Future performance modeling in athletism with low quality data-based genetic fuzzy systems

ANA M. PALACIOS[1], LUCIANO SÁNCHEZ[1], INÉS COUSO[2]

[1] *Departamento de Informática, Universidad de Oviedo,*
33071 Gijón, Asturias, Spain
palaciosana@uniovi.es, luciano@uniovi.es
[2] *Departamento de Estadística e I.O. y D.M., Universidad de Oviedo,*
33071 Gijón, Asturias Spain
couso@uniovi.es

A fuzzy rule-based decision system for assisting coaches in the configuration of an athletics team is presented. The knowledge base of this system combines the experience of the trainer with genetically mined information from training sessions and competitions. The novelty of our approach comes from the fact that these sources of data have low quality: they include subjective perceptions of mistakes of the athletes, different measurements taken by different observers, and interval-valued attributes. We will use a possibilistic representation of these categories of information, in combination with an extension principle-based reasoning method, and show that the predictive power of a genetic fuzzy system which is based in these principles improves other systems that discard the vagueness of the training data.

1 INTRODUCTION

One of the most important decision of a professional atlethics coach is selecting the team that will take part in a competition. The purpose of this selection is obtaining the highest score for the whole team. However, each athlete has to compete at different events, and it may happen that the same person that has consistently good performance at certain activity does not score well at other. To avoid this, the individual capabilities must be balanced in the team.

In other words, gathering points at the largest possible number of races is preferred to obtaining the best marks in few of them.

If the marks that each athlete will obtain at each race could be known in advance, then the composition of the best team, with respect to the rival team, could be regarded as a variant of the knapsack problem [1]. However, to the best of our knowledge, there are not previous works about intelligent models of the future performance of athletes. According also to our own research, most coaches believe that an accurate prediction of the performance of an athlete at a future event is not possible, and they rely instead in a simpler, threshold-based mechanism. They establish a baseline mark, and decide whether an athlete will be able to improve that mark or not. This decision is based mainly on the personal expertise of the trainer, and supported by the values of some indicators measuring the ability of the athlete for rating high at certain activity. The selection of those baseline marks and the mentioned set of indicators for each race is not a trivial subject; it is required to keep track of each athlete in different races, and it is also needed to agree in the set of properties that best describe how an athlete fits an sport. Time ago, it was common that a trainer set an unique mark for the whole team, that depended on the results of the rival teams. By contrast, the current trend is to select a different mark for each race [17], let it be a value that serves the coach to decide whether the athlete is needed (i.e. the regional record) or the best or most relevant marks of the rival team (see, for instance, Table 1, where we have included the actual marks of a team for two different races, 100 and 400 meters hurdles.) The coach can also decide that the baseline mark is the personal mark of one athlete, and evaluate whether this athlete will be able to improve his/her mark or not.

Once that baseline mark is settled and the indicators of the races are obtained, predicting whether this mark will be reached or not is a complex decision. A coach uses his expertise, his personal knowledge of the athletes and also the values of the indicators. In this work, as we will show later, these values can be numbers, words, interval or fuzzy ranges of values, or compound measures. We will propose a method for discovering a list of linguistic rules that model the expertise of a coach, by mining a database that contain the past performance of the athletes, and the values of the aforementioned indicators. The mining task will be carried by a Genetic Fuzzy System (GFS), as described in our previous works [15]. It is remarked that we have used a non-standard model that accounts for the imprecision in the indicators, and based the modeling of that imprecision in the theory of possibility; our representation of the data is explained in Section 2. As we will show later in Section

100 meters hurdles					
Lic/Dor	Name/Club	Cat/Year	Mark	Position	Points
L-2761 8	Cepeda I. Oviedo Atl.	Seni 1977	15.13	1	8
CO-1813 3	Lopez C. Diputac.Cordoba	Juni 1988	16.28	2	7
O-4084 27	Palacios A. Universidad Oviedo	Seni 1982	16.32	3	6
O-4995 7	Menendez L. Oviedo Atl.	Juve 1989	16.39	4	5
CO-1969 4	Gallardo I. Diputac.Cordoba	Juve 1989	17.16	5	4
O-4312 19	Perez L. Esnova Gijon	Seni 1982	18.89	6	3
O-4448 28	Barragan P. Universidad Oviedo	Pro 1984	19.09	7	2
O-4423 20	Rodriguez A. Esnova Gijon	Pro 1984	21.39	8	1

400 meters hurdles					
Lic/Dor	Name/Club	Cat/Year	Mark	Position	Points
L-2761 8	Cepeda I. Oviedo Atl.	Seni 1977	1.03.88	1	8
O-4084 27	Palacios A. Universidad Oviedo	Seni 1982	1.05.49	2	7
SE-4669 4	Jimenez M. Diputac.Cordoba	Pro 1984	1.08.74	3	6
O-4448 28	Barragan P. Universidad Oviedo	Pro 1984	1.12.35	4	5
O-5331 7	Escudero Y. Oviedo Atl.	Juve 1990	1.13.42	5	4
CO-2099 3	Aranda M. Diputac.Cordoba	Juve 1989	1.15.88	6	3
O-5175 20	Vigil M. Esnova Gijon	Juve 1990	1.22.10	7	2
O-4313 19	Alvarez N. Esnova Gijon	Pro 1984	DNF	DNF	0

TABLE 1
Marks in the races of 100 and 400 meters hurdles. The coach decides whether an athlete will be in the team depending on the personal mark of the athlete, a regional record or the most relevant marks of the rival team.

3, this representation requires some changes in the inference procedure. In the same section we justify the use of weights in the consequents of the rules, that are introduced for the first time in this paper in the context of possibilistic data. In Section 4 we summarize all the changes that have to be effected to a standard GA in order to cope with this problem, and in Section 5 we explain the structure of the decision model, and review the indicators of each race. Lastly, in Section 6, we setup two Genetic Fuzzy Systems that only differ on the representation of the data and the inference mechanism, and show that the changes proposed in this paper account for a better prediction capability. We have included also a brief comparision between these new results and other previous works involving crisp algorithms.

2 POSSIBILISTIC SEMANTICS AND VAGUE INFORMATION

As we have mentioned in the introduction, we need a common framework for reasoning with numbers, words, interval or fuzzy ranges of values, and also with compound measures. In this section we show that these kinds of data are well suited for a possibilistic representation, which is commonly used in fuzzy statistics, but not so common in fuzzy logic-based models.

