International Journal of Approximate Reasoning 51 (2010) 993–1009

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

journal homepage: www.elsevier.com/locate/ijar

Diagnosis of dyslexia with low quality data with genetic fuzzy systems

Ana M. Palacios [a], Luciano Sánchez [a,*], Inés Couso [b]

[a] Universidad de Oviedo, Departamento de Informática, Campus de Viesques, 33071 Gijón, Asturias, Spain
[b] Universidad de Oviedo, Departamento de Estadstíca e I.O. y D.M., Campus de Viesques, 33071 Gijón, Asturias, Spain

A R T I C L E   I N F O

Article history:
Received 31 December 2009
Received in revised form 19 July 2010
Accepted 27 July 2010
Available online 7 August 2010

Keywords:
Genetic fuzzy systems
Low quality data
Dyslexia

A B S T R A C T

For diagnosing dyslexia in early childhood, children have to solve non-writing based graphical tests. Human experts score these tests, and decide whether the children require further consideration on the basis of their marks.

Applying artificial intelligence techniques for automating this scoring is a complex task with multiple sources of uncertainty. On the one hand, there are conflicts, as different experts can assign different scores to the same set of answers. On the other hand, sometimes the experts are not completely confident with their decisions and doubt between different scores. The problem is aggravated because certain symptoms are compatible with more than one disorder. In case of doubt, the experts assign an interval-valued score to the test and ask for further information about the child before diagnosing him.

Having said that, exploiting the information in uncertain datasets has been recently acknowledged as a new challenge in genetic fuzzy systems. In this paper we propose using a genetic cooperative–competitive algorithm for designing a linguistically understandable, rule-based classifier that can tackle this problem. This algorithm is part of a web-based, automated pre-screening application that can be used by the parents for detecting those symptoms that advise taking the children to a psychologist for an individual examination.

1. Introduction

Dyslexia can be defined as a learning disability in people with normal intellectual coefficient, and without further physical or psychological problems that can explain such disability. According to the Orton Society [38],

> Dyslexia is a neurologically based, often familial, disorder which interferes with the acquisition and processing of language [...]. Although dyslexia is lifelong, individuals with dyslexia frequently respond successfully to timely and appropriate intervention.

In this research we are interested in the early diagnosis (ages between 6 and 8) of schoolchildren of Asturias (Spain), where this disorder is not rare. It has been estimated that between 4% and 5% of these schoolchildren have dyslexia. The average number of children in a Spanish classroom is 25, therefore there are cases at most classrooms [1]. Notwithstanding the widespread presence of dyslexic children, detecting the problem at this stage is a complex process, that depends on many different indicators, mainly intended to detect whether reading, writing and calculus skills are being acquired at

* Corresponding author.
E-mail address: luciano@uniovi.es (L. Sánchez).

0888-613X/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.ijar.2010.07.008

the proper rate. Moreover, there are disorders different than dyslexia that share some of their symptoms and therefore the tests not only have to detect abnormal values of the mentioned indicators; in addition, they must also separate those children which actually suffer dyslexia from those where the problem can be related to other causes (inattention, hyperactivity, etc.).

All schoolchildren at Asturias are routinely examined by a psychologist that can diagnose dyslexia (in Table 1 there is a list of the tests that are applied in Spanish schools for detecting this problem). Nevertheless, an early detection of the problem can allay the treatment. We are in the first stages of the design of an automated pre-screening application that can be used by the parents of preschoolers for detecting those symptoms that advise taking their children to a psychologist for an individual examination. We want that this automated application includes a fuzzy rule based system (FRBS), whose knowledge base is to be obtained from a sample of labelled data by means of a genetic algorithm. Using Soft Computing techniques for diagnosing dyslexia seems to us a natural choice, because of the properties of our data (linguistic terms, and vague measurements). As a matter of fact, there are many references where fuzzy techniques were used to learn medical diagnosis models from data. In particular, in [15,23], fuzzy techniques have been used in the diagnosis of disabilities in language. However, in all of the preceding works, the data was crisp or categorical. Instead, most of our measurements are not crisp, as we will discuss later. Moreover, a high percentage of cases have missing values. None of the cited approaches are directly applicable to the problem at hand.

For building this FRBS, we have collected data from 65 schoolchildren during our research, comprising their answers to the tests in Fig. 1. In addition, each case has been individually classified by a psychologist into one or more of the classes "no dyslexia", "control and revision", "dyslexic" and "other disorders" (inattention, hyperactivity, etc.). It is remarked that we have not tried to remove the uncertainty in our data prior to the learning, but we will introduce a novel rule extraction algorithm that is able to exploit better the information contained in these low quality sets of data. Summarizing, in this paper we will propose a new methodology for designing an augmented genetic fuzzy system (GFS) that can operate with low quality data, and apply the resulting knowledge base for determining, on the basis of the answers to certain graphical tests, whether a preschooler should be diagnosed by an expert. In a machine learning context, this means that we will design a classifier that is able to operate when we cannot accurately observe all the properties of the object. In the most simple case (interval-valued data) we will perceive sets that contain these values. In the general case, we will be given a nested family of sets, each one of them containing the true value with a probability higher or equal than its level; we will represent each of these families by means of a fuzzy set, with a possibilistic interpretation.

In the remaining of the paper we will discuss in detail the particular properties of the data originated in the mentioned tests (Section 2), the use of fuzzy sets for representing this data (Section 3), how this data can be fed to a FRBS to produce a set of outputs (Section 4), the measurement of the performance of a FRBS with this data, and how to genetically optimize it (Section 5). In Section 6 we will apply the new algorithm to different benchmark and real-world problems, and compare the results to those of the crisp algorithms and with previous works. In Section 7 we conclude the paper and discuss future work in the subject.

Table 1
Categories of the tests currently applied in Spanish schools for detecting dyslexia in children between 6 and 8 years.

Category	Test	Description
Verbal comprehension	BAPAE [10]	Vocabulary
	BADIG [41]	Verbal orders
	BOEHM [5]	Basic concepts
Logic reasoning	RAVEN [28]	Color
	BADIG [41]	Figures
	ABC [13]	Actions and details
Memory	Digit WISC-R [40]	Verbal-additive memory
	BADIG [41]	Visual memory
	ABC [13]	Auditive memory
Level of maturation	ABC [13]	Combination of different tests
Sensory-motor skills	BENDER [2]	Visual-motor coordination
	ABD [14]	Motor coordination
	BADIG [41]	Perception of shapes
	BAPAE [10]	Spatial relations, Shapes, Orientation
	STAMBACK [33]	Auditive perception, Rhythm
	HARRIS/HPL [17]	Laterality, Pronunciation
	ABC [13]	Pronunciation
	GOODENOUGHT [16]	Spatial orientation, Body scheme
Attention	Toulose [35]	Attention and fatigability
	ABC [13]	Attention and fatigability
Reading–Writing	TALE [39]	Analysis of reading and writing

2. Subjectiveness and uncertainty in the evaluation of a test

At the present time, an automated classification is not being used and an expert in dyslexia or psychologist is needed for diagnosing each case. Take, for instance, the sensory-motor skills test "BENDER" [2]. In this test, the child has to copy the geometric drawings displayed in Fig. 1, and the expert has to score each copy depending on the differences with the original.

In Fig. 2 we have included actual copies of the fifth drawing of Bender's test by eight children in our study. Interestingly enough, for evaluating these differences the human expert follows by hand an algorithm described by a list of linguistic "if-then" rules (see the aforementioned Ref. [2] for a complete description of these). In this case, the expert has to decide whether the angles, relative positions, orientations and other geometrical properties have been accurately copied or not. In particular, the evaluation criterion about the angles for this fifth drawing is given by the rules that follow:

- If both angles measure 90° and both small curves are equal then assign 3 points.
- If one angle does not measure 90° or one small curve is not well copied then assign 2 points.
- If neither angle measures 90° or none of the small curves are well copied then assign 1 point.
- If one angle or one small curve is missing then assign 0 points.

The authors of this rule-based algorithm did not envision the use of approximate reasoning techniques for interpolating between the scores, however it is clear that there is still a degree of subjectiveness in the task of scoring a figure and arguably this means that there is uncertainty on the input data. In particular, we have found that

1. it is possible that two experts assign a different number of points to the same drawing,
2. it is natural for some experts to assign a range of values to a drawing (i.e. "the score of the drawing is between 1 and 2") and
3. sometimes the experts prefer using a linguistic term (i.e. "near 2").

