0.083,\ x_{13} = (-0.76),\ x_{14} = (-0.66)].$$
$$(\alpha_2 \cdot y) = 90.283, \quad \vec{x}_2 = [x_{21} = (-0.88),\ x_{22} = (-0.75),\ x_{23} = (-0.89),\ x_{24} = (-0.83)].$$
$$(\alpha_3 \cdot y) = (-186.4), \quad \vec{x}_3 = [x_{31} = (-0.55),\ x_{32} = (-0.58),\ x_{33} = (-0.32),\ x_{34} = (-0.16)].$$

Thus, the λ-FRBS extracted from SVM_{12} is:

R_1: **If** $\|\vec{x} - \vec{x}_1\|$ is $Sigm(1922.42 \cdot K(x)) * \|\vec{x} - \vec{x}_2\|$ is $Sigm(1805.66 \cdot K(x)) *$ $\|\vec{x} - \vec{x}_3\|$ is $Sigm((-3728) \cdot K(x)) * (-0.308)$ is $Sigm(x)$
then $Y_1 = 1$ (Setosa),

R_2: **If** $\|\vec{x} - \vec{x}_1\|$ is $Sigm^*(1922.42 \cdot K(x)) * \|\vec{x} - \vec{x}_2\|$ is $Sigm^*(1805.66 \cdot K(x)) *$ $\|\vec{x} - \vec{x}_3\|$ is $Sigm^*(-3728 \cdot K(x)) * (-0.308)$ is $Sigm^*(x)$
then $Y_2 = (-1)$ (Versicolor).

It can be interpreted as:

R_1: **If** $\|\vec{x} - \vec{x}_1\|$ is approximately$_{\text{OR}}$ 0 OR $\|\vec{x} - \vec{x}_2\|$ is approximately$_{\text{OR}}$ 0
comp
$\|\vec{x} - \vec{x}_3\|$ is not approximately$_{\text{AND}}$ 0 AND (-0.308) is approximately larger than 0
then $Y_1 = 1$ (Setosa),

R_2: **If** $\|\vec{x} - \vec{x}_3\|$ is approximately$_{\text{OR}}$ 0 OR (-0.308) is not approximately larger than 0
comp
$\|\vec{x} - \vec{x}_1\|$ is not approximately$_{\text{AND}}$ 0 AND $\|\vec{x} - \vec{x}_2\|$ is not approximately$_{\text{AND}}$ 0
then $Y_2 = (-1)$ (Versicolor).

SVM_{13}: In this case, $b = (-0.03)$ and the values α_i together with the SVs are:

$$(\alpha_1 \cdot y) = 41.568, \quad \vec{x}_1 = [x_{11} = (-0.55),\ x_{12} = 0.083,\ x_{13} = (-0.76),\ x_{14} = (-0.66)].$$
$$(\alpha_2 \cdot y) = 16.027, \quad \vec{x}_2 = [x_{21} = (-0.88),\ x_{22} = (-0.75),\ x_{23} = (-0.89),\ x_{24} = (-0.83)].$$

$(\alpha_3 \cdot y) = (-49.633), \quad \vec{x}_3 = [x_{31} = (-0.66),\ x_{32} = (-0.58),\ x_{33} = 0.18,\ x_{34} = 0.33].$
$(\alpha_4 \cdot y) = (-7.962), \quad \vec{x}_4 = [x_{41} = 0.11,\ x_{42} = (-0.33),\ x_{43} = 0.38,\ x_{44} = 0.16].$

Thus, the λ-FRBS extracted from SVM_{13} is:

R_1: **If** $\|\vec{x} - \vec{x}_1\|$ is $Sigm(831.36 \cdot K(x)) * \|\vec{x} - \vec{x}_2\|$ is $Sigm(320.54 \cdot K(x)) *$
$\|\vec{x} - \vec{x}_3\|$ is $Sigm((-99.66) \cdot K(x)) * \|\vec{x} - \vec{x}_4\|$ is $Sigm((-159.24) \cdot K(x)) *$
(-0.03) is $Sigm(x)$
then $Y_1 = 1$ (Setosa),

R_2: **If** $\|\vec{x} - \vec{x}_1\|$ is $Sigm^*$ $(831.36 \cdot K(x)) * \|\vec{x} - \vec{x}_2\|$ is $Sigm^*$ $(320.54 \cdot K(x)) *$
$\|\vec{x} - \vec{x}_3\|$ is $Sigm^*$ $((-99.66) \cdot K(x)) * \|\vec{x} - \vec{x}_4\|$ is $Sigm^*$ $((-159.24) \cdot K(x)) *$
(-0.03) is $Sigm^*$ (x)
then $Y_2 = (-1)$ (Virginica).