The possibilistic representation we use in this paper models those situations where we cannot accurately observe a property of an object, but we are given a nested family of sets, each one of them containing the true value of the property with certain probability. Observe that all the cases mentioned in the last paragraph match well with this description, and many other common types of data can be represented too with the same model. For instance, we can consider datasets with missing values (one interval that spans the whole range of the variable), left and right censored data (the value is higher or lower than a cutoff value, or it is between between a couple of bounds), compound data (each item comprises a disperse list of values), mixes of punctual and set-valued measurements (as produced by certain sensors, for instance GPS receivers) etc. All these cases share a certain degree of ignorance about the actual value of a variable, thus we will refer to them with the generic term "low quality data".

Recent works in fuzzy statistics suggest using a fuzzy representation when the data is known through a family of confidence intervals [3]. This representation assumes that a fuzzy set can be interpreted as a possibility distribution (which, in turn, is a family of probability distributions) and each $\alpha-$cut of a fuzzy feature is a random set that contains the unknown crisp value of the feature with probability $1 - \alpha$ (see [19, 20] and Figure 1). The adoption of this representation is not, however, compatible with other interpretations of a fuzzy set, that must be modified in accordance, as we will discuss in the next section.

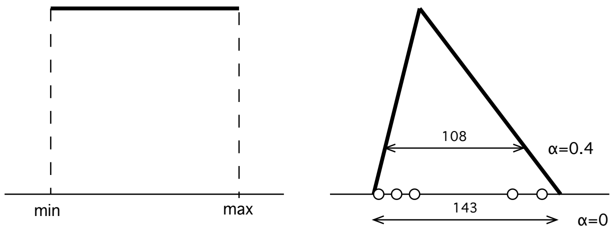


FIGURE 1
Fuzzy representation of vague data. Left: A missing value is codified with an interval that spans the whole range of the variable, or $P([\min,\max]) \leq 1$. Right: A compound value (in this example, five different measurements of the variable) can be described by a fuzzy membership, that can also be understood as an upper probability. Each α-cut contains the true value of the variable with probability at least $1-\alpha$.

3 COMPUTING THE OUTPUT OF AN FRBS WITH VAGUE DATA

The meaning attributed to a membership function in fuzzy logic differs from the semantics we have introduced in the preceding section; none of the standard methods used to compute the output of a FRBS given a fuzzy input preserve the possibilitic meaning of the data [6]. That is to say, it may happen that, given a fuzzy input that has a possibilistic meaning, we come out with a fuzzy output that is not compatible with that interpretation. In order to obtain meaningful results, in this paper we will use a reasoning method, that was proposed by us in [20] for fuzzy models, and later adapted to the classification case in [15].

Consider a fuzzy classifier comprising M rules like this:

$$\text{If } (x \text{ is } \widetilde{A}_i) \text{ then class is } C_i. \tag{1}$$

Let us use the single-winner inference mechanism for obtaining the output of this classifier. In the first place, let us suppose that we have a crisp perception x of the properties of an object. We will assign to that object the class that follows:

$$\text{class(x)} = C_{\arg\max_i \{\widetilde{A}_i(x)\}}. \tag{2}$$

Now let the object be imprecisely observed, thus all our information is "$x \in X$." If we apply the fuzzy logic based inference mechanism mentioned before, the class of the object is still a singleton:

$$\text{class'(X)} = C_{\arg\max_i \{\min\{\widetilde{A}_i(x) | x \in X\}\}} \tag{3}$$

which is not the result we need. We want to obtain the set of labels that follows:

$$\text{class(X)} = \{C_{\arg\max_i \{\widetilde{A}_i(x)\}} \mid x \in X\} \tag{4}$$

or, in other words,

$$\text{class(X)} = \{\text{class}(x) \mid x \in X\}. \tag{5}$$

which is different than eq. (3). Let us make clear this with the help of a particular case; imagine that we have a classification system defined by these rules:

$$\begin{array}{l} \text{if } x < 1 \text{ then class is } A \\ \text{if } x \in [1,2] \text{ then class is } B \\ \text{if } x > 2 \text{ then class is } C \end{array} \tag{6}$$

and the input that follows:

$$x < 1.8. \tag{7}$$

The output of the classifier —according to eq. (5)— is the set of classes {A, B}, and further refinements of this output would be arbitrary. That is to say, if the object being classified is of class C, then we know that the classifier has failed. Otherwise, we cannot precisely compute the error; alternatively, we can say that it is in the set $\{0, 1\}$.

It is remarked too that, in certain (semi-supervised) problems, there might be imprecision also in the independent variable. For instance, if an instance is labeled as "class {A, B}", we are not stating that it belongs to both categories at the same time (which is not an imprecise assert); we are expressing that we are not sure about the class of the object, as we only know that it is not in class "C". Therefore, if the output of the classifier is the set of classes {A, B} and the point is also labeled with the same set "class {A, B}", the error in this point is still $\{0, 1\}$ and not 0, as it would have been if we had used a distance-based criterion. Because of this, we will restrict the search of knowledge bases (the search algorithm will be described in Section 4) to those where each rule contains a single consequent. We will not consider knowledge bases like this:

$$\begin{array}{l}\text{if } x < 1 \text{ then class is } \{A, B\} \\ \text{if } x \in [1,2] \text{ then class is } B \\ \text{if } x > 2 \text{ then class is } C\end{array} \tag{8}$$

becase, as we have mentioned, for any dataset we can always find a KB with single consequents whose error is equal or better than that of (8).

3.1 FRBS with weights in the consequent part: definition of confidence for imprecise data

The learning algorithm that we will use in this paper produces fuzzy rules with one assert in the consequent, as mentioned, but at the same time we will permit that each consequent has a numerical weight. That is to say, according to the nomenclature in [2], we want to obtain “type 2” rules, whose structure is as follows:

$$\text{Rule } R_i\text{: If } x \text{ is } \widetilde{A}_i \text{ then Class is } C_i \text{ with } \mathrm{CF}_i, \tag{9}$$

where CF_i is the rule weight.

The weights of the rules will be obtained through extensions of the four heuristic methods defined in [8], that in the remaining of the paper will be denoted CF^{I}, $\mathrm{CF}^{\mathrm{II}}$, $\mathrm{CF}^{\mathrm{III}}$, $\mathrm{CF}^{\mathrm{IV}}$. All these heuristics depend on the confidence degree of the fuzzy rule under study (and also on the confidence degrees of those fuzzy rules with the same antecedent and different consequents) and therefore it is needed to extend the definition of the concept of “confidence” to fuzzy data before we can use type 2 rules in problems with low quality data.

