16th World Congress of the International Fuzzy Systems Association (IFSA)
9th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT)

FuGePSD: Fuzzy Genetic Programming-based algorithm for Subgroup Discovery

C.J. Carmona[1] **P. González**[2] **M.J. del Jesus**[2]

[1]Department of Civil Engineering, Languages and Systems Area, University of Burgos, 09001, Burgos (Spain)
[2]Department of Computer Science, University of Jaén, 23071, Jaén (Spain)

Abstract

Evolutionary Fuzzy Systems (EFSs) are fuzzy systems augmented by a learning process based on evolutionary computation such as evolutionary algorithms (EAs). These systems contribute with several advantages in the development of algorithms, and specifically in the development of subgroup discovery (SD) approaches. SD is a descriptive data mining technique using supervised learning in order to describe data with respect to a property of interest.

This paper present the main features of the FuGePSD algorithm, an EFS based on genetic programming and fuzzy logic. An experimental study with a wide number of datasets shows the quality of this algorithm with respect to the remaining EFSs for SD presented throughout the literature.

Keywords: Genetic Programming, Subgroup Discovery, Evolutionary Fuzzy System

1. Introduction

SD is a descriptive data mining technique for describing unusual features with respect to a variable of interest (or target variable) [1, 2]. This data mining technique contributes with interesting knowledge to the scientific community from different points of view: interest, generality and precision. Throughout the literature, there is a wide number of SD algorithms based on exhaustive strategies (such as CN2-SD [3], Apriori-SD [4], SD-Map [5]), or genetic algorithms (such as SDIGA [6], MESDIF [7] and NMEEF-SD [8]), amongst others.

This contribution is focused on the use of EFSs for SD because this type of systems are very suitable in order to develop algorithms to solve this task. An EFS can be described as a fuzzy system [9] augmented with a learning process based on evolutionary computation [10]. Fuzzy systems are very suitable for knowledge representation in SD. This is because they are usually considered in the form of fuzzy-rule based systems, which is the more common knowledge representation in SD. In addition, fuzzy logic allow to consider uncertainty, and represent the continuous variables in a manner which is close to human reasoning. In this way, interpretable fuzzy rules consider continuous variables as linguistic ones where values are represented through fuzzy linguistic labels (LLs). On the other hand, evolutionary computation is a well known and widely used global search technique with the ability to explore a large search space, which is usually the case in SD tasks, so it is a beneficial strategy to tackle SD problems.

Specifically, this contribution presents FuGePSD [11], an algorithm based on genetic programming [12] that employs a tree with a variable-length structure to represent the individuals of the population. This algorithm employs different mechanisms and specific operators in order to maximise the quality with respect to the SD algorithms presented up to the moment. Its benefits are highlighted in an experimental study using a wide number of datasets and its validity is analysed with respect to all the EFSs for SD presented throughout the literature.

To do so, the paper is organised as follows. Firstly, preliminary concepts are described in Section 2. Next, Section 3 presents the FuGePSD algorithm in which a complete description of the algorithm can be observed. Section 4 present all information related to the experimental framework and the study. Finally, Section 5 outlines the main conclusions.

2. Related Work

This section introduces main concepts used for the FuGePSD: SD is presented in Section 2.1 where the definition, main properties and quality measures for SD are outlined. Next, an introduction to EFSs together the main proposals based on EFSs for SD are presented in Section 2.2.

2.1. Subgroup Discovery

SD is a descriptive data mining technique based on supervised learning. The concept of SD was initially introduced by Kloesgen [1] and Wrobel [2]. The main purpose of SD is to seek and explore relationships between different properties or variables with respect to a target variable, and representations of the knowledge are performed through rules which consist of induced subgroup descriptions [13, 3]. Each rule R can be formally defined as:

$$R : Cond \; \rightarrow \; Target_{value}$$

where $Target_{value}$ is a value for the variable of interest (target variable) for the SD task (which also appears as *Class* in the literature), and *Cond* is commonly a conjunction of features (attribute-value pairs) which is able to describe an unusual statistical distribution with respect to the $Target_{value}$.

Despite the use of a target variable, SD is a descriptive induction task using supervised learning while classification is a predictive task. Main differences between SD and classification can be observed in [14].

The most important elements considered for an SD approach are [15]: the target variable, the search strategy, the descriptive language of the subgroups and the quality measures used. Reviews about major properties, features, algorithms and real-world problems solved through the application of SD algorithms can be found in [14, 16].