It can be interpreted as:

R_1: **If** $\|\vec{x} - \vec{x}_1\|$ is approximately$_{\text{OR}}$ 0 OR $\|\vec{x} - \vec{x}_2\|$ is approximately$_{\text{OR}}$ 0
comp
$\|\vec{x} - \vec{x}_3\|$ is not approximately$_{\text{AND}}$ 0 AND $\|\vec{x} - \vec{x}_4\|$ is not approximately$_{\text{AND}}$ 0 AND
(-0.03) is approximately larger than 0
then $Y_1 = 1$ (Setosa),

R_2: **If** $\|\vec{x} - \vec{x}_3\|$ is approximately$_{\text{OR}}$ 0 OR $\|\vec{x} - \vec{x}_4\|$ is approximately$_{\text{OR}}$ 0 OR
(-0.03) is not approximately larger than 0
comp
$\|\vec{x} - \vec{x}_1\|$ is not approximately$_{\text{AND}}$ 0 AND $\|\vec{x} - \vec{x}_2\|$ is not approximately$_{\text{AND}}$ 0
then $Y_2 = (-1)$ (Virginica).

SVM_{23}: In this case, $b = (-3.82)$ and the values α_i together with the SVs are:

$(\alpha_1 \cdot y) = 1763.8, \quad \vec{x}_1 = [x_{11} = 0.05,\ x_{12} = (-0.83),\ x_{13} = 0.18,\ x_{14} = 0.16].$
$(\alpha_2 \cdot y) = 2048, \quad \vec{x}_2 = [x_{21} = (-0.11),\ x_{22} = 0.00,\ x_{23} = 0.28,\ x_{24} = 0.41].$
$(\alpha_3 \cdot y) = 2048, \quad \vec{x}_3 = [x_{31} = 0.11,\ x_{32} = (-0.58),\ x_{33} = 0.32,\ x_{34} = 0.16].$
$(\alpha_4 \cdot y) = 2048, \quad \vec{x}_4 = [x_{41} = 0.33),\ x_{42} = (-0.16),\ x_{43} = 0.35,\ x_{44} = 0.33].$
$(\alpha_5 \cdot y) = 2048, \quad \vec{x}_5 = [x_{51} = (-0.05),\ x_{52} = (-0.41),\ x_{53} = 0.39,\ x_{54} = 0.25].$
$(\alpha_6 \cdot y) = 972.98, \quad \vec{x}_6 = [x_{61} = (-0.38),\ x_{62} = (-0.16),\ x_{63} = 0.18,\ x_{64} = 0.16].$
$(\alpha_7 \cdot y) = (-2048), \quad \vec{x}_7 = [x_{71} = (-0.05),\ x_{72} = (-0.83),\ x_{73} = 0.35,\ x_{74} = 0.16].$
$(\alpha_8 \cdot y) = (-2048), \quad \vec{x}_8 = [x_{81} = 0.05,\ x_{82} = (-0.33),\ x_{83} = 0.28,\ x_{84} = 0.41].$
$(\alpha_9 \cdot y) = (-1946.61), \quad \vec{x}_9 = [x_{91} = 0.0,\ x_{92} = (-0.16),\ x_{93} = 0.32,\ x_{94} = 0.41].$
$(\alpha_{10} \cdot y) = (-81.763), \quad \vec{x}_{10} = [x_{10,1} = 0.61,\ x_{10,2} = (-0.16),\ x_{10,3} = 0.62,\ x_{10,4} = 0.25].$
$(\alpha_{11} \cdot y) = (-2048), \quad \vec{x}_{11} = [x_{11,1} = 0.11,\ x_{11,2} = (-0.33),\ x_{11,3} = 0.39,\ x_{11,4} = 0.16].$
$(\alpha_{12} \cdot y) = (-708.41), \quad \vec{x}_{12} = [x_{12,1} = 0.0,\ x_{12,2} = (-0.5),\ x_{12,3} = 0.56,\ x_{12,4} = 0.08].$
$(\alpha_{13} \cdot y) = (-2048), \quad \vec{x}_{13} = [x_{13,1} = (-0.05),\ x_{13,2} = (-0.16),\ x_{13,3} = 0.28,\ x_{13,4} = 0.41].$