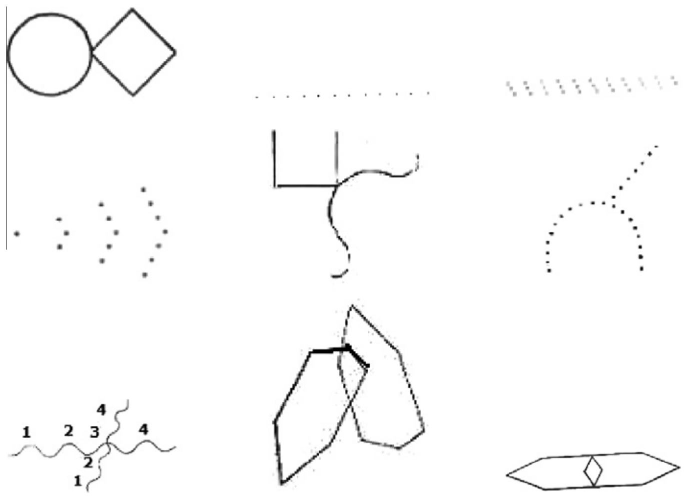

Fig. 1. Example of Bender's tests for detecting dyslexia. This test contains nine geometric drawings that the child has to copy.

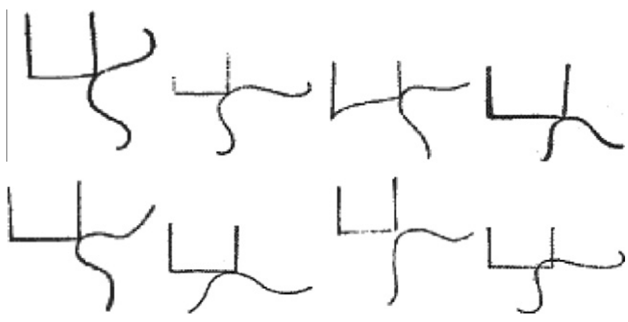

Fig. 2. Different copies about the fifth figure and the expert has to decide whether the figure is correct or not.

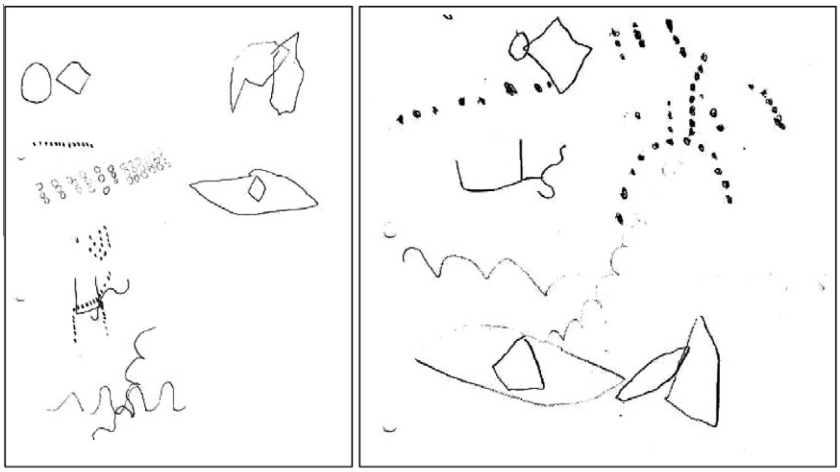

Fig. 3. Left: Bender's test of one child assigned to the class "no dyslexia". Right: The same test solved by a child for who the expert could not decide between the classes "no dyslexia" and "control and revision".

In our experimentation we have given the experts freedom for expressing the score of the tests to their best convenience, thus our data comprises a mix of numerical, interval-valued and fuzzy data.

Furthermore, this imprecision in the scoring propagates through the output of the algorithm and sometimes the expert doubts about the diagnosing. For example, in the left part of Fig. 3 and also in Table 2 we have displayed the scoring assigned

Table 2
Results of the test the one child diagnosed as "no dyslexia".

	Lower	Medium–lower	Medium	Medium–higher	Higher
1. Verbal comprehension					
Vocabulary	( )	( )	( )	(×)	( )
Verbal orders	( )	( )	( )	(×)	( )
Basic concepts	( )	( )	( )	( )	(×)
2. Reasoning	( )	( )	( )	( )	(×)
3. Visual-motor coordination	( )	( )	(×)	( )	( )
4. Verbal memory	( )	( )	( )	( )	(×)
5. Perceptive ability					
Perception of shapes	( )	( )	( )	( )	(×)
Spatial orientation	( )	( )	( )	(×)	( )
6. Auditive perception rhythm	( )	( )	( )	( )	(×)
7. Laterality	Right	Left	Crusade	Ambidextrous	
	(×)	( )	( )	( )	
8. Pronunciation	Correct				

Table 3
Results of the test where the expert could not decide between the classes "no dyslexia" and "control and revision". Observe that we might as well have classified the child as "dyslexic".

	Lower	Medium–lower	Medium	Medium–higher	Higher
1. Verbal comprehension					
Vocabulary	(×)	( )	( )	( )	( )
Verbal orders	( )	(×)	( )	( )	( )
Basic concepts	( )	( )	(×)	( )	( )
2. Reasoning	( )	( )	(×)	( )	( )
3. Visual-motor coordination	( )	(×)	( )	( )	( )
4. Verbal memory	( )	( )	(×)	( )	( )
5. Perceptive ability					
Perception of shapes	( )	( )	(×)	( )	( )
Spatial orientation	( )	( )	( )	( )	(×)
6. Auditive perception rhythm	( )	(×)	( )	( )	( )
7. Laterality	Right	Left	Crusade	Ambidextrous	
	(×)	( )	( )	( )	
8. Pronunciation	Correct				

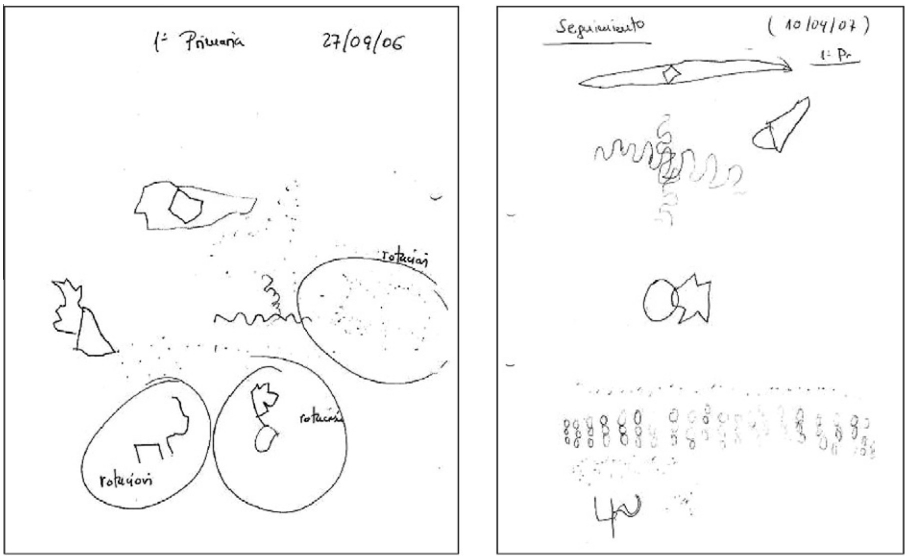

Fig. 4. Bender's test of one child assign in the "class {2,4}". The left picture is doing in the beginning of the course and the picture of the right in the end of the course.

by the expert to a child that she labelled as "no dyslexic". By contrast, in Table 3 and in the right part of the same Fig. 3 we can see the drawing of a child for who the expert could not decide between "no dyslexic" and "control and revision" (this last label means that the individual is marked as suspicious and the test is repeated at the end of the term). Comparing the evaluations in Tables 2 and 3 we observe that this last child is near the group average in most tests, but below average in others. While the total score might suggest that the child does not suffer from dyslexia, some of the results could be interpreted otherwise, thus the labeling of the individual with multiple classes provides us with more information than any of the alternatives.

Another example is shown in Fig. 4, where we have represented two Bender's tests made at the beginning of the course (6 year old child) and at the end, seven months later. This expert was in doubt about the assignment of this child to the classes "dyslexic" or "attention disorder/hyperactivity". In this case, an early decision is very important in order to choose an adequate education but in the first evaluation we lack information for completely deciding which label to choose. Having these cases in mind, we have allowed that the expert marks any individual with as many labels as he/she wish.

3. Fuzzy sets, aggregated data and metainformation

In the preceding section we have stated that our dataset contains uncertain values in both output and input variables. On the one hand, each child might be assigned more than one label, thus the desired output of the system may be partially known. On the other hand, in the input side, we have found cases where the expert prefers using a range of numbers or a linguistic value. To this we can add those cases where more than one expert evaluates the same case and they do not agree in the score, or those where the result of a test is missing.

In these situations, an aggregated value signals that our information is incomplete. For example, a child who is assigned two class labels does not suffer from two different problems; we are only stating that we cannot make a finer distinction. Accordingly, a child who has not been diagnosed would be labelled with the set of all class marks. With respect to the input values, a score of [1,2] in a test means that the actual score is an unknown number between 1 and 2. Lastly, those cases where the input value is linguistic or we have unreconciled scores by different experts have a similar semantic; there is a only partial knowledge about the actual value of the attribute.