Let $\{(x_1, c_1), \ldots, (x_m, c_m)\}$ be a crisp training set, and let the confidence of a fuzzy rule $c(A_i \Rightarrow C_i)$ for this crisp dataset be [8]:

$$c(A_i \Rightarrow C_i)_{(x_1,c_1,\ldots,x_m,c_m)} = \frac{\sum_{c_p=C_i} \mu_{A_i}(x_p)}{\sum_{p=1}^{m} \mu_{A_i}(x_p)}. \tag{10}$$

For a a low quality (fuzzy) dataset $\{(\widetilde{X}_1, c_1), \ldots, (\widetilde{X}_m, c_m)\}$, the direct application of the extension principle to eq. (10) is the fuzzy subset of $[0, 1]$ defined by

$$\begin{array}{l}\widetilde{c}(A_i \Rightarrow C_i)(t)_{(\widetilde{X}_1,c_1,\ldots,\widetilde{X}_m,c_m)} = \\ \max\left\{ \min\limits_{p=1\ldots m} \mu_{X_p}(x_p) \mid t = c(A_i \Rightarrow C_i)_{(x_1,c_1,\ldots,x_m,c_m)} \right\}.\end{array} \tag{11}$$

The computation of this set is computationally costly. Nevertheless it is contained in the set obtained by replacing the arithmetic operators in eq. (10) by their corresponding fuzzy arithmetic counterparts, and we will use this last approximation in our experiments. Lastly, it is remarked that eq. (11) is fuzzy valued and we need a crisp value between 0 and 1, thus we have to replace this last approximation by its defuzzified value.

4 OBTAINING FUZZY RULES FROM LOW QUALITY DATA

We will use the cooperative-competitive algorithm introduced in [7], extended to low quality data [15][16]. This extension affects two parts of the GFS: how the consequent of a rule is determined, given an antecedent and a vague dataset, and how the fitness of a rule is computed. The remaning parts (representation of the rules, generational scheme, operators, etc.) can be left unaltered provided that we define a total order between the values of the fuzzy-valued fitness function. Summarizing, in this section we will detail the three following procedures:

1. Assignment of consequents (value and weight).
2. Computation of set-valued fitness functions.
3. Genetic selection and replacement of the worst individuals.

4.1 Assignment of consequents

In [7], consequents were assigned after computing the confidences of the rules "if (x is $\widetilde{A}$) then class is C" for all the values of "C", then selecting the alternative with maximum confidence. In [15] and [13] we proposed that the confidences of a rule was the compatibility grade defined by a set of values. In the new extension of GFS the assignment of consequents (see Figure 2) depends on confidences defined by the compatibility degree of the antecedent of the rule with the examples, divided by the number of examples compatibles with the antecedent of the rule (11) (see lines 4 to 12 in Figure 2).

As we have mentioned before, the confidence of a rule is a set; since we need to select the alternative with higher confidence, we need to sort them. We have decided to build first a list of nondominated values of confidence, for choosing one value from this list and using its corresponding consequent. This is achieved through the use of the operation "dominates" in line 16. It is remarked that this operation can have different meanings, ranging from the strict dominance (A dominates B iff $a < b$ for all $a \in A$, $b \in B$) [21] to other

```
function assignImpreciseConsequent(rule)
1   for c in $\{1, \ldots, N_c\}$
2     grade = 0
3     compExample = 0
4     for example in $\{1, \ldots, N\}$
5       $\widetilde{m}$ = fuzMembership(Antecedent,example,c)
6       grade = grade $\oplus$ $\widetilde{m}$
7       if (sup $\{x : \widetilde{m}(x) > 0\} > 0$) then
8         compExample = compExample + 1
9       end if
10    end for example
11    weight[c] = grade $\oslash$ compExample
12  end for c
13  mostFrequent = $\{1, \ldots, N_c\}$
14  for c in $\{1, \ldots, N_c\}$
15    for $c_1$ in $\{c+1, \ldots, N_c\}$
16      if (weight[c] dominates weight[$c_1$]) then
17        mostFrequent = mostFrequent - { $c_1$}
18      end if
19    end for $c_1$
20  end for c
21  Consequent = select(mostFrequent)
return rule
```

FIGURE 2
If the examples are imprecise, we might not know the most frequent class label –lines 13 to 20–. In this paper we have used the dominance proposed in [10] to reduce this set to one element.

definitions that induce a total order in the set of confidences. In this paper, we have used the uniform dominance defined in [10], that induces a total order and thus the set of nondominated consequents has size 1 [15].

4.2 Computation of fitness

The fitness function depends on the winner rule for each object in the training set; we increment the fitness of the winner rule if its consequent matches with the class of the object. In this case, this function is fuzzy-valued, and we will use the procedure introduced [15] and described again in detail in Figure 4. Observe that we are using weighted rules and therefore there is an small

```
**function** assignImpreciseFitnessApprox(population,dataset)
  **for** example in $\{1, \dots, \mathrm{N}\}$
    setWinnerRule = $\emptyset$
    **for** r in $\{1, \dots, \mathrm{M}\}$
      dominated = FALSE
      r.$\widetilde{\mathrm{m}}$ = fuzMembership(Antecedent[r],example)*CF[r]
      **for** sRule in setWinnerRule
        **if** (sRule **dominates** r) **then**
          dominated = TRUE
        **end if**
      **end for** sRule
      **if** (**not** dominated **and** r.$\widetilde{\mathrm{m}} > 0$) **then**
        **for** sRule in setWinnerRule
          **if** (r.$\widetilde{\mathrm{m}}$ **dominates** sRule) **then**
            setWinnerRule = setWinnerRule $-\{$ sRule $\}$
          **end if**
        **end for** sRule
        setWinnerRule = setWinnerRule $\cup\{$ r $\}$
      **end if**
    **end for** r
    **if** (setWinnerRule == $\emptyset$) **then**
      setWinnerRule = setWinnerRule $\cup\{$ rule_freq_class $\}$
    setOfCons= $\emptyset$
    **for** sRule in setWinnerRule
      setOfCons= setOfCons $\cup\{$ consequent(sRule) $\}$
    **end for** sRule
    deltaFit= 0
    **if** ($\{$class(example)$\}$ == setOfCons **and**
        size(setOfCons)==1) **then**
      deltaFit = $\{1\}$
    **else**
      **if** ($\{$class(example)$\}\cap$ setOfCons $\neq \emptyset$) **then**
        deltaFit = $\{0, 1\}$
      **end if**
    **end if**
    Select winnerRule $\in$ setWinnerRule
    fitness[winnerRule] $=$ fitness[winnerRule] $\oplus$ deltaFit
  **end for** example
**return** fitness
```