The most relevant quality measures used throughout the literature of SD are presented below:

- *Unusualness*: The weighted relative accuracy of a rule [17] measures interest and a trade-off between generality and precision. It can be computed as:

$$Unus(R_i) = \frac{n(Cond)}{n_s} \cdot \qquad (1)$$

$$\left(\frac{n(Target_{value} \cdot Cond)}{n(Cond)} - \frac{n(Target_{value})}{n_s}\right)$$

  It can be described as the balance between the coverage of the rule $p(Cond_i)$ and its accuracy gain $p(Target_{value} \cdot Cond) - p(Target_{value})$, where $n(Cond)$ is the number of examples which satisfy the conditions determined by the antecedent part of the rule, n_s is the number of total examples, $n(Target_{value} \cdot Cond)$ is the number of examples which satisfy the conditions and also belong to the value for the target variable within the rule, and $n(Target_{value})$ are all the examples of the target variable.
- *Sensitivity*: This measure is the proportion of actual matches that have been classified correctly [1]. Sensitivity has a component based on generality. It is computed as:

$$Sens(R_i) = \frac{n(Target_{value} \cdot Cond)}{n(Target_{value})} \qquad (2)$$

  This quality measure can be found in the literature as the Support based on the examples of the class, Recall or $TPrate$, and its domain is $[0, 1]$.
- *Confidence*: This measures the relative frequency of examples satisfying the complete rule amongst those satisfying only the antecedent for fuzzy rules [18]. It is computed as:

$$Conf(R_i) = \frac{n(Target_{value} \cdot Cond)}{n(Cond)} \qquad (3)$$

2.2. Evolutionary Fuzzy Systems

An EFS [19] is basically a fuzzy system augmented by a learning process based on evolutionary computation [10]. Specifically, fuzzy systems are usually considered in the form of fuzzy-rule based systems (FRBSs), which are composed of “IF-THEN” rules where both the antecedent and consequent can contain fuzzy logic statements. On the other hand, EAs are well known and widely used global search techniques with the ability to explore a large search space. Therefore, EAs can be used in the development of FRBSs offering a great potential as a search tool, allowing the inclusion of domain knowledge and the obtaining of better rules.

In summary, the properties of this type of systems make them highly suitable for the development of SD approaches. In fact, the use of fuzzy rules, based on fuzzy logic [9], already allow to consider uncertainty, and also to represent the continuous variables in a manner which is close to human reasoning. In this way, interpretable fuzzy rules consider continuous variables as linguistic ones, where values are represented through fuzzy linguistic labels (LLs).

Eq. 4 represents a canonical fuzzy rule:

$$R: \; IF \; X_1 = (LL_1^2) \;\; AND \;\; X_3 = (LL_3^1) \qquad (4)$$

$$THEN \;\; Target_{value}$$

where:

- $X = \{X_m / m = 1, \ldots, n_v\}$ is a set of features used to describe the subgroups, and n_v is the number of descriptive features.
- $T = \{Target_{value} / j = 1, \ldots, n_{tv}\}$ is a set of values for the target variable, and n_{tv} is the number of values for the target variable.
- $LL_{n_v}^{l_{n_v}}$ is the LL number l_{n_v} of the variable n_v.

The fuzzy set corresponding to each LL can be specified by the user or defined by means of uniform partitions if knowledge is not available. On the other hand, the most used schemes of representation for the EAs considered within EFS are “*Chromosome = rule*” and “*Chromosome = set of rules*” approaches.

Throughout the literature different EFSs for SD have been presented [16]. All of them use EAs for the search process, and are able to obtain models which are both simple and precise. These proposals are summarised below:

- SDIGA [6] is a mono-objective EFS for SD. This algorithm follows the IRL approach [20], a specific type of the “chromosome=rule” representation. It searches for rules for each value of the target variable. SDIGA is able to represent fuzzy canonical or DNF rules with a predefined set of linguistic labels. Fitness function is an aggregation function with different quality measures such as unusualness, sensitivity and confidence.

- MESDIF [21] is a multiobjective EFS following the SPEA2 approach [22]. It also uses the “chromosome=rule” representation. It is executed for each value of the target variable so obtaining the Pareto front for each of its values. Due to its multiobjective approach, the expert can choose between a wide number of quality measures in order to maximise the defined objectives.
- NMEEF-SD [8] is a multiobjective EFS following the NSGA-II [23] approach. It codifies each candidate solution according to the “chromosome=rule” approach. The consequent is prefixed to one of the possible values of the target variable and so NMEEF-SD is executed as many times as the number of different values of the target variable. It is able to represent canonical and DNF fuzzy rules and allows to choose up to three quality measures as objectives of the evolutionary process.