Thus, the λ-FRBS extracted from SVM_{23} is:

R_1: **If** $\|\vec{x} - \vec{x}_1\|$ is $Sigm(35276 \cdot K(x)) * \|\vec{x} - \vec{x}_2\|$ is $Sigm(40960 \cdot K(x)) *$
$\|\vec{x} - \vec{x}_3\|$ is $Sigm(40960 \cdot K(x)) * \|\vec{x} - \vec{x}_4\|$ is $Sigm(40960 \cdot K(x)) *$
$\|\vec{x} - \vec{x}_5\|$ is $Sigm(40960 \cdot K(x)) * \|\vec{x} - \vec{x}_6\|$ is $Sigm(19459.6 \cdot K(x)) *$
$\|\vec{x} - \vec{x}_7\|$ is $Sigm((-40960) \cdot K(x)) * \|\vec{x} - \vec{x}_8\|$ is $Sigm((-40960) \cdot K(x)) *$
$\|\vec{x} - \vec{x}_9\|$ is $Sigm((-38932.2) \cdot K(x)) * \|\vec{x} - \vec{x}_{10}\|$ is $Sigm((-1635.26) \cdot K(x)) *$

$\|\vec{x}-\vec{x}_{11}\|$ is $Sigm((-40960)\cdot K(x)) * \|\vec{x}-\vec{x}_{12}\|$ is $Sigm((-14168.2)\cdot K(x)) *$
$\|\vec{x}-\vec{x}_{13}\|$ is $Sigm((-40960)\cdot K(x)) *$ (-3.82) is $Sigm(x)$
then $Y_1 = 1$ (Versicolor),

R_2: **If** $\|\vec{x}-\vec{x}_1\|$ is $Sigm^*(35276\cdot K(x)) * \|\vec{x}-\vec{x}_2\|$ is $Sigm^*$ $(40960\cdot K(x)) *$
$\|\vec{x}-\vec{x}_3\|$ is $Sigm^*$ $(40960\cdot K(x)) * \|\vec{x}-\vec{x}_4\|$ is $Sigm^*$ $(40960\cdot K(x)) *$
$\|\vec{x}-\vec{x}_5\|$ is $Sigm^*$ $(40960\cdot K(x)) * \|\vec{x}-\vec{x}_6\|$ is $Sigm^*$ $(19459.6\cdot K(x)) *$
$\|\vec{x}-\vec{x}_7\|$ is $Sigm^*$ $((-40960)\cdot K(x)) * \|\vec{x}-\vec{x}_8\|$ is $Sigm^*$ $((-40960)\cdot K(x)) *$
$\|\vec{x}-\vec{x}_9\|$ is $Sigm^*$ $((-38932.2)\cdot K(x)) * \|\vec{x}-\vec{x}_{10}\|$ is $Sigm^*$ $((-1635.26)\cdot K(x)) *$
$\|\vec{x}-\vec{x}_{11}\|$ is $Sigm^*$ $((-40960)\cdot K(x)) * \|\vec{x}-\vec{x}_{12}\|$ is $Sigm^*$ $((-14168.2)\cdot K(x)) *$
$\|\vec{x}-\vec{x}_{13}\|$ is $Sigm^*$ $((-40960)\cdot K(x)) * (-3.82)$ is $Sigm^*$ (x)
then $Y_2 = (-1)$ (Virginica).

It can be interpreted as:

R_1: **If** $\|\vec{x}-\vec{x}_1\|$ is approximately$_{\mathrm{OR}}$ 0 OR $\|\vec{x}-\vec{x}_2\|$ is approximately$_{\mathrm{OR}}$ 0 OR
$\|\vec{x}-\vec{x}_3\|$ is approximately$_{\mathrm{OR}}$ 0 OR $\|\vec{x}-\vec{x}_4\|$ is approximately$_{\mathrm{OR}}$ 0 OR
$\|\vec{x}-\vec{x}_5\|$ is approximately$_{\mathrm{OR}}$ 0 OR $\|\vec{x}-\vec{x}_6\|$ is approximately$_{\mathrm{OR}}$ 0
comp
$\|\vec{x}-\vec{x}_7\|$ is not approximately$_{\mathrm{AND}}$ 0 AND $\|\vec{x}-\vec{x}_8\|$ is not approximately$_{\mathrm{AND}}$ 0 AND
$\|\vec{x}-\vec{x}_9\|$ is not approximately$_{\mathrm{AND}}$ 0 AND $\|\vec{x}-\vec{x}_{10}\|$ is not approximately$_{\mathrm{AND}}$ 0 AND
$\|\vec{x}-\vec{x}_{11}\|$ is not approximately$_{\mathrm{AND}}$ 0 AND $\|\vec{x}-\vec{x}_{12}\|$ is not approximately$_{\mathrm{AND}}$ 0 AND
$\|\vec{x}-\vec{x}_{13}\|$ is not approximately$_{\mathrm{AND}}$ 0 AND (-3.82) is approximately larger than 0
then $Y_1 = 1$ (Versicolor),