FIGURE 3
Generalization of the function “assignFitness” to imprecise data by means of a fast approximation.

```
function assignImpreciseFitnessExhaustive(population,dataset)
  for dataset in $\{1, \ldots, 1000\}$
    fitness[dataset] = 0
    for example in $\{1, \ldots, \mathrm{N}\}$
      bestMatch = 0
      WRule = -1
      for r in $\{1, \ldots, \mathrm{M}\}$
        m = membership(Antecedent[r],example)*CF[r]
        if (m $>$ bestMatch) then
          WRule = r
          bestMatch = m
        end if
      end for r
      if (WRule == -1) then
        WRule = rule_fre_class
      end if
      if (consequent(WRule) == class(example)) then
        score = 1
      else
        if consequent(WRule) $\subset$ class(example)) then
          score= $\{0, 1\}$
        end if
      end if
      fitness[dataset] $=$ fitness[dataset] $\oplus$ score
    end for example
  end for dataset
  fitness=0
  for dataset in $\{1, \ldots, 1000\}$
    fitness=fitness $\oplus$ fitness[dataset]
  end for dataset
  fitness=mean(fitness)
return fitness
```

FIGURE 4
Other generalization of the function “assignFitness” to interval-valued or fuzzy data. This function is computationally too expensive for being used as a fitness function; it will be used instead for obtaining better estimations of test errors of the final rule bases.

change with respect to the algorithm in [15], as we need to take into account the matching between the antecedent of the rule and the object and also the weight of the rule, as shown in line 5 of Figure 3.

The algorithm shown in that figure computes an interval or fuzzy set of values of matching between each rule and the input, then discards all rules but the winner rule, and approximates the output of the FRBS by the set of the consequents of the non-discarded rules. Being based on an approximation, this output always includes the theoretical output, but it might include extra class labels. In Figure 4 we have included a different, more accurate approximation, which is based on a sample of values of the support of the input. This second approximation will be used in the next section to better determine the quality of a classifier. However, our learning will be guided by the function defined in Figure 3, because of its lower computational cost.

4.3 Genetic selection and replacement

The two other parts in the original algorithm that must be altered in order to use an imprecise fitness function are: (a) the selection of the individuals (see [7]) and (b) the removal of the worst individuals. The selection is carried by a tournament, that we have made to depend on a total order on the set of fitness values (the uniform dominance defined in [10] and also used in [15]). The same order is used to determine the worst individuals.

5 STRUCTURE OF THE PROPOSED FUZZY RULE-BASED DECISION MODEL

Once the representation of the data and the algorithm used to extract fuzzy rules from this data has been explained, we describe the model that we will use for deciding whether an athlete will take part of the team or not.

The different events where the team will collaborate are divided into speed, middle distance, long distance running, hurdling, relays, walk, jumps and throwing. Each event has, in turn, different categories. For instance, there are 100 metre hurdles and 400 metre hurdles; both are speed races. The races that form the competition are shown in Table 2. In this section we will review an outline of the selection process and the indicators used only three of these races: long jump, 100 meters and 200 meters.

5.1 Computer aided selection of an athletics team

The score of an athletics team is the sum of the individual scores of the athletes in the different events. As we have mentioned before, it is the coach's

Race	**Type**
100 meters	Speed
200 meters	Speed
400 meters	Speed
800 meters	Middle distance
1500 meters	Middle distance
3000 meters	Long distance
100 hurdles	Speed and hurdle
400 hurdles	Speed and hurdle
3000 steeplechase	Long distance
High Jump	Jumps
Pole Vault	Jumps
Long Jump	Jumps
Triple Jump	Jumps
Shot Put	Throwing
Discus	Throwing
Hammer	Throwing
Javaline	Throwing
5000 walk	Long distance and walk
4x100 meters	Relays and speed
4x400 meters	Relays and speed

TABLE 2
Races that form the competition. In this paper we have restricted ourselves to 100 and 200 meters and long jump.

responsibility to balance the capabilities of the different athletes in order to maximize the score with a team according to the regulations [18]. The scheme used to obtain the best team has the following parts:

1. Introduction of the marks of the rival teams for each race in the competition (see Figure 5 for a screen capture of the computer application).

2. Introduction of the marks of the available athletes.

3. Determination of the expected marks (or "thresholds") for each one of the races of the competition. These thresholds depend on the expert knowledge of the trainer, that sets them according to the past perfor-

mance of the athlete and the marks of the rival teams.

4. Predict whether one athlete is needed or not for each race. This prediction is fed with the thresholds obtained in the previous step and with a set of indicators for each race. These indicators are described later in this section.

5. Selection of best team. Once we have all the available information of the athletes and their expected future performance with respect to the threshold for all races, we obtain the best team and their expected marks. Combining these marks and those of the rival team we can estimate the score that the team will obtain, thus the selection of the athletes can be finally solved [1].

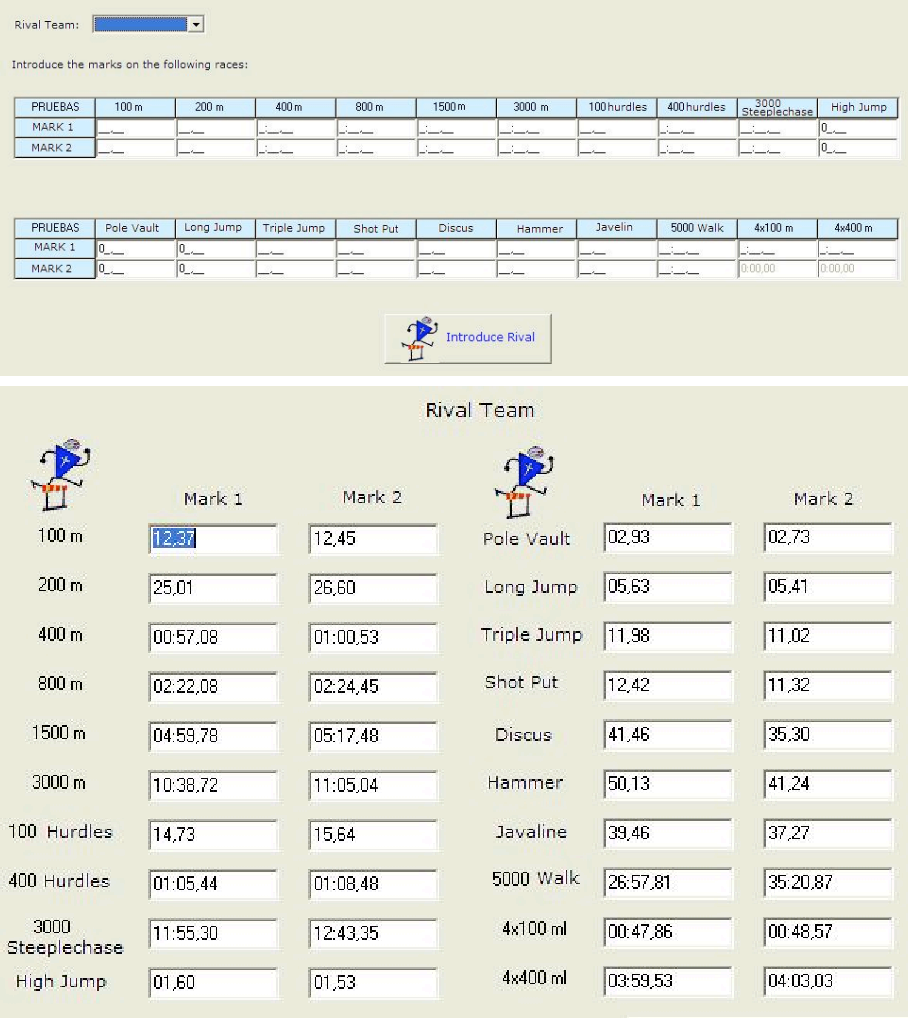
All these steps are summarized in Figure 6. It is remarked that, in this paper, we focus in in the prediction of whether an athlete will surpass his/her threshold for each race (grey box in Figure 6).