3. Fuzzy Genetic Programming-based algorithm for Subgroup Discovery

The Fuzzy Genetic Programming-based algorithm for Subgroup Discovery, FuGePSD [11], is an EFS based on a genetic programming algorithm [12] with the ability to extract descriptive fuzzy rules for the SD task.

Firstly, it is important to note the representation and the approach used in the algorithm. FuGePSD represents individuals through the “chromosome=rule” approach including both the antecedent and the consequent of the rule. In this way, this algorithm is executed only once obtaining rules for the different values of the target variable in opposite to the remaining EFSs for SD presented in the literature.

FuGePSD utilises a context-free grammar which allows the learning of fuzzy rules and the absence of some input features, a process giving rise to compact and simple rules. Table 1 represents grammar example for a SD task with two features (X_1, X_2), five LLs per feature (LL_1^1, LL_1^2, …, LL_1^5, LL_2^1, …, LL_2^5) and two values for the target variable (Tv_1, Tv_2) where the symbol $?a$ in some of the production rules of the grammar represents one, and only one, of the values separated by commas in the square brackets. This algorithm employs uniform partitions with triangular membership functions as shown in Fig. 1.

The fuzzy sets (LL^1,LL^2,$\cdots$,LL^n) are defined by means of an uniform partition. On the other hand, FuGePSD employs the genetic cooperative-competition approach [24] where rules of the population cooperate and compete between them in order to obtain the optimal solution. It is also important to remark that individuals are of variable length just like population with a variable number of individuals throughout the evolutionary process. In this way, FuGePSD is able to obtain rules with different number of variables in the antecedent part of the rule associated to the complexity of the subgroup to describe.

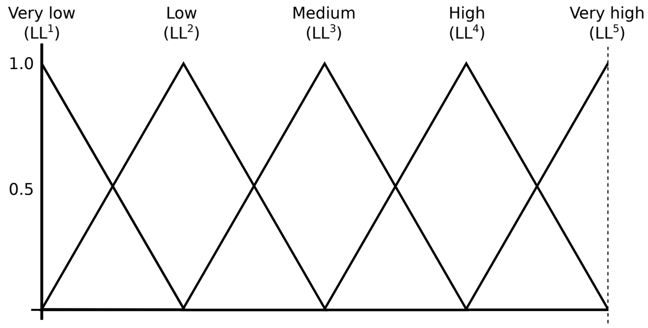

Figure 1: Example of fuzzy partition for a continuous variable with five linguistic labels

Table 1: Grammar example

Start $\longrightarrow$ [If], antec, [then], target_variable, [$\cdot$]
antec $\longrightarrow$ descriptor1, [and], descriptor2
descriptor1 $\longrightarrow$ [any]
descriptor1 $\longrightarrow$ [X_1] is label
descriptor2 $\longrightarrow$ [any]
descriptor2 $\longrightarrow$ [X_2] is label.
label $\longrightarrow$ member(?a,[LL^1,LL^2,LL^3,LL^4,LL^5]), [?a]
target_variable $\longrightarrow$ [Target_value is] descriptor
descriptor $\longrightarrow$ member(?a,[Tv_1,Tv_2]),[?a]

The algorithm starts with the generation of a main population ($MainPop$) which is evaluated and adapted throughout the evolutionary process by means of different operators through the generation, and the offspring population ($OffSpingPop$) is generated and evaluated. The offspring population has the same size as the main one.

Different genetic operators are applied to generate the offspring population from the main population, so that all the individuals generated in the offspring population are a modification of one individual of the main population. All the new individuals obtained must meet with the rules of the context-free grammar, which is defined in Table 1. Through a probabilistic way a child individual is generated with one of the following operators:

- Crossover: It is a genetic operator that combines two individuals. The new individual may be better than both of the parents if it heritages part of the properties from each of them. It is necessary to select two individuals in order to apply this operator. A component of the first parent is randomly selected and exchanged with another part of the second parent (also randomly selected), but under the constraint that the offspring produced must be valid according to the defined grammar. Only one of the two children is returned as a descendant.