In this work we will unify the treatment of all these cases by means of a possibilistic representation. We will use fuzzy sets for representing metainformation, understood as our knowledge about the uncertainty of the data. In particular, we will admit that we cannot perceive the actual value of a variable, neither we have a complete knowledge about the probability distribution of the difference between the perceived and the actual values. This incomplete knowledge will be modeled with a fuzzy set, which we will identify with the family of all the probability distributions for which the alpha-cuts of this fuzzy set have a probability mass higher or equal than 1-alpha [7]. Observe that this includes the interval and the crisp situations as particular cases. For instance, the interval [1,2] mentioned before is regarded as the family of probability distributions whose support is contained in [1,2]. Missing values are represented by an interval that spans the whole range of the variable. It is also easy to assign a meaning to the linguistic value "near 2" that we mentioned in the preceding section, or to reconcile different values of the same attribute, building a membership function from a bootstrap estimation of the distribution of a central tendency measure of the data, as described in [31].

Fuzzy datasets like these originated in our interpretation are the main research area in fuzzy statistics, however this kind of information is seldom considered in genetic fuzzy systems (GFS) [6,12,18,27,30]. GFSs obtain fuzzy rule based systems (FRBS) from crisp data, and the role of fuzzy sets in a FRBS is to model vague asserts, using fuzzy logic-based tools. Fuzzy logic techniques are not, generally speaking, compatible with those fuzzy statistical techniques used for modeling vague observations of variables [3,29], thus we need to design an extension of the fuzzy logic-based reasoning methods that closes the gap between standard FRBS and the use of fuzzy sets for modeling metainformation, as described before. This extension will be explained in the next section.

4. An extension principle-based reasoning method

In this section we discuss how to compute the output of an FRBS, given a vague input. At a first glance, this should consist in computing the cylindrical extension of the input, intersecting it with the fuzzy graph implicitly defined by the FRBS, and projecting this intersection on the output space. However, this procedure is not valid, because its result can be a nonnormal fuzzy set, that has not possibilistic meaning. In this section we adapt a reasoning method, that was proposed in [31] for fuzzy models, to the classification case and extend it to weighted fuzzy rules.

Let us make clear the problem with the help of a particular case; consider a fuzzy classifier comprising M rules:

$$\textbf{if } (x \textbf{ is } \widetilde{A}_i) \textbf{ then class is } C_i \textbf{ with confidence } w_i, \tag{1}$$

and let us use the single-winner inference mechanism. In the first place, let us suppose that we have a crisp perception x of the properties of an object. Its class is, therefore,

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

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

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

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

$$\text{class}(X) = \left\{ C_{\arg\max_i \{\widetilde{A}_i(x)\cdot w_i\}} | x \in X \right\} \tag{4}$$

or,

$$\text{class}(X) = \{\text{class}(x) | x \in X\}, \tag{5}$$

which is different than Eq. (3).

In other words, the use of a fuzzy logic-based reasoning method allows us to combine the set of linguistic rules defined in Eq. (1) with a crisp perception x, for obtaining a precise conclusion, as shown in Eq. (2). The same fuzzy logic-based reasoning could have been applied for deriving the most appropriate class when our perception is vague ("$x \in X$"). In this case, we would have obtained the class given by Eq. (3). However, it may happen that there exist a value $x_0 \in X$ such that class$'(X) \neq$ class(x_0). We want to avoid this discrepancy and therefore we are defining class(X) as the set of all possible values for class(x), when $x \in X$.

Our reasoning method is as follows: Let X be the input space, let N_c be the number of classes, thus $K = \{1, \ldots, N_c\}$ is the output space, and let $\{\widetilde{A}_i \to (C_i, w_i)\}_{i=1,\ldots,M}$ be a set of M fuzzy rules. Recall that, given a crisp input $x \in X$, the most common reasoning method for computing the output of a FRBS takes two stages [6]:

1. An intermediate fuzzy set is composed:

$$\widetilde{\text{out}}(x)(k) = \max_{\substack{i=1,\ldots,M \\ C_i=k}} \widetilde{A}_i(x) \cdot w_i. \tag{6}$$

2. This intermediate fuzzy set is transformed in a crisp value defuz$(\widetilde{\text{out}}(x)) \in K$ by means of a suitable defuzzification operator. In classification problems, the "maximum" defuzzification is mostly used. Therefore, the value defuz$(\widetilde{\text{out}}(x)) \in K$ is often equivalent to

$$\text{defuz}(\widetilde{\text{out}}(x)) = \arg\max_k \{\widetilde{\text{out}}(x)(k)\}. \tag{7}$$

The extension to set valued inputs is as follows: Given an input $A \subseteq X$ (that, in our context, means "all we know about the input is that it is in the set A"),

1. We determine a family of intermediate fuzzy sets in the universe $\mathscr{F}(K)$, $\widetilde{\text{out}}(A) \in \wp(\mathscr{F}(K))$, defined as

$$\widetilde{\text{out}}(A) = \{\widetilde{\text{out}}(x) \text{ s.t. } x \in A\}. \tag{8}$$

2. An element of $\wp(K)$ (that is to say, a set of crisp outputs $\mathrm{defuz}(\widetilde{\mathrm{out}}(A)) \in \wp(K)$) is obtained, according to the following definition:

$$\mathrm{defuz}(\widetilde{\mathrm{out}}(A)) = \{\mathrm{defuz}(\widetilde{\mathrm{out}}(x)) \text{ s.t. } x \in A\}. \tag{9}$$

Lastly, given a fuzzy input $\widetilde{A} \in \mathscr{F}(X)$, we will assign it, according to the extension principle (which is compatible with the possibilistic interpretation of fuzzy sets) a fuzzy set computed as follows:

1. We determine an intermediate fuzzy set on the universe $\mathscr{F}(K)$, $\widetilde{\mathrm{out}}(\widetilde{A}) \in \mathscr{F}(\mathscr{F}(K))$, defined as

$$\widetilde{\mathrm{out}}(\widetilde{A})(\widetilde{B}) = \sup\{\widetilde{A}(x) \text{ s.t. } \widetilde{\mathrm{out}}(x) = \widetilde{B}\}, \quad \forall \widetilde{B} \in \mathscr{F}(K). \tag{10}$$

2. An element of $\mathscr{F}(K)$ (that is to say, a fuzzy output) $\mathrm{defuz}(\widetilde{\mathrm{out}}(\widetilde{A})) \in \mathscr{F}(K)$ is obtained as follows:

$$\mathrm{defuz}(\widetilde{\mathrm{out}}(\widetilde{A}))(k) = \sup\{\widetilde{A}(x) \text{ s.t. } \mathrm{defuz}(\widetilde{\mathrm{out}}(x)) = k\}, \quad \forall k \in K. \tag{11}$$

Observe that the fuzzy set $\mathrm{defuz}(\widetilde{\mathrm{out}}(\widetilde{A}))$ is associated to the nested family of sets $\{\mathrm{defuz}(\widetilde{\mathrm{out}}(\widetilde{A}_\alpha))\}_{\alpha\in[0,1]}$, and that explains the possibilistic interpretation of this procedure.

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

The weights of the rules will be obtained through extensions of the four heuristic methods defined in [20], that in the remaining of the paper will be denoted CF^{I}, CF^{II}, CF^{III}, and CF^{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 this kind of rules in problems with low quality data. This extension is as follows:

Definition 1. 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 [20]:

$$c(A_i \Rightarrow C_i)_{(x_1,c_1,\ldots,x_m,c_m)} = \frac{\sum_{c_p=C_i}\mu_{A_i}(x_p)}{\sum_{p=1}^{m}\mu_{A_i}(x_p)}. \tag{12}$$

For a low quality (fuzzy) dataset $\{(\widetilde{X}_1, c_1), \ldots, (\widetilde{X}_m, c_m)\}$, we will define the confidence of a rule as the direct application of the extension principle to Eq. (12), which is the fuzzy subset of $[0,1]$ given by

$$\widetilde{c}(A_i \Rightarrow C_i)_{(\widetilde{X}_1,c_1,\ldots,\widetilde{X}_m,c_m)}(t) = \max\left\{\min\{\mu_{X_1}(x_1),\ldots,\mu_{X_m}(x_m)\}|c(A_i \Rightarrow C_i)_{(x_1,c_1,\ldots,x_m,c_m)} = t\right\}. \tag{13}$$

5. Definition of the extended genetic fuzzy system

The GFS that we will generalize to vague data in this section was introduced in [19], and we have chosen this algorithm because of its balance between simplicity and performance. This approach can be regarded as a genetic cooperative–competitive learning (GCCL) [18], where the complete population encodes the rule base, and the chromosomes compete and cooperate simultaneously. On the one hand, the cooperation between individuals arises because the fitness of a chromosome is defined as the count of instances of the training set that are well classified by the rule encoded in it. That way, in case that there are not duplicates, the sum of the fitness of the individuals equals the number of well classified instances in the whole training set by the rule base, which is encoded in the whole population, as mentioned. On the other hand, the competition is based on the survival of the fittest; those rules that cover a higher number of instances have better chances of being selected for recombination.