5.2 Indicators of Long Jump

There are four indicators in long jump that are used to predict whether an athlete will pass a given threshold [22]: the ratio between the weight and the height, the maximum speed in the 40 metre race and the tests of central (abdominal) muscles and lower extremities. The first two indicators are determined by the coach, who was allowed to use numbers, intervals or linguistic values (fuzzy intervals) at his convenience. The two last tests are repeated three times, and produce different numbers. The abdominal muscle test consists in counting how many flexion movements the athlete can repeat in a minute. Lastly, the lower extremities test measures how much the athlete can jump from standstill.

5.3 Indicators of 100 meters sprint

There are also four indicators in this event: the ratio between weight and height, the reaction time, the starting or 20 metre speed, and the maximum or 40 metre speed. We have collected two different databases for this problem. In the first database, three different people measure the actual reaction time, starting and maximum speed of the athletes. These three measurements are joined to form an imprecise value. On the contrary, in the second database the trainer has graded each speed and time with a mark between 0 and 10. He was allowed to express his grades with numbers, intervals or linguistic values. This second database has a highly subjective component; it serves to



Rival Team:

Introduce the marks on the following races:

PRUEBAS	100 m	200 m	400 m	800 m	1500 m	3000 m	100 hurdles	400 hurdles	3000 Steeplechase	High Jump
MARK 1	__,__	__,__	_:__,__	_:__,__	_:__,__	__:__,__	__,__	_:__,__	__:__,__	0_,__
MARK 2	__,__	__,__	_:__,__	_:__,__	_:__,__	__:__,__	__,__	_:__,__	__:__,__	0_,__

PRUEBAS	Pole Vault	Long Jump	Triple Jump	Shot Put	Discus	Hammer	Javelin	5000 Walk	4x100 m	4x400 m
MARK 1	0_,__	0_,__	__,__	__,__	__,__	__,__	__,__	__:__,__	_:__,__	_:__,__
MARK 2	0_,__	0_,__	__,__	__,__	__,__	__,__	__,__	__:__,__	0:00,00	0:00,00

Rival Team

	Mark 1	Mark 2		Mark 1	Mark 2
100 m	12,37	12,45	Pole Vault	02,93	02,73
200 m	25,01	26,60	Long Jump	05,63	05,41
400 m	00:57,08	01:00,53	Triple Jump	11,98	11,02
800 m	02:22,08	02:24,45	Shot Put	12,42	11,32
1500 m	04:59,78	05:17,48	Discus	41,46	35,30
3000 m	10:38,72	11:05,04	Hammer	50,13	41,24
100 Hurdles	14,73	15,64	Javaline	39,46	37,27
400 Hurdles	01:05,44	01:08,48	5000 Walk	26:57,81	35:20,87
3000 Steeplechase	11:55,30	12:43,35	4x100 ml	00:47,86	00:48,57
High Jump	01,60	01,53	4x400 ml	03:59,53	04:03,03

FIGURE 5
Screen capture of the computer application that assists the selection of an athletics team, showing the marks of the rival team for all races.

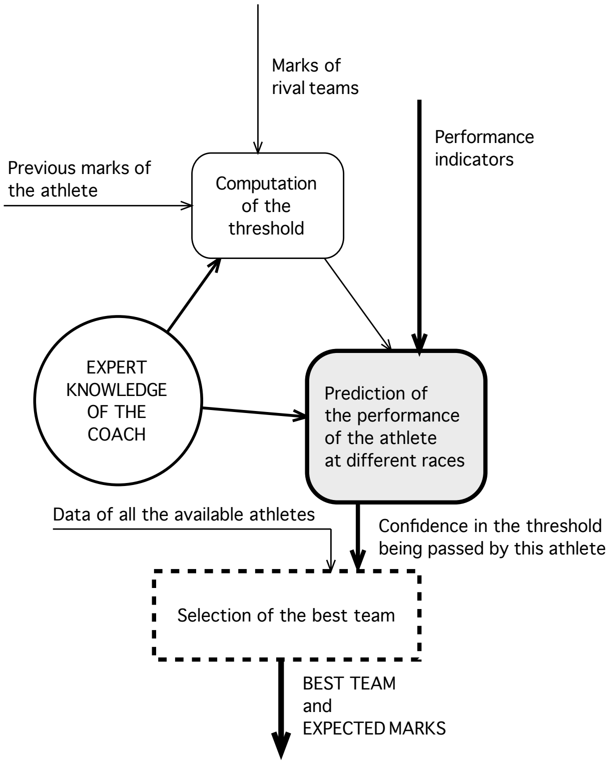


FIGURE 6
Structure of the model. In this paper we focus in the prediction of whether an athlete will surpass his/her threshold for each race (grey box)

assess the expert knowledge of the trainer about the athletes, by comparing this results with the actual measurements.

5.4 200 meters sprint

There are four indicators in this event: the ratio between weight and height, the reaction time, action in the curve or 30 meters in curve and the maximum speed in 60 meters. We have collected two different databases for this problem. In the first database, the coach measure the actual reaction time, action in the curve and maximum speed of the athletes. These measurements are combined into fuzzy values. In the second database we have included a new feature about the trainer's personal knowledge of the athletes, defined by a fuzzy term. Therefore, this second database has a subjective component so we can compare the different results for judging whether the subjective perception of the coach is a relevant variable.