R_2: **If** $\|\vec{x}-\vec{x}_7\|$ is approximately$_{\mathrm{OR}}$ 0 OR $\|\vec{x}-\vec{x}_8\|$ is approximately$_{\mathrm{OR}}$ 0 OR
$\|\vec{x}-\vec{x}_9\|$ is approximately$_{\mathrm{OR}}$ 0 OR $\|\vec{x}-\vec{x}_{10}\|$ is approximately$_{\mathrm{OR}}$ 0 OR
$\|\vec{x}-\vec{x}_{11}\|$ is approximately$_{\mathrm{OR}}$ 0 OR $\|\vec{x}-\vec{x}_{12}\|$ is approximately$_{\mathrm{OR}}$ 0 OR
$\|\vec{x}-\vec{x}_{13}\|$ is approximately$_{\mathrm{OR}}$ 0 OR (-3.82) is not approximately larger than 0
comp
$\|\vec{x}-\vec{x}_1\|$ is not approximately$_{\mathrm{AND}}$ 0 AND $\|\vec{x}-\vec{x}_2\|$ is not approximately$_{\mathrm{AND}}$ 0 AND
$\|\vec{x}-\vec{x}_3\|$ is not approximately$_{\mathrm{AND}}$ 0 AND $\|\vec{x}-\vec{x}_4\|$ is not approximately$_{\mathrm{AND}}$ 0 AND
$\|\vec{x}-\vec{x}_5\|$ is not approximately$_{\mathrm{AND}}$ 0 AND $\|\vec{x}-\vec{x}_6\|$ is not approximately$_{\mathrm{AND}}$ 0 AND
then $Y_2 = (-1)$ (Virginica).

With these λ-FRBSs, we can understand the methodology used by the SVMs to solve a problem:

- Several training examples of each class are chosen as prototypes (SVs).
- The similarity between a new example and each prototype is evaluated. It corresponds with the membership degree of each fuzzy proposition in the rules.
- The similarities with the prototypes of a class are added to obtain a global value. It is the aggregation of propositions in the antecedent of each rule.
- Finally, the class with the highest global value is selected as output value. It corresponds with the output of the winner rule.

This philosophy is different from the one used by a standard fuzzy system. This can be summarized so:

- Several prototypes are selected (cluster centers).
- A fuzzy rule is associated with each prototype.
- The output of the fuzzy rule associated with the prototype nearest to a new example is selected as output value.

Due to this fact, the standard fuzzy systems suffer the "curse of dimensionality". They need a rule to focus on every portion of the input space. However, SVMs and λ-FRBSs only need to select some training examples and to define its importance when the similarity is calculated.

6.3. United State Postal Service (USPS) classification problem

The USPS database is a handwritten digits recognition problem [33]. It has 9298 handwritten digits and it is divided into two different folds: one of them includes the training data and another one the samples to test. The training data set includes 7291 digits and the test data set has 2007 samples. Each digit image is of size 16×16, represented by a 256-dimensional vector with entries in the interval $[-1, 1]$.

This handwritten recognition problem has been successfully solved using SVMs by choosing the different used kernels in this paper. In all cases, error rates around 4% have been obtained and the average number of SVs is around 250 for each two-class classifier. These experimental results have been obtained from [33].

From the trained SVMs above, λ-FRBSs can be found by using the proposed fuzzy-rule extraction method. These fuzzy systems have the following characteristics:

(a) The number of rules is fixed to two.
(b) The number of fuzzy propositions is determined by the number of SVs. As the average number of SVs is around 250 for each two-class classifier, the fuzzy rules have an average number of propositions equal to 250.
(c) The accuracy is the same as the one of the trained SVM. The λ-FRBSs have error rates around 4%.