The outer loop of the GFS (whose pseudocode can be found in the Appendix A) consists in selecting two parents with the help of a double binary tournament, recombining and mutating them and inserting the offspring into a secondary population whose size is smaller than that of the primary population. The worse individuals of the primary population will be eventually replaced by those in the secondary population at each generation. In our implementation, duplicates are not removed, but assigned fitness zero for preserving the cooperation. We will not include in this short explanation details about the representation of individuals, genetic operators or the initialization of the initial population, as these do not depend on the input data being crisp or fuzzy and can be found elsewhere [19–21]. Let us only remark that this algorithm does not codify the consequent of the fuzzy rules in the chromosomes, that are restricted to the antecedent part. The best consequent is always set to the alternative with a higher confidence.

Generalizing this GCCL-based GFS to imprecise data involves changes to the inference mechanism, that we have discussed in Section 4, and also to the fitness function [29]. Therefore, in the remainder of this section, we will study how to assign consequents when the confidence is not a number, as discussed before, and how to determine the fitness when the input and output variables are intervals or fuzzy intervals. That is to say, we will discuss

1. new procedures for assigning of the consequents,
2. computing set-valued fitness functions, and
3. the genetic selection and replacement of the worst individuals, including a short discussion about the meaning of "best" and "worst" when the fitness is a set-valued function.

5.1. Assignment of consequents

In the crisp version of the GFS, consequents are determined by computing the confidences $c(A_i \Rightarrow k)$ for each class $k = 1,\ldots,N_c$, then selecting the alternative with maximum confidence and assigning that value to C_i. In this paper the weight of a rule and the assignment of consequents depends on the defuzzified values of the fuzzy confidences $\widetilde{c}(A_i \Rightarrow k)$, as defined in Eq. (13).

The computation of this set is computationally costly. Nevertheless it is contained in the set obtained by replacing the arithmetic operators in Eq. (12) by their corresponding fuzzy arithmetic counterparts, and we will use this last approximation in our experiments.

Once the set-valued confidences have been computed, it is needed to rank them in order to find the best alternative; in other words, we need to choose one of the values in the set of nondominated confidences. The word "dominates" can have different meanings, ranging from the strict dominance (A dominates B iff $a < b$ for all $a \in A$, $b \in B$) [34] to other definitions that induce a total order in the set of confidences. In this paper, we have used the uniform dominance defined in [24], that induces a total order and thus the set of nondominated consequents has size 1.

5.2. Computation of fitness

The error of the FRBS at an imprecisely perceived object is a fuzzy set, and therefore the genetic optimization of the classifier cannot be designed for finding the minimum value of the fitness function (which we cannot precisely determine) but to discard those chromosomes whole quality is not optimal, keeping a set of nondominated individuals. In other words, we want to capture the set of minimal elements in a partial order defined between fuzzy intervals. This problem is closely related to certain cases in preference modeling [36], imprecise probabilities and possibility theory-based generalizations of stochastic orderings [9,11,32,37,42] or recent approaches in the context of label ranking [4].

The computation of the fitness of a chromosome is described now. Let us suppose first that the ith object of the training set is perceived through a crisp set. The output of the FRBS will be the set of classes

$$C_{\mathrm{FRBS}}(X_i) = \left\{ C_{\arg\max_j \{\widetilde{A}_j(x)\cdot w_j\}} | x \in X_i \right\}. \tag{14}$$

Accordingly, for a fuzzy value $\widetilde{X}_i$ the output is the fuzzy subset of $\{1,\ldots,N_c\}$ that follows:

$$\widetilde{C}_{\mathrm{FRBS}}(\widetilde{X}_i)(k) = \max\{\alpha | k \in C_{\mathrm{FRBS}}([X_i]_\alpha)\} \tag{15}$$

for $k \in \{1,\ldots,N_c\}$. It can be inferred that the theoretical expression of the fitness function of the FRBS is:

$$\widetilde{f} = \oplus \widetilde{e}_i, \tag{16}$$

where $\oplus$ is the fuzzy arithmetic-based sum [22]. $\widetilde{e}_i$ is a fuzzy subset of $\{0,1\}$, whose α-cuts are:

$$[\widetilde{e}_i]_\alpha = \begin{cases} 1 & C_{\mathrm{FRBS}}([X_i]_\alpha) = C_i \text{ and } \#(C_i) = 1, \\ 0 & C_{\mathrm{FRBS}}([X_i]_\alpha) \cap C_i = \emptyset, \\ \{0,1\} & \text{else,} \end{cases} \tag{17}$$

where $\#(\cdot)$ is the cardinality operator. In words, if the output of the FRBS is a single class label that matches the class label of the example, this point scores 1. If the set of classes emitted by the FRBS does not intersect with that of the object, this point scores 0. Otherwise, it scores the set $\{0,1\}$.

The evaluation of this function is computationally very expensive, and we will use an approximation for interval-valued data. The approximated algorithm computes an interval of values for the matching between each rule and the input,

$$\mathrm{match}(\widetilde{A}_j, [X_i]_\alpha) = \{\widetilde{A}_j(x)\cdot w_j | x \in [X_i]_\alpha\} \tag{18}$$

then discards all rules that can not be the winner rule. The set of possible winners is

$$\mathrm{winner}([X_i]_\alpha) = \{j | \mathrm{match}(\widetilde{A}_j, [X_i]_\alpha) \preceq \mathrm{match}(\widetilde{A}_r, [X_i]_\alpha) \text{ for all } r\}. \tag{19}$$

Lastly, the output of the FRBS is approximated by the set of the consequents of the non-discarded rules

$$C_{\mathrm{FRBS}}([X_i]_\alpha) \approx \{C_r | r \in \mathrm{winner}([X_i]_\alpha)\}. \tag{20}$$

This set includes the theoretical output, but sometimes it also includes extra class labels. We will use the approximated value for guiding the genetic learning but apply instead a Monte-Carlo based approximation for determining the result of

Eq. (15), which is more precise but slower, for assessing the results of the learning. The pseudocodes of both functions are included in the Appendix A.

5.3. Genetic selection and replacement

There are two other parts in the original algorithm that must be altered in order to use an imprecise fitness function: (a) the selection of the individuals in [19] is based on a tournament, that depends on a total order on the set of fitness values. And (b) the same happens with the removal of the worst individuals.In both cases, we have used the uniform dominance defined in [24] to impose such a total order.

6. Numerical results

In this part we apply the algorithms described in the paper to different subsets of the data that we have described in Section 2. The purpose of this study is double:

1. Comparing the new approach (learning rules for diagnosing dyslexia from vague data with a genetic algorithm that uses a fuzzy valued fitness function) to the results of a crisp valued fitness function-based GFS.
2. Determining whether our data contains information enough for any of these GFSs to produce a powerful enough FRBS, than can be used by unqualified personnel for screening prescholars and detect dyslexia as early as possible.

Therefore, we want to isolate the benefits of using fuzzy sets-based metainformation in this particular problem, and determine whether there is a statistically significant improvement in the learning process due to the use of a fuzzy fitness function. To achieve this, we have built a crisp version of each dataset by removing the uncertainty in both input and output variables. We have applied the original algorithm in [19,20] to these crisp versions, and the results have been compared with those of the approach presented here. Details about the procedures for removing the fuzziness of the measurements are also given in the following.

6.1. Description of the datasets

We have considered five different datasets regarding this experimentation. Their names are "Dyslexic-12", "Dyslexic-12-01", "Dyslexic-12-12", "Dyslexic-11-01" and "Dyslexic-11-12". The first three datasets contain vague data in both input and output variables, and in the last two we have avoided the use of fuzzy sets and intervals for representing data, thus they have precise inputs and vague outputs. It is remarked that all these problems are real-world data and thus we do not know the best attainable error with a FRBS.

The output variable for each of these datasets is a subset of the labels that follow:

- No dyslexic.
- Control and revision.
- Dyslexic.
- Inattention, hyperactivity or other problems.

It is remarked that the class "control and revision" is not, properly speaking, a valid diagnosis. Being intermediate between "no dyslexic" and "dyslexic" it could have been represented by a set of classes, however the expert expressed her concerns about some particular properties of these cases, thus we have decided to make a compared analysis where these individuals are treated separately first and later they are integrated into either alternative.