6 NUMERICAL ANALYSIS OF THE ALGORITHM

In this section we have compared the results of a GFS that uses crisp datasets to the same GFS, extended for using possibilistic data, using the representation and inference function we have mentioned in the preceding sections. We have also compared the results of the extended GFS with other crisp algorithms: Linear Discriminant Analysis (LDA) [4], Multilayer Perceptrons (MLP) [5], K-Nearest Neighbours (KNN) classifier, Fuzzy rule-based Wang-Mendel (WM) [23] and Pal-Mandal (PM) [11] algorithms) and with our own results obtained in previous works [15].

All our studies have been carried with real-world data with athletes of the Oviedo University that participate in the Spanish Women's Athletic Club Championship.

6.1 Settings

Description of the datasets

We have collected eight datasets, whose descriptions are as follows:

1. Dataset Long-4: This dataset is used to predict whether an athlete will improve certain threshold in the long jump, given the indicators mentioned before. We have measured 25 athletes, thus the set has 25 instances, 4 features, 2 classes, no missing values. All the features, and also the output variable, are interval-valued and the coach introduced his personal knowledge.

2. Dataset "BLong-4": Same dataset as "Long-4", but now the measurements are defined by fuzzy-valued data, obtained by reconciling different measurements taken by three different observers.

3. Dataset "100ml-4-I": Used for predicting whether a threshold in the 100 metres sprint race is being achieved. Each measurement was repeated by three observers. 25 instances, 4 features, 2 classes, no missing data. All input and output variables are intervals.

4. Dataset “100ml-4-P”: Same dataset as “100ml-4-I”, but the measurements have been replaced by the subjective grade the trainer has assigned to each indicator (i.e.“reaction time is low” instead of “reaction time is 0.1 seg”).

5. Dataset “B200ml-I”: This dataset is used to predict whether an athlete will improve certain threshold in 200 meters. We have 19 athletes, 4 features, 2 classes, missing values. All the indicators are fuzzy-valued and the outputs are interval-valued.

6. Dataset “C200ml-I”: Same dataset as “B200ml-I”, with crisp outputs, so that the approach in this paper can be compared with other crisp algoritms.

7. Dataset “B200ml-P”: Same dataset as “B200ml-I”, with an extra feature: the subjective grade that the trainer has assigned to each athlete. We have 19 athletes, 5 features, 2 classes, missing values. All the indicator are fuzzy-valued and the outputs are interval-valued.

8. Dataset “C200ml-P”: Same dataset as “B200ml-P”, with crisp output. Similarly to “C200ml-I”, this dataset will be used in comparisons with other crisp algoritms.

Experimental design

All the experiments have been run with a population size of 100, probabilities of crossover and mutation of 0.9 and 0.1, respectively, and limited to 200 generations. The fuzzy partitions of the labels are uniform and their size is 5 except when mentioned otherwise. All the imprecise experiments were repeated 100 times with bootstrapped resamples of the training set; the “test” error is computed with the “out of the bag” instances. We have not included p-values of the statistical tests (in our case, those would be interval valued or fuzzy p-values, in turn) but graphical descriptions of the results, by means of boxplots, from which the relevance of the differences is readily obtained.

All the datasets used in this paper are available in the website of the KEEL project: `http://www.keel.es`.*.

Comparison and representation of results involving a mix crisp and low quality data-based algorithms

For comparing the performance of the generalized algorithm with that of previous statistical and rule-based approaches, each crisp algorithm has been fed

* Note to the reviewers: the web page of the KEEL project undergoes some changes. Some of the datasets will be available to the public as soon as possible.

with precise datasets that we have built by removing all the sources of uncertainty in the original, imprecise datasets, using the method proposed in [15][16]. The results will be displayed twice, with numerical (tables) and graphical (boxplots) representations:

1. Tables: We show the mean of 100 repetitions. In the "Crisp" group of columns we represent the results of learning and the quality of the original GFS [7] ("Train" column and "Test column", respectively). In the "Low Quality" columnnms, we show the results of learning (see Figure 3) and the quality of the extended GFS, (see Figure 4) is in "Approx.Train" and "Exh.Test" columns, respectively. Lastly the column, "Low Quality [15]", contains the results of the extended GFS proposed in [15], where the rules were of type 1 (no weights).

2. Boxplot: It is remarked that our boxplots are not standard, because we represent imprecise results. We use a box showing the 75% percentile of the maximum and the 25% percentile of the minimum fitness (thus the box display at least the 50% of data). In addition, we represent the interval-valued median of the maximum and minimum fitness, for this reason, we draw two marks inside the box. The dotted lines (not a part of an standard boxplot) represent the mean of the minimum and maximum fitness. Again, the information displayed in the boxplots are the results of 100 repetitions of the learning algorithm.

6.2 Compared results

Original vs. extended GFS with and without weights

We have included the numerical values of the classification error in the Table 3, and the boxplots are shown in Figure 7. The results are promising in all the experiments. We expected that the extra freedom that the coach has when he is allowed to use ranges of values and linguistic terms instead of numbers allowed us to capture better his expertise, and the results seem to confirm this intuition. For example, in Table 3, "Long-4" and "BLong-4", (the first one includes this expertise of the coach, not so the second) we can observe this difference. The same happens with "100ml-4-I" and "100ml-4-P": we obtain better results when we are using the knowlegde of the coach ("100ml-4-P"). However, in "B200ml-P" and "C200ml-P" as the knowlegde of the trainer is included as an extra feature (5 features) the results are similar to the case where we work only with the measurement obtained by three people (4 features).

The use of weights seems to improve the results, as expected; in Table 3, the new definition in the function “assign Consequent” (Figure 2) improves the results of previous works [15]. In general, the results demonstrate that there is a remarkable improvement when the heuristic weights are introduced.