The traditional fuzzy modeling tools fall through when solving the USPS classification problem as a consequence of the "curse of the dimensionality" problem [11]. Fortunately, a λ-FRBS that solves this classification problem can be obtained by means of our proposal.

7. Analysis of the extracted rules

In [2], several criteria are presented to measure the quality of the rules extracted from neural networks. These criteria include:

- *Fidelity* indicates the extent to which the rules mimic the behavior of the neural network.
- *Accuracy* measures the correctness of classification of previously unseen examples.
- *Consistency* is the extent to which the generated rule sets produce the same classification of unseen examples, even if they are extracted from different neural networks trained on the same problem.
- *Comprehensibility* is the number of rules plus the number of propositions per rule.

These measures can be used for evaluating the quality of the rules extracted from SVMs:

- Regarding to *Fidelity* and *Accuracy*, when the value λ is sufficiently high (Theorem 2), the behavior of a λ-FRBS and the one of the original SVM are the same. Therefore, the fuzzy rules of a λ-FRBS are correct according to these measures.
- Regarding to *Consistency*, the comparison is not evaluated because there is not any type of randomness in the training algorithm of SVMs or in the extraction method. So, rules extracted under different training sessions will produce the same classification of unseen examples.
- Regarding to *Comprehensibility*, the number of rules in the λ-FRBS is just two and the number of propositions per rule is equal to the cardinality of the set of SVs. When the number of SVs is high, the quantity of propositions might be excessive. It is interesting to find a method to reduce the number of SVs. Thus, the number of propositions is diminished. On the other hand, when SVM is trained on a high-dimensional data set, the comprehensibility of the λ-FRBS is rather good compared with the rest of standard fuzzy-rule extraction methods that suffer the "curse of dimensionality" problem.

A future work is to find a compromise between accuracy and interpretability of the λ-FRBSs. This can be controlled by means of the number of SVs:

- Accuracy SVM $\rightarrow$ High number of SVs $\rightarrow$ Many propositions per rule $\rightarrow$ Less interpretable λ-FRBS.
- Less accuracy SVM $\rightarrow$ Low number of SVs $\rightarrow$ Few propositions per rule $\rightarrow$ More interpretable λ-FRBS.

One approach to reduce the number of SVs is to search for Pareto-optimal solutions along the interpretability–accuracy tradeoff curve. The multiobjective design of fuzzy rule-based systems has been previously discussed in the literature [20,39].

8. Conclusions

It has been proposed the λ-FRBS which is a fuzzy system constructed from a trained SVM. As main advantages of the λ-FRBS we outline its total independency of the dimension of the problem to solve and the exact coincidence in approximating the decision function of the SVM. For that reason the λ-FRBS is strongly recommended for high-dimensional classification problems solved by low-medium complexity boundaries, i.e. few number of SVs with respect to the dimension of the space. Besides,

- The input–output mapping of the extracted .λ-FRBS is equivalent to the decision function of the SVM.
- The extraction method is not restricted by the selected kernel. The presented method is valid for every widely used kernel: hyperbolic tangent, polynomial and Gaussian.
- It is suitable to find a FRBS that model high-dimensional data sets.

On the other hand, the combination of the advantages of two important tools has been achieved:

- SVMs have demonstrated its ability to solve classification problems in an optimal way by using only necessary resources, with a solid mathematical background.
- Fuzzy Systems have shown their usefulness in function approximation [8] and its capability as mathematical models in an interpretable way.

This fact involves interesting connections between SVM and fuzzy logic in particular TSK fuzzy systems. We hope to contribute towards a better understanding on the interpretation of SVMs and the usefulness of fuzzy systems in machine learning.

Appendix A. Proof of results

Proof of Theorem 1. Let $\vec{x}_0$ be a vector belonging to the input space. It is evaluated into the *TSK* FRBS announced above:

- If $h(\vec{x}_0) \in (-\infty, 0)$ then the output fired by the FRBS is

$$Y = \frac{\sum_{i=1}^{m} Y_i \cdot g_i}{\sum_{i=1}^{m} g_i} = \frac{Y_1 \cdot I_{(0,\infty)}(h(\vec{x}_0)) + Y_2 \cdot I^*_{(0,\infty)}(h(\vec{x}_0))}{I_{(0,\infty)}(h(\vec{x}_0)) + I^*_{(0,\infty)}(h(\vec{x}_0))} = \frac{Y_1 \cdot 0 + Y_2 \cdot 1}{0+1} = Y_2 = -1,$$