Concretizing, the datasets "Dyslexic-12", "Dyslexic-12-01", "Dyslexic-12-12" have 65 objects and 12 features. There is imprecision in both the input and the output, and also missing values. These three datasets differ only in their outputs:

- "Dyslexic-12" comprises the four mentioned classes.
- "Dyslexic-12-01" does not make use of the class "control and revision", whose members are included in class "no dyslexic".
- "Dyslexic-12-12" does not make use of the class "control and revision", whose members are included in class "dyslexic".

The last two datasets, "Dyslexic-11-01" and "Dyslexic-11-12" have, as we have mentioned, crisp inputs and imprecise outputs. We have not considered, in this last case, the class "control and revision"; summarizing, the datasets "Dyslexic-11" are two:

- "Dyslexic-11-01" does not make use of the class "control and revision", whose members are included in class "no dyslexic".

- "Dyslexic-11-12" does not make use of the class "control and revision", whose members are included in class "dyslexic".

Both datasets have 65 objects, 3 classes, and 11 features. There is imprecision in the output but not in the input, and there are missing values.

6.2. Experimental settings

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, amounting to approximately 10,000 evaluations. The fuzzy partitions of the

Table 4
Means of 100 repetitions of the generalized GFS for the imprecise datasets "Dyslexic-12", "Dyslexic-12-01" and "Dyslexic-12-12" with four and five labels/variable.

Dataset	Crisp		Low quality		Low quality [26]
	Train	Test	Fitness train	Test error (Exh.)	Test error (Exh.)
Dyslexic-12 (four labels) CF^0	0.444	[0.572, 0.694]	[0.003, 0.237]	[0.405, 0.548]	[0.421, 0.558]
Dyslexic-12 (four labels) CF^{I}	0.430	[0.572, 0.692]	[0.072, 0.303]	[0.413, 0.553]	
Dyslexic-12 (four labels) CF^{II}	0.426	[0.566, 0.687]	[0.007, 0.244]	[0.423, 0.564]	
Dyslexic-12 (four labels) CF^{III}	0.483	[0.584, 0.695]	[0.066, 0.244]	[0.448, 0.555]	
Dyslexic-12 (four labels) CF^{IV}	0.487	[0.591, 0.700]	[0.052, 0.244]	[0.450, 0.559]	
Dyslexic-12 (five labels) CF^0	0.556	[0.614, 0.731]	[1.538, 0.233]	[0.480, 0.621]	[0.490, 0.609]
Dyslexic-12 (five labels) CF^{I}	0.560	[0.612, 0.730]	[0.086, 0.308]	[0.487, 0.629]	
Dyslexic-12 (five labels) CF^{II}	0.557	[0.614, 0.731]	[3.076, 0.238]	[0.480, 0.620]	
Dyslexic-12 (five labels) CF^{III}	0.581	[0.618, 0.728]	[0.065, 0.238]	[0.505, 0.608]	
Dyslexic-12 (five labels) CF^{IV}	0.582	[0.621, 0.731]	[0.057, 0.238]	[0.504, 0.610]	
Dyslexic-12-01 (four labels) CF^0	0.336	[0.452, 0.553]	[0.005, 0.193]	[0.330, 0.440]	[0.219, 0.759]
Dyslexic-12-01 (four labels) CF^{I}	0.340	[0.456, 0.556]	[0.083, 0.268]	[0.344, 0.450]	
Dyslexic-12-01 (four labels) CF^{II}	0.348	[0.460, 0.559]	[0.007, 0.198]	[0.338, 0.449]	
Dyslexic-12-01 (four labels) CF^{III}	0.377	[0.472, 0.562]	[0.065, 0.197]	[0.359, 0.440]	
Dyslexic-12-01 (four labels) CF^{IV}	0.383	[0.472, 0.563]	[0.066, 0.199]	[0.367, 0.444]	
Dyslexic-12-01 (five labels) CF^0	0.460	[0.508, 0.605]	[0.0, 0.187]	[0.394, 0.522]	[0.323, 0.797]
Dyslexic-12-01 (five labels) CF^{I}	0.458	[0.507, 0.605]	[0.088, 0.264]	[0.398, 0.527]	
Dyslexic-12-01 (five labels) CF^{II}	0.466	[0.509, 0.607]	[0.001, 0.192]	[0.393, 0.520]	
Dyslexic-12-01 (five labels) CF^{III}	0.488	[0.515, 0.604]	[0.066, 0.192]	[0.413, 0.503]	
Dyslexic-12-01 (five labels) CF^{IV}	0.487	[0.518, 0.608]	[0.067, 0.193]	[0.414, 0.502]	
Dyslexic-12-12 (four labels) CF^0	0.390	[0.511, 0.664]	[0.003, 0.243]	[0.325, 0.509]	[0.199, 0.757]
Dyslexic-12-12 (four labels) CF^{I}	0.376	[0.506, 0.659]	[0.049, 0.280]	[0.332, 0.516]	
Dyslexic-12-12 (four labels) CF^{II}	0.391	[0.503, 0.658]	[0.005, 0.245]	[0.341, 0.523]	
Dyslexic-12-12 (four labels) CF^{III}	0.418	[0.517, 0.667]	[0.028, 0.244]	[0.354, 0.508]	
Dyslexic-12-12 (four labels) CF^{IV}	0.424	[0.520, 0.670]	[0.023, 0.246]	[0.362, 0.516]	
Dyslexic-12-12 (five labels) CF^0	0.485	[0.539, 0.692]	[0.0, 0.239]	[0.393, 0.591]	[0.211, 0.700]
Dyslexic-12-12 (five labels) CF^{I}	0.484	[0.540, 0.692]	[0.050, 0.280]	[0.399, 0.599]	
Dyslexic-12-12 (five labels) CF^{II}	0.485	[0.539, 0.691]	[0.0, 0.240]	[0.396, 0.593]	
Dyslexic-12-12 (five labels) CF^{III}	0.515	[0.538, 0.688]	[0.023, 0.240]	[0.411, 0.574]	
Dyslexic-12-12 (five labels) CF^{IV}	0.513	[0.540, 0.690]	[0.024, 0.240]	[0.408, 0.571]	

Table 5
Means of 100 repetitions of the generalized GFS and original GFS for the dataset "Dyslexic-11-01" and "Dyslexic-11-12".

Dataset	Crisp		Low quality		Low quality [26]
	Train	Test	Fitness train	Test error (Exh.)	Test error (Exh.)
Dyslexic-11-01 (four labels) CF^0	0.332	[0.423, 0.546]	[0.017, 0.237]	[0.332, 0.473]	[0.375, 0.566]
Dyslexic-11-01 (four labels) CF^{I}	0.321	[0.424, 0.546]	[0.019, 0.224]	[0.327, 0.477]	
Dyslexic-11-01 (four labels) CF^{II}	0.316	[0.418, 0.540]	[0.016, 0.221]	[0.322, 0.471]	
Dyslexic-11-01 (four labels) CF^{III}	0.338	[0.440, 0.548]	[0.106, 0.221]	[0.342, 0.454]	
Dyslexic-11-01 (four labels) CF^{IV}	0.353	[0.432, 0.540]	[0.106, 0.222]	[0.341, 0.450]	
Dyslexic-11-12 (four labels) CF^0	0.350	[0.516, 0.675]	[0.022, 0.263]	[0.384, 0.571]	[0.491, 0.600]
Dyslexic-11-12 (four labels) CF^{I}	0.355	[0.524, 0.684]	[0.025, 0.254]	[0.399, 0.581]	
Dyslexic-11-12 (four labels) CF^{II}	0.350	[0.517, 0.677]	[0.029, 0.258]	[0.403, 0.584]	
Dyslexic-11-12 (four labels) CF^{III}	0.388	[0.528, 0.684]	[0.070, 0.256]	[0.427, 0.575]	
Dyslexic-11-12 (four labels) CF^{IV}	0.387	[0.522, 0.676]	[0.068, 0.256]	[0.418, 0.568]	

labels are uniform and their size depends of the problem, as shown in Tables 4 and 5. All the experiments are repeated 100 times for bootstrap resamples with replacement of the training set. The test set comprises the "out of the bag" elements.