- Mutation: This operator is used to improve the diversity of the offspring population. This operator alters one variable (selected in a random manner) of the individual by selecting a value, different of the original one, and considering the rules of the grammar.
- Insertion: The insertion of variables in an individual aims to include more precise rules in the model, so improving the precision and confidence of the set of rules obtained. This operator inserts a new variable in the individual whose value is generated in a random manner. It is not applied if the individual already has all the variables initialised according to the grammar.
- Dropping: The algorithm randomly selects a variable of the individual to be removed. This variable is no longer considered in the rule, so obtaining a more general one. This operator is not applied if the individual hence has only one variable following with the grammar of the algorithm.

Both populations (main and offspring) are joined in a new population (*JoinPop*). Due to the use of a cooperative-competitive approach, both individuals and population of the main population are evaluated in a separate manner, since it is necessary to evaluate the individuals and populations through two independent fitness functions. Hence, individuals compete between themselves with respect to a local fitness, and cooperate in order to obtain a population which is more adapted to the problem. Both functions will be referred to as fitness function and global fitness, respectively.

- Fitness function: This is calculated through the unusualness (Eq. 1). The use of a single quality measure as objective in the evolutionary process usually allows the algorithm to choose and select the individuals with the best values in this quality measure during the evolutionary process.
- Global fitness: This is estimated through an adaptation score in order to obtain the best population during the whole evolutionary process. In this way, it is necessary to calculate the accuracy of the set of rules using the normalised sum of the predictions for each rule. The global fitness is defined as follows:

$$GlobalFitness = \qquad (5)$$

$$\frac{w_1 * AAR + w_2 * (1 - n_v) + w_3 * (1 - n_R)}{w_1 + w_2 + w_3}$$

where n_R is the average number of rules of the population, n_v the number of descriptive variables, and AAR [25] the mean value of the accuracy for each single value of the target variable (calculated as):

$$AAR = \frac{1}{n_{tv}} * \sum_{i=1}^{n_{tv}} TPrate_i \qquad (6)$$

The global fitness employs different weights (w_1, w_2 and w_3) in order to give a 'trade-off' between accuracy and interpretability. The most suited values are 0.7 out of 1 for w_1, and the remaining 0.3 out of 1 for w_2 and w_3, because the main idea is to obtain a precise model with a low number of rules and a low number of variables for the set of rules. The use of 1-n_v and 1-n_R gives rise to an excessive number of rules and variables being penalised in the population score.

Finally, the token competition operator [26, 27] is applied in order to improve the diversity among the individuals at phenotype level, emulating the behaviour of a natural environment. Individuals with good niches will attempt to exploit it exclusively, and prevent more individuals to share its resources, unless a newer one is stronger than that initially developed. Therefore, the other individuals are required to seek their own niches. These properties provide diversity in the population to the algorithm. Moreover, this operator reduces the number of rules because all individuals without tokens will be deleted.

This operator orders the individuals of the population from highest to lowest fitness. Next, the individual with the highest fitness will exploit its niches by seizing as many tokens as it can. The other individuals entering the same niches will have their strength decreased, since they cannot compete with the stronger ones. This is achieved by introducing a penalisation to the fitness score of each individual, a limit which is based on the number of tokens which each individual has seized:

$$PenalizedFitness(R_i) = \qquad (7)$$

$$unusualness(R_i) * \frac{count(R_i)}{ideal(R_i)}$$

where, $count(R_i)$ is the number of tokens of the rule actually seized, and $ideal(R_i)$ is the total number of tokens that it can seize which is equivalent to the number of examples that the rule matches. If one rule seizes zero tokens, its fitness is modified to zero directly. On termination of the application of this mechanism, the size of the population is reduced with the individuals, where *PenalizedFitness* is greater than zero.

This evolutionary process is controlled through the number of generations and at the end of this process, the algorithm performs a screening function on the best population (*BestPop*) of the complete evolutionary process in order to obtain rules only with values greater than a threshold of sensitivity and confidence. In general, these thresholds should be configured above 60% because subgroups obtained must be precise and general and both quality measures are ideal to meet these objectives. This function is able to control, through an external parameter, the necessity of obtaining rules for all the values of the target variable, only obtaining the best rules. If the final rule set is empty, FuGePSD returns the best rule with respect to confidence. In this way, the algorithm always obtains rules.