  which is equal to the output provided by $f(\vec{x}_0)$, because

$$f(\vec{x}_0) = \text{sign}(h(\vec{x}_0)) = -1.$$

- If $h(\vec{x}_0) \in (0, \infty)$ then the output fired by the FRBS is

$$Y = \frac{\sum_{i=1}^{m} Y_i \cdot g_i}{\sum_{i=1}^{m} g_i} = \frac{Y_1 \cdot I_{(0,\infty)}(h(\vec{x}_0)) + Y_2 \cdot I^*_{(0,\infty)}(h(\vec{x}_0))}{I_{(0,\infty)}(h(\vec{x}_0)) + I^*_{(0,\infty)}(h(\vec{x}_0))} = \frac{Y_1 \cdot 1 + Y_2 \cdot 0}{1+0} = Y_1 = 1,$$

  which is equal to the output provided by $f(\vec{x}_0)$, because

$$f(\vec{x}_0) = \text{sign}(h(\vec{x}_0)) = 1. \qquad \square$$

Proof of Theorem 2. As,

$$\lim_{\lambda\to\infty} (Sigm(\lambda \cdot x)) = \lim_{\lambda\to\infty} \left(\frac{1}{1+\mathrm{e}^{-\lambda \cdot x}}\right) = \begin{cases} 1 & x \in (0,\infty) \\ 0 & x \in (-\infty, 0) \end{cases} \quad \text{that is equivalent to } I_{(0,\infty)}(x).$$

And

$$\lim_{\lambda\to\infty} (Sigm^*(\lambda \cdot x)) = \lim_{\lambda\to\infty} (1 - Sigm(\lambda \cdot x)) = \begin{cases} 0 & x \in (0,\infty) \\ 1 & x \in (-\infty, 0) \end{cases} \quad \text{that is equivalent to } I^*_{(0,\infty)}(x),$$

we have that the following FRBS:

R_1: **If** $h(\vec{x})$ is $Sigm(\lambda \cdot x)$ **then** $Y_1 = 1$,
R_2: **If** $h(\vec{x})$ is $Sigm^*(\lambda \cdot x)$ **then** $Y_2 = -1$

is equivalent to the FRBS considered in Theorem 1 when $\lambda \to \infty$.

Thus, as the FRBS of Theorem 1 is equivalent to the decision function $f(\vec{x})$, the FRBS presented in Theorem 2 is also equivalent to $f(\vec{x})$. □

Proof of Proposition 1.

$$
\begin{aligned}
&Sigm(\beta \cdot x) * Sigm(\beta \cdot y)\\
&= \frac{Sigm(\beta \cdot x) \cdot Sigm(\beta \cdot y)}{Sigm(\beta \cdot x) \cdot Sigm(\beta \cdot y) + (1 - Sigm(\beta \cdot x)) \cdot (1 - Sigm(\beta \cdot y))}\\
&= \frac{1}{1 + \dfrac{(1 - Sigm(\beta \cdot x)) \cdot (1 - Sigm(\beta \cdot y))}{Sigm(\beta \cdot x) \cdot Sigm(\beta \cdot y)}} = \frac{1}{1 + \dfrac{\left(1 - \left(\dfrac{1}{1 + e^{-\beta \cdot x}}\right)\right) \cdot \left(1 - \left(\dfrac{1}{1 + e^{-\beta \cdot y}}\right)\right)}{\left(\dfrac{1}{1 + e^{-\beta \cdot x}}\right) \cdot \left(\dfrac{1}{1 + e^{-\beta \cdot y}}\right)}}\\
&= \frac{1}{1 + \left(1 - \left(\dfrac{1}{1 + e^{-\beta \cdot x}}\right)\right) \cdot \left(1 - \left(\dfrac{1}{1 + e^{-\beta \cdot y}}\right)\right) \cdot (1 + e^{-\beta \cdot x}) \cdot (1 + e^{-\beta \cdot y})}\\
&= \frac{1}{1 + \left(1 - \dfrac{1}{1 + e^{-\beta \cdot x}} - \dfrac{1}{1 + e^{-\beta \cdot y}} + \dfrac{1}{1 + e^{-\beta \cdot x}} \cdot \dfrac{1}{1 + e^{-\beta \cdot y}}\right) \cdot (1 + e^{-\beta \cdot x}) \cdot (1 + e^{-\beta \cdot y})}\\
&= \frac{1}{1 + \left((1 + e^{-\beta \cdot x}) \cdot (1 + e^{-\beta \cdot y}) - (1 + e^{-\beta \cdot y}) - (1 + e^{-\beta \cdot x}) + 1\right)}\\
&= \frac{1}{1 + \left(1 + e^{-\beta \cdot x} + e^{-\beta \cdot y} + e^{-\beta \cdot x} \cdot e^{-\beta \cdot y} - 1 - e^{-\beta \cdot y} - 1 - e^{-\beta \cdot x} + 1\right)}\\
&= \frac{1}{1 + \left(e^{-\beta \cdot x} \cdot e^{-\beta \cdot y}\right)} = \frac{1}{1 + e^{-\beta \cdot x - \beta \cdot y}} = \frac{1}{1 + e^{-\beta \cdot (x + y)}} = Sigm(\beta \cdot (x + y)). \qquad \square
\end{aligned}
$$