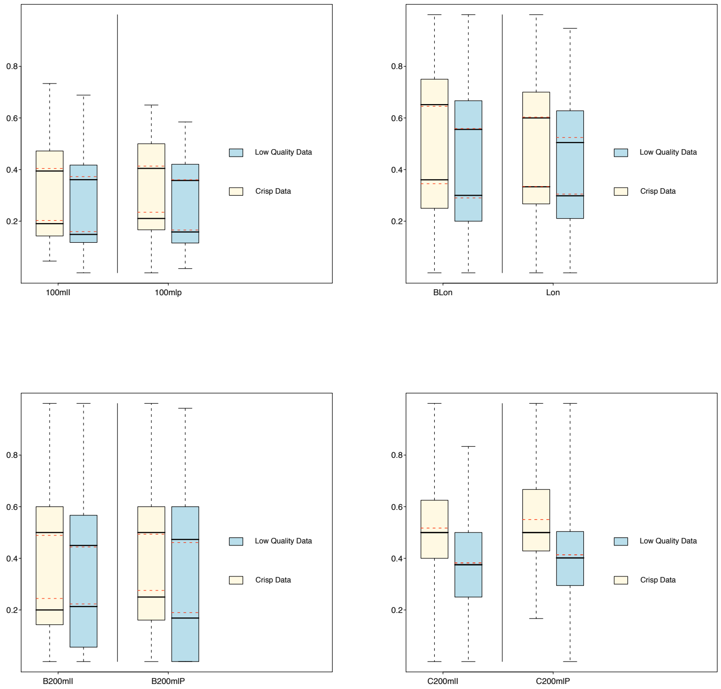


FIGURE 7
Boxplots illustrating the results of the 100 repetitions of original and extended GFS in the problems “100ml-4-I”, “100ml-4-P”, “B200ml-I”, “B200ml-P”,“BLong-4”, “Long-4”, “C200ml-I” and “C200ml-P” with 5 labels/variable

Fuzzy data vs. crisp data

In this section, we compare the extended GFS with other crisp algorithms, as mentioned before: LDA [4], MLP [5], KNN, WM [23] and PM [11]. All the experiments in these crisp algorithms have been run with the same 100

	Crisp		Low Quality		Low Quality [15]
Dataset	Train	Test	Approx. Train	Exh.Test	Exh.Test
Long-4 CF_i^0	0.143	**[0.334,0.603]**	[0.003,0.286]	**[0.304,0.524]**	[0.349,0.616]
Long-4 CF_i^I			[0.010,0.294]	[0.295,0.519]	
Long-4 CF_i^{II}			[0.007,0.291]	[0.297,0.522]	
Long-4 CF_i^{III}			[0.005,0.288]	[0.299,0.520]	
Long-4 CF_i^{IV}			[0.007,0.291]	[0.299,0.522]	
BLong-4 CF_i^0	0.135	**[0.345,0.645]**	[0.023,0.294]	**[0.299,0.586]**	-
BLong-4 CF_i^I			[0.300,0.569]	[0.290,0.558]	
BLong-4 CF_i^{II}			[0.018,0.288]	[0.304,0.575]	
BLong-4 CF_i^{III}			[0.011,0.281]	[0.297,0.569]	
BLong-4 CF_i^{IV}			[0.022,0.292]	[0.293,0.563]	
100ml-4-I CF_i^0	0.111	**[0.202,0.404]**	[0.074,0.273]	**[0.159,0.372]**	[0.189,0.476]
100ml-4-I CF_i^I			[0.095,0.283]	[0.184,0.400]	
100ml-4-I CF_i^{II}			[0.069,0.258]	[0.171,0.385]	
100ml-4-I CF_i^{III}			[0.083,0.271]	[0.159,0.372]	
100ml-4-I CF_i^{IV}			[0.069,0.258]	[0.169,0.383]	
100ml-4-P CF_i^0	0.127	**[0.234,0.413]**	[0.067,0.279]	**[0.165,0.360]**	[0.17,0.406]
100ml-4-P CF_i^I			[0.085,0.283]	[0.182,0.385]	
100ml-4-P CF_i^{II}			[0.060,0.258]	[0.195,0.394]	
100ml-4-P CF_i^{III}			[0.073,0.271]	[0.165,0.361]	
100ml-4-P CF_i^{IV}			[0.061,0.259]	[0.194,0.393]	
B200ml-I CF_i^0	0.129	**[0.244,0.488]**	[0.001,0.252]	**[0.244,0.449]**	-
B200ml-I CF_i^I			[0.246,0.497]	[0.243,0.479]	
B200ml-I CF_i^{II}			[0.003,0.254]	[0.223,0.444]	
B200ml-I CF_i^{III}			[0.002,0.253]	[0.238,0.457]	
B200ml-I CF_i^{IV}			[0.003,0.254]	[0.227,0.445]	
B200ml-P CF_i^0	0.141	**[0.275,0.493]**	[0.001,0.272]	**[0.213,0.474]**	-
B200ml-P CF_i^I			[0.236,0.507]	[0.207,0.494]	
B200ml-P CF_i^{II}			[0.001,0.272]	[0.189,0.460]	
B200ml-P CF_i^{III}			[5.263,0.271]	[0.199,0.470]	
B200ml-P CF_i^{IV}			[0.002,0.273]	[0.198,0.469]	
C200ml-I CF_i^0	0.005	**0.517**	[0.026,0.028]	**0.392**	-
C200ml-I CF_i^I			[0.377,0.377]	0.382	
C200ml-I CF_i^{II}			[0.019,0.019]	0.360	
C200ml-I CF_i^{III}			[0.007,0.007]	0.365	
C200ml-I CF_i^{IV}			[0.022,0.022]	0.367	
C200ml-P CF_i^0	0.004	**0.549**	[0.009,0.010]	**0.405**	-
C200ml-P CF_i^I			[0.385,0.385]	0.399	
C200ml-P CF_i^{II}			[0.005,0.005]	0.413	
C200ml-P CF_i^{III}			[0.006,0.006]	0.412	
C200ml-P CF_i^{IV}			[0.005,0.005]	0.406	

TABLE 3
Means of 100 repetitions of the generalized GFS for the imprecise datasets “BLong-4”, “100ml-4-I”, “100ml-4-P”, “B200ml-I” and “B200ml-P” with 5 labels/variable.

bootstrapped datasets used in the extended GFS. The only difference is that we have replaced each imprecise value by its middle point.

To compare these algorithms we have used only two datasets: “C200ml-I” and “C200ml-P”. These are the only datasets that have crisp outputs, thus we can evaluate the error of all crisp and fuzzy data-based models by means of a crisp numerical value. For example, in the Figure 7 if we compare the medians of the dataset “B200ml-I” with “C200ml-I”, in the original GFS the median of the dataset “C200ml-I” is similar at the median of the maximum fitness the “B200ml-I”. However in the extended GFS the median of “C200ml-I” is in the middle-upper between the median of the minimum and maximum fitness the “B200ml-I”. The same happens with the datasets “B200ml-P” and “C200ml-P”.

The test results are shown in Table 4, and in the Figure 8 the boxplots showing the relevance differences between crisp algorithms and the generalized GFS are included too. Observe that the results of the extended GFS are uniformly better than the remaining crisp and rule-based algorithms, showing that we have captured better the information in the real world low quality data.