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

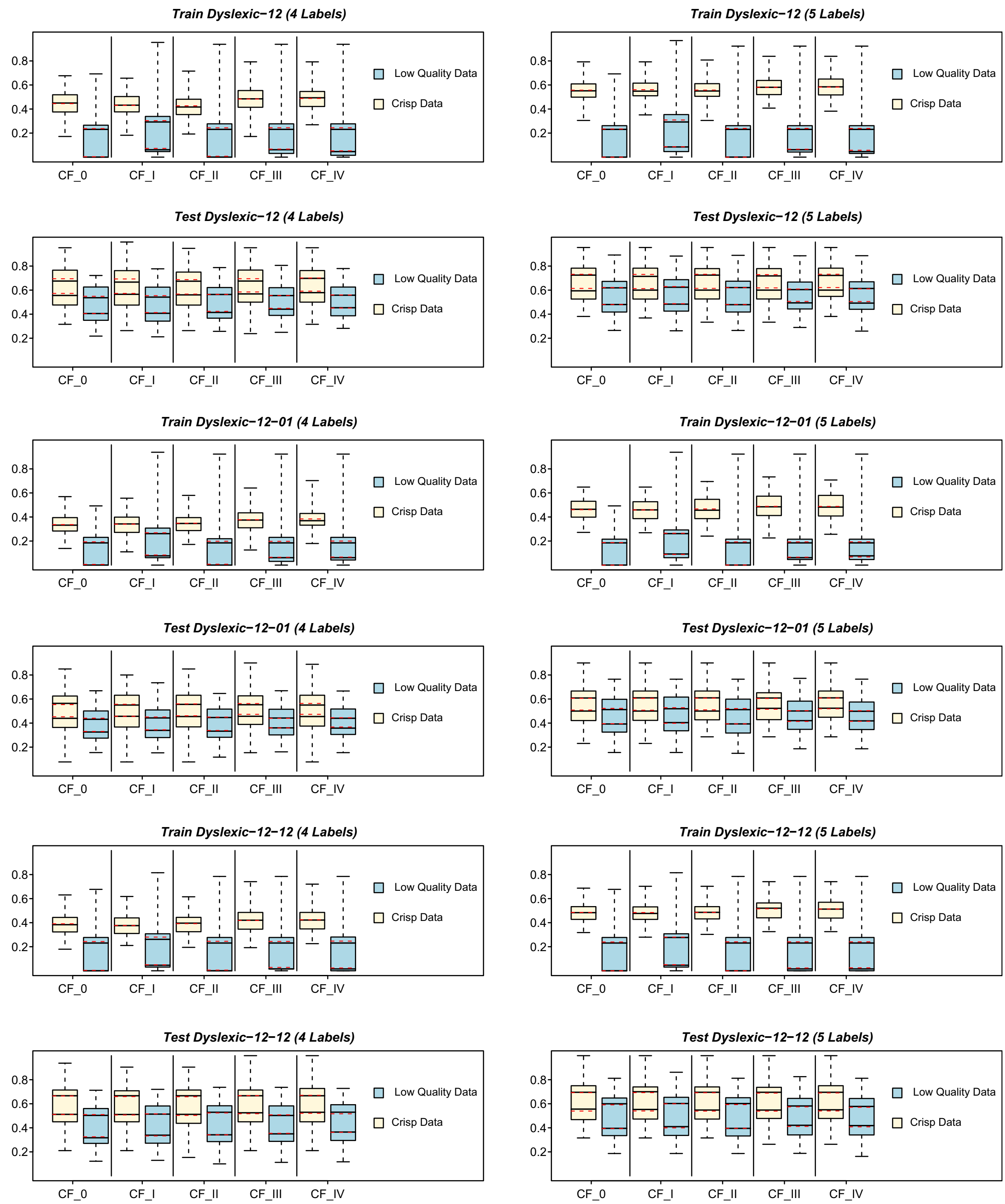

Fig. 5. Boxplots illustrating the distance of the 100 repetitions of original and extended GFS in the problem "Dyslexic-12", "Dyslexic-12-01" and "Dyslexic-12-12" with 4/5 labels.

6.2.1. Methodology for comparing results between crisp and low quality data-based algorithms

Comparing results of crisp and low quality data-based algorithms is difficult, as it requires using a method for removing the observation error from the data, and an inadequate removal may distort the results. In this section we have applied the following rules, proposed first in [26], for producing crisp data from imprecise data:

- For removing the uncertainty of an input variable, intervals are replaced by its midpoint and fuzzy sets are replaced by their center of gravity.
- For removing the imprecision of an output variable, each sample is been replicated so many times as different alternatives exist, as described in [25]. Each replication is assigned a degree of importance such that the contribution of the example to the total fitness is not influenced by the number of replicas. For instance, an example ($x = 2$, $C = \{A, B\}$) is converted in two examples ($x = 2$, $C = A$) and ($x = 2$, $C = B$), and each one of them is assigned an importance 0.5. It is remarked that this use of weighted instances (not to be confused with the weights in the consequent of the rules that has also been introduced here) required a minor change of the original algorithm (see Appendix A). Lastly, it is remarked that this procedure essentially assumes a prior uniform probability distribution on the output variable for those cases that are labelled with more than one class.

6.2.2. Statistical significance of results and graphical comparison

The statistical comparison between samples of fuzzy data is not a mature field yet [8] and there is still some controversy in the definition of the most appropriate statistical tests. We have decided not to compute an interval of p-values (as proposed in the mentioned reference and cites contained therein) but to make a graphical representation, based on boxplots, instead. In the remaining of this section we will use two different presentations of the results:

1. Tables: In the "crisp" columns we represent the error of the original GFS [19] with weights in both examples and rules. Conversely, in the "low quality data" columns, we show the results of the learning: the mean fitness (training error) of the 100 repetitions, and an interval that contains the test error, computed with the exhaustive algorithm described in Fig. A.12. The column "low quality [26]" contains the results of an earlier GFS proposed by us in [26].
2. Boxplots: The boxplots are not standard because we represent crisp and imprecise results. We propose using a box showing the 75% percentile of the maximum and the 25% percentile of the minimum error (thus the box displays 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.

6.3. Analysis of the results and discussion

The compared results of the crisp GFS, the fuzzy fitness driven version and our former algorithm in [26] are shown in Tables 4 and 5, and the statistical significance of the differences is displayed in Figs. 5 and 6. As a first conclusion, the new algorithm is a significant improvement over the crisp GFS, showing that, in this problem, it is preferable to use an algorithm which is capable of learning rules with low quality data than removing the interval or fuzzy information and using a conventional algorithm. Of a secondary importance, the use of weights in the consequents does not seem to noticeably improve the power of the classification system. If weights are to be used, the most effective strategy for assigning a confidence degree to the consequent of a rule is, in this case, CF^{III}. However, the use of weights for dealing with multi-labelled instances

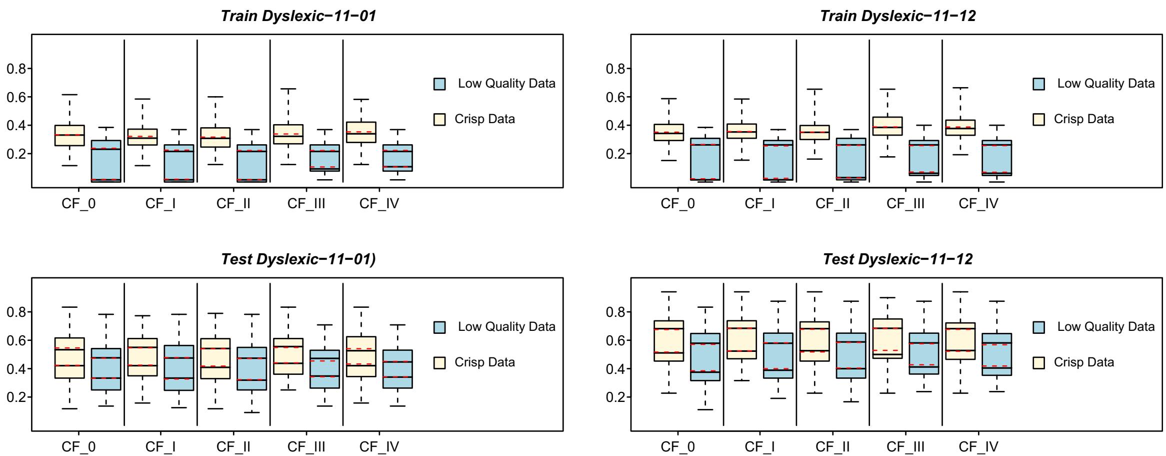

Fig. 6. Boxplots illustrating the distance of the 100 repetitions of crisp and generalized GFS in the dataset "Dyslexic-11-01" and "Dyslexic-11-12".

outperforms the approach used in previous works [26] (which was based in instance duplication), as deduced from the comparison between the two last columns in Tables 4 and 5; notice that the upper bound of the test error has been consistently decreased in all the experiments.

From a different scope, we have found that the FRBS is powerful enough for separating the dyslexic children from those with hyperactivity, attention disorders or other problems. The differences between the results in the three datasets shown in Table 4 seem also to support the necessity of the class "control and revision" as a fourth category in the diagnose. This can be seen, for instance, in the results of the dataset "Dyslexic-12" and four labels, that are similar to those of "Dyslexic-12-01" and better than "Dyslexic-12-12". We have also observed that, if the class "control and revision" is not used, the individuals in this class tend to be assigned to the group "no dyslexic" rather than to the group "dyslexic". In other words, even though the most probable evolution of a child in this class is towards the absence of dyslexia, the existence of this group is recommended, as it permits following the evolution of these possibly problematic schoolchildren.