Algorithm 1 Operation pseudo code for the FuGePSD algorithm

```
Output
$RuleSet$
Begin
Generate $MainPop$
Evaluate $MainPop$
$BestPop \longleftarrow MainPop$
repeat
  Generate $OffspringPop$ through $GeneticOperators$
  Evaluate $OffspringPop$
  Join $MainPop$ and $OffspringPop$ in $JoinPop$
  $MainPop \longleftarrow TokenCompetition(JoinPop)$
  if $MainPop.Fitness > BestPop.Fitness$ then
    $BestPop \longleftarrow MainPop$
  end if
until Number of generations is reached
$RuleSet = ScreeningFunction(BestPop)$
End
```

4. Experimental Study

The experimental study has been performed with the following datasets obtained from the UCI Repository of machine learning databases [28]: Appendicitis, Australian, Balance, Bridges, Cleveland, Diabetes, Echo, German, Glass, Haberman, Heart, Hepatitis, Ionosphere, Iris, Led, Vehicle and Wine[1]. It is important to remark that all datasets used in this experimental study contain at least one continuous variable.

To analyse the interest of the FuGePSD algorithm, the study compares the results with those of the remaining EFSs algorithms for SD (SDIGA, MESDIF and NMEEF-SD). The estimation of the quality measures (unusualness, sensitivity and fuzzy confidence) are obtained through a 10-fold cross-validation. In addition, according to the fact that all algorithms are stochastic, three executions are performed, and an average result from 30 values is shown for each dataset. It is important to highlight that values of a set of rules in unusualness, sensitivity and fuzzy confidence are computed as the average for all the rules in the set.

As we have previously mentioned, all the algorithms employ fuzzy confidence. This quality measure was defined in [6] and it is an adaptation of the standard confidence which measures precision within the domain [0, 1] through the antecedent part compatibility, which is the degree of compatibility between an example and the antecedent component of a fuzzy rule, i.e., the degree of membership for the example to the fuzzy subspace delimited by the antecedent part of the rule.

Average results of the algorithms are shown in Table 2, where name of the algorithms are abbreviated to NM (NMEEF-SD), SD (SDIGA), ME (MESDIF) and Fu (FuGePSD).

FuGePSD obtains the best values in average for all the quality measures analysed. However, in order to to complete the analysis, statistical tests are needed to check for significant differences between both proposals. In this way, the Wilcoxon statisical test [29] is applied to establish a ranking between FuGePSD and NMEEFSD. Table 4 represents the results of this test where R^+ corresponds to the sum of the ranks for algorithn FuGePSD and R^- corresponds to the ranks of algorithm NMEEFSD.

An analysis for each quality measure is performed below:

- Unusualness measures the novelty and interest of the subgroups obtained. In this way, FuGePSD gets the significance differences and the best results in 11 out of 17 datasets. In addition, in some datasets such as Iris, Ionosphere or Wine, the FuGePSD obtains results very close to the maximum possible, which is a good indicator of the quality of this algorithm in order to obtain interesting and interpretable rules.
- Sensitivity measures generality and precision in subgroups. As can be observed in the results, although FuGePSD obtains the best results only in 3 out of 17 datasets, it obtains the best average results in the experimental study. In this way, FuGePSD obtains a good level of sensitivity in a homogeneous manner throughout the experimental study in opposite to the remaining algorithms. The average result in this quality measure is higher than 80%, which show the general character of this algorithm.
- Fuzzy confidence measures the precision of the subgroups obtained. In this quality measure, FuGePSD also obtains significance differences with a value very close to 90% of precision in the subgroups obtained, also obtaining the best results in 11 out of 17 datasets.

Considering the properties of the SD task and the guidelines about SD presented in [16], the key factors for an SD approach are the obtaining of interesting and simple subgroups, covering the majority

[1] Complete descriptions of these datasets are outlined in http://archive.ics.uci.edu/ml/