	Low Quality	Crisp					
Dataset	Exh.Test	Linear	Neural	KNN	WM	PM	ISH
C200ml-I CF_i^0	**0.392**	0.512	0.602	0.584	0.462	0.473	0.483
C200ml-I CF_i^I	0.382						
C200ml-I CF_i^{II}	**0.360**						
C200ml-I CF_i^{III}	0.365						
C200ml-I CF_i^{IV}	0.367						
C200ml-P CF_i^0	**0.405**	0.541	0.635	0.467	0.430	0.450	0.515
C200ml-P CF_i^I	**0.399**						
C200ml-P CF_i^{II}	0.413						
C200ml-P CF_i^{III}	0.412						
C200ml-P CF_i^{IV}	0.406						

TABLE 4
Comparations of the means of 100 repetitions of the generalized GFS and other crisp algorithms for the imprecise datasets “C200ml-I” and “C200ml-P”.

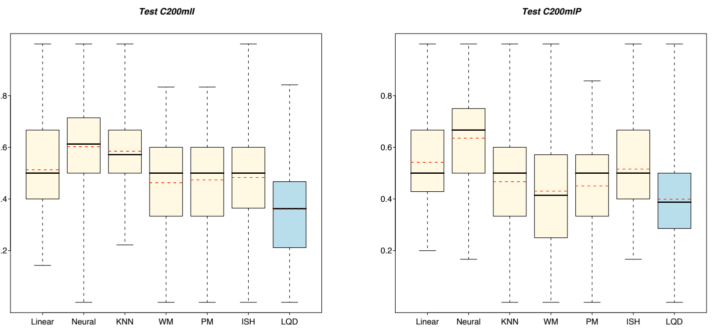


FIGURE 8
Boxplots illustrating the results of the 100 repetitions of generalized GFS respect to crisp algorithms for the imprecise datasets “C200ml-I” and “C200ml-P”.

7 CONCLUDING REMARKS

In this work we have shown that the use of a possibilistic representation has allowed us to obtain linguistic models that exploit the information in real world, low quality datasets in an efficient manner. We are aware, however, that there is still much room for improvement, as the difficulty of the problem is high. The number of athletes in the team (25) is too low for obtaining an adequately sized knowledge base and the percentage of wrong classifications produced by any GFS is still too high for this model being an alternative to the expert knowledge of the trainer, who can however use the system as an aid to his/her decision.

ACKNOWLEDGEMENTS

This work was supported by the Spanish Ministry of Education and Science, under grants TIN2008-06681-C06-04, TIN2007-67418-C03-03, and by Principado de Asturias, under grant PCTI 2006-2009.

REFERENCES

[1] Chen, S. Analysis of maximum total return in the continuous knapsack problem with fuzzy object weights. Applied Mathematical Modelling 33 (7): 2927-2933 (2009)

[2] Cordón O, Jesus M.J, Herrera F. A proposal on reasoning methods in fuzzy rule-based classification systems. International Journal of Approximate Reasoning, 20 (1): 21-45 (1999)

[3] Couso, I., Sánchez, L. Higher order models for fuzzy random variables. Fuzzy Sets and Systems 159: 237-258 (2008)

[4] Hand, D. J. Discrimination and Classification. Wiley Series in Probability and Mathematical Statistics, Chichester (1981)

[5] Haykin, S. Neural Networks: a comprehensive foundation, 2nd Edition. Prentice Hall. (1999)

[6] Cordón O, Herrera F, Hoffmann F, Magdalena L, Genetic fuzzy systems. Evolutionary tuning and learning of fuzzy knowledge bases. World Scientific, Singapore (2001)

[7] Ishibuchi, H., Nakashima, T., Murata, T, A fuzzy classifier system that generates fuzzy ifthen rules for pattern classification problems. In Proc. of 2nd IEEE International Conference on Evolutionary Computation: 759-764 (1995)

[8] Ishibuchi, H., Takashima, T., Effect of rule weight in fuzzy rule-based classification systems. IEEE Transactions on Fuzzy Systems 3 (3): 260-270 (2001)

[9] Ishibuchi, H., Yamamoto, T., Rule weight specification in fuzzy rule-based classification systems. IEEE Transactions on Fuzzy Systems 13 (4): 428-435 (2005)

[10] Limbourg, P., Multi-objective optimization of problems with epistemic uncertainty. Lecture Notes in Computer Science 3410, Evolutionary Multi-criterion Optimization EMO 2005: 413-427. (2005)

[11] Pal, S. K., Mandal, D. P. "Linguistic recognition system based in approximate reasoning". Information Sciences 61: 135-161 (1992)

[12] Palacios, A., Couso, I., Sánchez, L. GFS-based analysis of vague databases in High Performance Athletics. Lecture Notes in Computer Science 5788, Intelligent Data Engineering and Automated Learning IDEAL 2009: 620-609 (2009)

[13] Palacios, A., Sánchez, L., Couso, I. A baseline genetic fuzzy classifier based on low quality data. Proc IFSA-EUSFLAT 2009: 803-808 (2009)

[14] Palacios, A., Sánchez, L., Couso, I. A minimum-risk genetic fuzzy classifier based on low quality data. Lecture Notes in Computer Science 5572, Hybrid Artificial Intelligence Systems IDEAL 2009: 654-661 (2009)

[15] Palacios, A., Sánchez, L., Couso, I. Extending a simple Genetic Cooperative-Competitive Learning Fuzzy Classifier to low quality datasets. Evolutionary Intelligence 2 (1): 73-90 (2009).

[16] Palacios, A., Sánchez, L., Couso, I. Diagnosis of dyslexia with Low Quality Data with Genetic Fuzzy Systems. Int. J. Approximate Reasoning. Admitted for publication.

[17] Palacios Martín, J. L. Comunicación personal. (2009).

[18] Reglamento Internacional de Atletismo. Ed. Escolar A. G. (1995)

[19] Sánchez, L., Couso, I., Casillas, J. Modelling vague data with genetic fuzzy systems under a combination of crisp and imprecise criteria Proc. 2007 IEEE Symp. on Comp. Int. in Multicriteria Decision Making, Honolulu, USA: 30-37. (2007)

[20] Sánchez, L., Couso, I., Casillas, J. Genetic learning of fuzzy rules based on low quality data. Fuzzy Sets and Systems. 160 (17): 2524-2552 (2009)

[21] Teich J., Pareto-front exploration with uncertain objectives. Lecture Notes in Computer Science 1993, Evolutionary Multi-Criterion Optimization EMO 2001: 314-328. (2001)

[22] Vinuessa, M., Coll., J. Tratado de atletismo. Servicio Geográfico del Ejercito Español. (1984)

[23] Wang L.X., Mendel J.M., Generating fuzzy rules by learning from examples. IEEE Transactions on Systems, Man, and Cybernetics 22 (6): 1414-1427 (1992)