With respect to the results in the datasets "Dyslexic-11-01" and "Dyslexic-11-12" in Table 5 and Fig. 6 (these should be compared to "Dyslexic-12-01 (four labels)" and 'Dyslexic-12-12 (four labels)" in Table 4, respectively), the first conclusion is that the classification power of the system decreases when the uncertain attributes are not used in the input variables, and the difference with respect to the crisp algorithm is also smaller, confirming our hypothesis that the use of fuzzy metainformation during the learning can improve the results of the FRBS.

To conclude this section, the main objective of the extended GFS in the scope of this research was to obtain a FRBS from low quality data that can be used by unqualified personnel to detect whether a children has suspicious symptoms and then suggest consulting with the psychologist. We consider that this objective is mostly achieved, but the percentage of misclassifications is still high and there is room for improvements. Being a pre-screening, in future works, we intend to use techniques of imbalanced classification and cost-guided classification (Bayesian minimum risk classifiers) in order to obtain an adequate screening, where the probability of a dyslexic student is not being detected can be bounded by a low enough value.

7. Concluding remarks

The relevance of the use of fuzzy metadata in the diagnosis of dyslexia is related to the high cost of the data acquisition. Obtaining data from 65 children required months of work, thus any progress that we can make for better exploiting of the data justifies the use of a rather complex learning algorithm as the proposed in this paper. We have found that the new technique makes a difference and allows us to draw sounder conclusions from the same data than standard GFSs. In addition, in this paper we have introduced some minor changes on the basic GCCL algorithm for imprecise data: the use of weighted instances for dealing with multi-labelled instances and the use of weights in the consequents. The first change has proven itself effective, not so the second.

Acknowledgements

This work was supported by the Spanish Ministry of Education and Science, under Grants TIN2008-06681-C06-04, TIN2010-20900-C04-01 and by Principado de Asturias, under Grant PCTI 2006-2009.

Appendix A. Pseudocode of the algorithms

In this appendix we provide a detailed description of the algorithms referenced in the text. The outer loop of the GCCL-based GFS is listed in Fig. A.7. It can be seen that it depends on two additional functions: "assignConsequent" (called in line 6)

```
function GFS
1    Initialize population
2    for iter in {1, ..., Iterations}
3      for sub in {1, ..., subPop}
4        Select parents
5        Crossover and mutation
6        assignConsequent(offspring)
7      end for sub
8      Replace the worst subPop individuals
9      assignFitness(population,dataset)
10   end for iter
11   Purge unused rules
return population
```

Fig. A.7. Outline of the GFS that will be generalized [19]. Each chromosome codifies one rule. The fitness of the classifier is distributed among the rules at each generation.

```
function assignConsequent(rule)
    for example in {1, ..., N}
       m = membership(Antecedent,example)
       weight[class[example]] = weight[class[example]] + m
    end for example
    mostFrequent = 0
    for c in {1, ..., N_c}
       if (weight[c]>weight[mostFrequent]) then
          mostFrequent = c
       end if
    end for c
    Consequent = mostFrequent
    CF[rule] = computeConfidenceOfConsequent
return rule
```

Fig. A.8. The consequent of a rule is not codified in the GA, but it is assigned the most frequent class label, between those compatible with the antecedent of the rule [19].

```
function assignFitness(population,dataset)
    for example in {1, ..., N}
       winnerRule = 0
       bestMatch = 0
       for rule in {1, ..., M}
          m = membership(Antecedent[rule],example)*CF[rule]
          if (m>bestMatch) then
             winnerRule = rule
             bestMatch = m
          end if
       end for rule
       if (consequent(winnerRule)==class(example)) then
          fitness[winnerRule] = fitness[winnerRule] + 1
       end if
    end for example
return fitness
```

Fig. A.9. The fitness of an individual is the number of examples that it classifies correctly. Single-winner inference is used, thus at most one rule changes its fitness when the rule base is evaluated in an example [19].

```
function assignImpreciseConsequent(rule)
1       for c in {1, ..., N_c}
2          grade = 0
3          compExample = 0
4          for example in {1, ..., N}
5             m̃ = fuzMembership(Antecedent,example,c)
6             grade = grade ⊕ m̃
7             if (sup {x : m̃(x) > 0} > 0) then
8                compExample = compExample + 1
9             end if
10         end for example
11         weight[c] = grade ⊘ compExample
12      end for c
13      mostFrequent = {1, ..., N_c}
14      for c in {1, ..., N_c}
15         for c_1 in {c+1, ..., 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)
22      CF[rule] = computeConfidenceOfConsequent
return rule
```

Fig. A.10. If the examples are imprecise, we might not know the most frequent class label – lines 13–20. In this paper we have used the dominance proposed in [24] to reduce this set to one element.

```
function assignImpreciseFitnessApprox(population,dataset)
1  for example in {1,...,N}
2    setWinnerRule = ∅
3    for r in {1,...,M}
4      dominated = FALSE
5      r.m̃ = fuzMembership(Antecedent[r],example)*CF[r]
6      for sRule in setWinnerRule
7        if (sRule dominates r) then
8          dominated = TRUE
9        end if
10     end for sRule
11     if (not dominated and r.m̃ > 0) then
12       for sRule in setWinnerRule
13         if (r.m̃ dominates sRule) then
14           setWinnerRule = setWinnerRule −{ sRule }
15         end if
16       end for sRule
17       setWinnerRule = setWinnerRule ∪{ r }
18     end if
19   end for r
20   if (setWinnerRule == ∅) then
21     setWinnerRule = setWinnerRule ∪{ rule_freq_class }
22   setOfCons= ∅
23   for sRule in setWinnerRule
24     setOfCons= setOfCons ∪{ consequent(sRule) }
25   end for sRule
26   deltaFit= 0
27   if ({class(example)} == setOfCons and
         size(setOfCons)==1) then
28     deltaFit = {1}
29   else
30     if ({class(example)}∩ setOfCons ≠ ∅) then
31       deltaFit = {0, 1}
32     end if
33   end if
34   Select winnerRule ∈ setWinnerRule
35   fitness[winnerRule] = fitness[winnerRule] ⊕ deltaFit
36 end for example
return fitness
```

Fig. A.11. Generalization of the function "assignFitness" to imprecise data. If the example is imprecisely perceived, there are three ambiguities that must be resolved: (a) some different crisp values compatible the same example might correspond to different winner rules – lines 3–19, (b) these rules might have different consequent, thus we do not know if the rule base fails in the example – lines 22–33 and (c) we must assign credit to just one of these rules – lines 34 and 35.

and "assignFitness" (line 9). The crisp version of both functions are also listed in Figs. A.8 and A.9. The function "assignConsequent" determines the class label that matches an antecedent with a maximum confidence. The function "assignFitness," in turn, determines the winner rule for each object in the training set and increments the fitness of the corresponding individual if its consequent matches the class of the object.

The crisp assignment of consequents seen in Fig. A.8 is extended to fuzzy data in Fig. A.10. Notice that the original assignment consisted in 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 the extended algorithm, the weight of a rule and the assignment of consequents depends on the defuzzified value of the confidence defined in Eq. (13), which is approximated as explained in lines 4–11 of Fig. A.10. Observe that the operation "dominates" used in line 16 of Fig. A.10 can be assigned different meanings, as mentioned. If the uniform dominance defined in [24] is used, the set of nondominated consequents has size 1.

The first extension of the function that assigns the fitness to a chromosome is displayed in Fig. A.11, that includes the approximation described in Eqs. (18)–(20). This approximation has not been used to compute the test results but to guide the genetic learning. In Fig. A.12 we have included the slower but more accurate Monte-Carlo approximation mentioned in the text.