References

[1] J.A. Alexander, M.C. Mozer, Template-based procedures for neural network interpretation, Neural Networks 12 (1999) 479–498.
[2] R. Andrews, J. Diederich, A.B. Tickle, Survey and critique of techniques for extracting rules from trained artificial neural networks, Knowledge-based Syst. 8 (6) (1995) 373–389.
[3] N. Barakat, J. Diederich, Eclectic rule-extraction from support vector machines, Internat. J. Comput. Intell. 2 (1) (2005) 59–62.
[4] J.M. Benítez, J.L. Castro, I. Requena, Are artificial neural networks black boxes?, IEEE Trans. Neural Networks 8 (5) (1997) 1156–1164.
[5] C.L. Blake, C.J. Merz, UCI repository of machine learning databases, University of California [Online] ⟨http://www.ics.uci.edu/mlearn/MLSummary.html⟩.
[6] C.J.C. Burges, A tutorial on support vector machines for pattern recognition, Knowledge Discovery and Data Mining 2 (2) (1998).
[7] J. Casillas, O. Cordon, F. Herrera, L. Magdalena (Eds.), Interpretability Issues in Fuzzy Modeling, Studies in Fuzziness and Soft Computing, vol. 28, Springer, Berlin, 2003.
[8] J.L. Castro, Fuzzy logic controllers are universal approximators, IEEE Trans. Systems Man Cybernet. 25 (4) (1995) 629–635.
[9] J.L. Castro, C.J. Mantas, J.M. Benítez, Interpretation of artificial neural networks by means of fuzzy rules, IEEE Trans. Neural Networks 13 (1) (2002) 101–116.
[10] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, 2001 [Online] ⟨http://www.csie.ntu.edu.tw/∼cjlin/libsvm⟩.
[11] Y. Chen, J. Wang, Support vector learning for fuzzy rule-based classification systems, IEEE Trans. Fuzzy Systems 11 (2003) 716–728.
[12] J. Chiang, P. Hao, Support vector learning mechanism for fuzzy rule-based modeling: a new approach, IEEE Trans. Fuzzy Systems 12 (1) (2004) 1–12.
[13] M. Detyniecki, Mathematical aggregation operators and their application to video querying, Doctoral Thesis—Research Report 2001–2002, Laboratoire d'Informatique de Paris, 2000.
[14] W. Duch, R. Setiono, J. Zurada, Computational intelligence methods for rule-based data understanding, Proc. IEEE 92 (5) (2004).
[15] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, second ed., Wiley Interscience, New York, 2000.

[16] S. Dumais, Using SVMs for text categorization, IEEE Intelligent Systems (1998) 21–23.
[17] R.A. Fisher, The use of multiple measurements in taxonomic problems, Ann. Eugenics 7 (1936) 179–188.
[18] L. Fu, Rule generation from neural networks, IEEE Trans. Systems Man Cybernet. 24 (8) (1994) 1114–1124.
[19] C. Hsu, C. Lin, A comparison of methods for multiclass support vector machines, IEEE Trans. Neural Networks 13 (2) (2002) 415–425.
[20] H. Ishibuchi, T. Murata, I.B. Turksen, Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems, Fuzzy Sets and Systems 89 (1997) 135–150.
[21] H. Jacobsson, Rule extraction from recurrent neural networks: a taxonomy and review, Neural Comput. 17 (6) (2005) 1223–1263.
[22] Y. Jin, Interpretability improvement of RBF-based neurofuzzy systems using regularized learning, in: J. Casillas, O. Cordon, F. Herrera, L. Magdalena (Eds.), Interpretability Issues in Fuzzy Modeling, Studies in Fuzziness and Soft Computing, vol. 28, Springer, Berlin, 2003.
[23] P. Klement, R. Mesiar, E. Pap, On the relationship of associative compensatory operators to triangular norms and conorms, Internat. J. Uncertainty, Fuzziness and Knowledge-based Systems 4 (2) (1996) 129–144.
[24] G.J. Klir, B. Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice-Hall, Englewood Cliffs, NJ, 1995.
[25] E. Kolman, M. Margaliot, Are neural networks white boxes?, IEEE Trans. Neural Networks 16 (4) (2005) 844–852.
[26] U.H.-G. Kressel, Pairwise classification and support vector machines, in: C.J.C. Burges, B. Scholkopf, A.J. Smola (Eds.), Advance in kernel Methods, MIT Press, Cambridge, MA, 1999, pp. 255–268, (Chapter 15).
[27] L.I. Kuncheva, Fuzzy classifier design, Studies in Fuzziness and Soft Computing, Physica-Verlag, 2000.
[28] S. Mitra, Y. Hayashi, Neuro-fuzzy rule generation: survey in soft computing framework, IEEE Trans. Neural Networks 11 (3) (2000).
[29] H. Nuñez, C. Angulo, A. Català, Rule extraction from support vector machines, in: ESANN'2002 European Symp. on Artificial Neural Networks Bruges, Belgium, 24–26 April 2002, pp. 107–112.
[30] E. Osuna, R. Freund, F. Girosi, Training support vector machines: an application to face detection, in: Conf. on Computer Vision and Pattern Recognition (CVPR '97), 1997.
[31] B. von Schmidt, F. Klawoon, Extracting fuzzy classification rules from fuzzy clusters on the basis of separating hyperplanes, in: J. Casillas, O. Cordon, F. Herrera, L. Magdalena (Eds.), Interpretability Issues in Fuzzy Modeling, Studies in Fuzziness and Soft Computing, vol. 28, Springer, Berlin, 2003.
[32] B. Schölkopf, Statistical learning and kernel methods, MSR-TR 2000-23, Microsoft Research, 2000.
[33] B. Schölkopf, A.J. Smola, Learning with Kernels. Support Vector Machines, Regularization, Optimization and Beyond, MIT Press, Cambridge, MA, 2002.
[34] W. Silvert, Symmetric summation: a class of operations on fuzzy sets", IEEE Trans. Systems Man Cybernet., SMC-9, (1979) 657–659. Reprinted in: D. Dubois, H. Prade, R.R. Yager (Eds.), Reading in Fuzzy Sets for Intelligent Systems, Morgan Kaufman, San Mateo, CA, 1993, pp. 77–79.
[35] I. Taha, J. Ghosh, Symbolic interpretation of artificial neural networks, IEEE Trans. Knowledge and Data Eng. 11 (3) (1999) 448–463.
[36] Y. Takagi, M. Sugeno, Fuzzy identification of systems and its application to modeling and control, IEEE Trans. Systems Man Cybernet. 15 (1985) 116–132.
[37] A.B. Tickle, R. Andrews, M. Golea, J. Diederich, The truth will come to light: directions and challenges in extracting the knowledge embedded within trained artificial neural networks, IEEE Trans. Neural Networks 9 (1998) 1057–1068.
[38] V. Wan, W.M. Campbell, Support vector machines for speaker verification and identification, in: Proc. Third Internat. Conf. Audio- and Video-Based Biometric Person Authentication, 2001, pp. 253–258.
[39] H. Wang, S. Kwong, Y. Jin, W. Wei, K.F. Man, Multi-objective hierarchical genetic algorithm for interpretable fuzzy rule-based knowledge extraction, Fuzzy Sets and Systems 149 (2005) 149–186.
[40] R. Yager, A. Rybalov, Uninorm aggregation operators, Fuzzy Sets and Systems 80 (1996) 111–120.
[41] R. Yager, A. Rybalov, Full reinforcement operators in aggregation techniques, IEEE Trans. Systems Man Cybernet. 28 (1998) 757–769.
[42] H.J. Zimmermann, Fuzzy Set Theory and Its Applications, second ed., Kluwer Academic Publishers, Dordrecht, Boston, MA, 1991.
[43] H.J. Zimmermann, P. Zysno, Latent connectives in human decision making, Fuzzy Sets and Systems 4 (1980) 37–51.