Lastly, in order to process weighted instance sets (see Section 6.2.1) a minor change is needed when computing the matching between an example and a rule (line 2 in function "assignConsequent", Fig. A.13) and when accumulating the fitness of a chromosome (line 12 in function "assignFitness" in the same figure).

```
function assignImpreciseFitnessExhaustive(population,dataset)
1  for dataset in {1, ..., 1000}
2    fitness[dataset] = 0
3    for example in {1, ..., N}
4      bestMatch = 0
5      WRule = -1
6      for r in {1, ..., M}
7        m = membership(Antecedent[r],example)*CF[r]
8        if (m > bestMatch) then
9          WRule = r
10         bestMatch = m
11       end if
12     end for r
13     if (WRule == -1) then
14       WRule = rule_fre_class
15     end if
16     if (consequent(WRule) == class(example)) then
17       score = 1
18     else
19       if consequent(WRule) ⊂ class(example)) then
20         score= {0, 1}
21       end if
22     end if
23     fitness[dataset] = fitness[dataset] ⊕ score
24   end for example
25 end for dataset
26 fitness=0
27 for dataset in {1, ..., 1000}
28   fitness=fitness ⊕ fitness[dataset]
29 end for dataset
30 fitness=mean(fitness)
return fitness
```

Fig. A.12. Other generalization of the function "assignFitness" to interval-valued 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. Lines 16–20 deal with the case where an object has imprecise output, i.e. "the class is A or C"; otherwise, the value of the variable "score" is crisp.

```
function assignConsequent(rule)
1    for example in {1, ..., N}
2      m = membership(Antecedent,example) * ω(example)
3      weight[class[example]] = weight[class[example]] + m
4    end for example
5    mostFrequent = 0
6    for c in {1, ..., N_c}
7      if (weight[c]>weight[mostFrequent]) then
8        mostFrequent = c
9      end if
10   end for c
11   Consequent = mostFrequent
return rule

function assignFitness(population,dataset)
1    for example in {1, ..., N}
2      winnerRule = 0
3      bestMatch = 0
4      for rule in {1, ..., M}
5        m = membership(Antecedent[rule],example)*CF[rule]
6        if (m>bestMatch) then
7          winnerRule = rule
8          bestMatch = m
9        end if
10     end for rule
11     if (consequent(winnerRule)==class(example)) then
12       fitness[winnerRule] = fitness[winnerRule] + ω(example)
13     end if
14   end for example
return fitness
```

Fig. A.13. The original algorithm in [19] is altered as shown in lines 2 (function assignConsequent) and 12 (function assignFitness) so that is able to learn from a database where each example has a fractional degree of importance "ω(example)."

References

[1] J. Ajuriaguerra, Manual de psiquiatría infantil, Toray-Masson, 1976 (in Spanish).
[2] L. Bender, Test Guestáltico Visomotor, Paidós, Buenos Aires, 1982 (in Spanish).
[3] P. Bonissone, J.M. Cadenas, M.C. Garrido, R.A. Díaz-Valladares, A fuzzy random forest, International Journal of Approximate Reasoning, in press, doi:10.1016/j.ijar.2010.02.003.
[4] K. Brinker, J. Fürnkranz, E. Hüllermeier, A unified model for multilabel classification and ranking, source frontiers in artificial intelligence and applications, in: Proceedings of the 17th European Conference on Artificial Intelligence ECAI 2006, 2006, pp. 489–493.
[5] A.E. Boehm, BOEHM. Tets de Boehm de Conceptos Básicos, TEA Ediciones, Madrid, 1967 (in Spanish).
[6] O. Cordón, F. Herrera, F. Hoffmann, L. Magdalena, Genetic Fuzzy Systems. Evolutionary Tuning and Learning of Fuzzy Knowledge Bases, World Scientific, Singapore, 2001.
[7] I. Couso, L. Sánchez, Higher order models for fuzzy random variables, Fuzzy Sets and Systems 159 (2008) 237–258.
[8] I. Couso, L. Sánchez, Defuzzification of fuzzy p-values, Advances in Soft Computing: Soft Methods for Handling Variability and Imprecision 48 (2009) 126–132.
[9] I. Couso, L. Sánchez, The behavioral meaning of the median, in: Borgelt et al. (Eds.), Combining Soft Computing and Statistical Methods in Data Analysis, Advances in Intelligent and Soft Computing, vol. 77, Springer, 2010.
[10] M.V. De la Cruz, Batería de Aptitudes para el Aprendizaje Escolar, Publicaciones de Psicologa Aplicada, TEA Ediciones, Madrid, 1989 (in Spanish).
[11] B. De Schuymer, H. De Meyer, B. De Baets, S. Jenei, On the cycle-transitivity of the dice model, Theory and Decision 54 (2003) 261–285.
[12] A. Fernández, M.J. Del Jesus, F. Herrera, Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets, International Journal of Approximate Reasoning 50 (3) (2009) 561–577.
[13] L. Filho, Test A.B.C, Paidós, Barcelona, 1937 (in Spanish).
[14] M. Frostig, Test de desarrollo de la percepción visual, TEA, Madrid, 1978 (in Spanish).
[15] V. Georgopulos, A fuzzy cognitive map to differential diagnosis of specific language impairment, Artificial Intelligence in Medicine 29 (2003) 261–278.
[16] F.L. Goodenought, Test de la figura humana, Paidós, Barcelona, 1991 (in Spanish).
[17] A.J. Harris, Test de dominancia lateral, TEA, Madrid, 1978 (in Spanish).
[18] F. Herrera, Genetic fuzzy systems: taxonomy, current research trends and prospects, Evolutionary Intelligence 1 (2008) 27–46.
[19] H. Ishibuchi, T. Nakashima, T. Murata, A fuzzy classifier system that generates fuzzy if-then rules for pattern classification problems, in: Proceedings of the of 2nd IEEE International Conference on Evolutionary Computation, 1995, pp. 759–764.
[20] H. Ishibuchi, T. Takashima, Effect of rule weight in fuzzy rule-based classification systems, IEEE Transactions on Fuzzy Systems 3 (3) (2001) 260–270.
[21] H. Ishibuchi, T. Yamamoto, Rule weight specification in fuzzy rule-based classification systems, IEEE Transactions on Fuzzy Systems 13 (4) (2005) 428–435.
[22] A. Kaufmann, M.M. Gupta, Introduction to Fuzzy Arithmetic: Theory and Applications, Van Nostrand Reinhold, 1991.
[23] D.V. Lakov, Soft computing agent approach to remote learning of disables, in: 2nd IEEE International Conference on Intelligent Systems, 2004, pp. 250–255.
[24] P. Limbourg, Multi-objective optimization of problems with epistemic uncertainty, in: EMO, 2005, pp. 413–427.
[25] A. Palacios, L. Sánchez, I. Couso, A baseline genetic fuzzy classifier based on low quality data, in: Proceedings of the IFSA-EUSFLAT, 2009, pp. 803–808.
[26] A. Palacios, L. Sánchez, I. Couso, Extending a simple genetic cooperative-competitive learning fuzzy classifier to low quality datasets, Evolutionary Intelligence 2 (1) (2009) 73–90.
[27] P. Pulkkinen, H. Koivisto, Fuzzy classifier identification using decision tree and multiobjective evolutionary algorithms, International Journal of Approximate Reasoning 48 (2) (2008) 526–543.
[28] J.C. Raven, El test de matrices progresivas, Paidós, Barcelona, 1997 (in Spanish).
[29] L. Sánchez, I. Couso, Advocating the use of imprecisely observed data in genetic fuzzy systems, IEEE Transactions on Fuzzy Systems 15 (4) (2007) 551–562.
[30] L. Sanchez, M.R. Suarez, J.R. Villar, I. Couso, Mutual information-based feature selection and partition design in rule-based classifiers from vague data, International Journal of Approximate Reasoning 49 (3) (2008) 607–622.
[31] L. Sánchez, I. Couso, J. Casillas, Genetic learning of fuzzy rules based on low quality data, Fuzzy Sets and Systems. 160 (17) (2009) 2524–2552.
[32] M. Shaked, J.G. Shanthikumar, Stochastic Orders, Springer, 2007.
[33] M. Stamback, Tono y psicomotricidad, Pablo del Río, Madrid, 1978 (in Spanish).
[34] J. Teich, Pareto-front exploration with uncertain objectives, in: EMO, 2001, pp. 314–328.
[35] E. Toulose, H. Piéron, Test Toulouse-Piéron. Prueba perceptiva y de Atencin, 1904 (in Spanish).
[36] M. Öztürk, A. Tsoukias, Valued Hesitation in intervals comparison, in: Proceedings of the SUM-07 Conference, LNAI, vol. 4772, Springer-Verlag, 2007, pp. 157–170.
[37] M.C.W. Troffaes, Decision making under uncertainty using imprecise probabilities, International Journal of Approximate Reasoning 45 (2007) 17–19.
[38] P. Thomson, P. Gilchrist, Dyslexia: A Multidisciplinary Approach, Nelson Thornes, 1996.
[39] J. Toro, M. Cervera, TALE Test de análisis de la lectoescritura, Pablo del Ro, Madrid, 1980 (in Spanish).
[40] D. Wechsler, WISC Escala de inteligencia de Wechsler para niños-revisada, TEA, Madrid, 1993 (in Spanish).
[41] C. Yuste, BADYG-E: Bateria de aptitud diferencial y general, CEPE, Madrid, 1985 (in Spanish).
[42] M. Zaffalon, K. Wesnes, O. Petrini, Reliable diagnoses of dementia by the naive credal classifier inferred from incomplete cognitive data, Artificial Intelligence in Medicine 29 (2003) 61–67.