	UNUSUALNESS				SENSITIVITY				FUZZY CONFIDENCE			
Dataset	NM	SD	ME	Fu	NM	SD	ME	Fu	NM	SD	ME	Fu
Appendicitis	0.094	**0.097**	0.072	0.045	**1.000**	0.976	0.950	0.926	0.904	0.710	0.541	**0.953**
Australian	**0.179**	0.053	0.101	0.162	0.799	**0.874**	0.867	0.800	**0.932**	0.697	0.700	0.888
Balance	0.071	0.066	0.045	**0.081**	0.526	0.584	0.421	**0.630**	**0.698**	0.582	0.528	0.673
Bridges	0.045	0.032	0.025	**0.052**	**0.879**	0.708	0.736	0.723	0.914	0.735	0.638	**0.948**
Cleveland	**0.130**	0.016	0.033	0.115	0.776	0.358	0.738	**0.770**	**0.823**	0.744	0.281	0.814
Diabetes	**0.086**	0.065	0.037	0.082	0.918	**0.981**	0.877	0.704	0.790	0.623	0.600	**0.871**
Echo	0.039	0.033	0.028	**0.068**	0.858	**0.983**	0.550	0.809	0.761	0.571	0.623	**0.837**
German	**0.067**	0.002	0.031	0.032	0.485	**0.841**	0.758	0.791	**0.877**	0.568	0.620	0.743
Glass	0.082	0.018	0.019	**0.099**	0.808	**0.919**	0.690	0.893	0.865	0.550	0.310	**0.972**
Haberman	**0.050**	0.042	0.012	0.039	**0.933**	0.837	0.854	0.919	**0.803**	0.629	0.561	0.797
Heart	0.107	0.059	0.075	**0.132**	0.718	**0.971**	0.702	0.744	0.765	0.578	0.656	**0.873**
Hepatitis	0.065	0.035	0.042	**0.077**	**0.806**	0.585	0.773	0.743	0.880	0.827	0.587	**0.950**
Ionosphere	0.141	0.029	0.096	**0.189**	**0.970**	0.811	0.848	0.952	0.866	0.556	0.669	**0.968**
Iris	0.207	0.169	0.191	**0.210**	**1.000**	0.990	1.000	0.987	**0.991**	0.899	0.869	0.983
Led	0.066	0.059	0.050	**0.077**	0.801	0.817	**0.855**	**0.855**	0.649	0.495	0.328	**0.826**
Vehicle	0.000	0.024	0.090	**0.093**	0.000	0.596	**0.983**	0.708	0.000	0.307	0.350	**0.790**
Wine	0.145	0.086	0.104	**0.183**	**0.919**	0.907	0.906	0.882	0.887	0.893	0.664	**0.982**
AVERAGE	0.093	0.052	0.062	**0.102**	0.776	0.808	0.795	**0.814**	0.789	0.645	0.560	**0.875**

Table 2: Average results for the different quality measure

Comparison	Quality measure	R^+	R^-	p-value	Hypothesis
FuGePSD Vs. NMEEF-SD	UNUSUALNESS	118	35	0.049	**Rejected by FuGePSD**
	SENSITIVITY	97	56	0.332	Non-rejected
	FUZZY CONFIDENCE	108	28	0.039	**Rejected by FuGePSD**

Table 3: Results of the Wilcoxon test between NMEEFSD and FuGePSD

of the examples of the target variable in a precise way. In this way, FuGePSD obtains:

- novelty subgroups because values obtained in unusualness are relevant, so it provides the experts with information to describe unusual and interesting behaviour within the data; and
- the best relation between sensitivity and confidence, as it obtains results with a good precision where the majority examples covered belong to the target variable. It is important to remark this factor because it is very difficult to obtain this compromise due to the frequent loss in a measure when trying to increase the other.

5. Conclusions

In this work, we have presented an EFS for SD based on genetic programming. Throughout the literature, EFSs have shown an excellent behaviour to cope with SD problems. In fact, such systems have been successful in several experimental studies presented in recent times.

FuGePSD algorithm is based on genetic programming, which together with fuzzy logic provides interpretability to the rules extracted. This is because the use of a knowledge representation close to the expert. In addition, fuzzy logic also avoids the necessity to make a previous discretisation. Moreover, FuGePSD obtains a compact and diverse rule set through the use of the token competition operator. This operator promotes the evolution of different individuals, i.e. this operator forces individuals to seek their own niches in the search space, so extending the diversity.

A wide experimental study focused on datasets with continuous variables shows that FuGePSD obtains a good trade-off between novelty and precision.

Specifically, this study shows that FuGePSD improves the results of the rest of the EFSs for SD presented up to the moment. Fuzzy subgroups obtained by FuGePSD are more accurate and cover more examples. These subgroups considered separately demonstrate not only improved relationships between sensitivity and confidence, but excellent results in unusualness which is a key quality measure in SD.

Acknowledgement

This work was partially supported by the Spanish Ministry of Economy and Competitiveness under project TIN2012-33856.

References

[1] W. Kloesgen. Explora: A Multipattern and Multistrategy Discovery Assistant. In *Advances in Knowledge Discovery and Data Mining*, pages 249–271. American Association for Artificial Intelligence, 1996.

[2] S. Wrobel. An Algorithm for Multi-relational Discovery of Subgroups. In *Proceedings of the 1st European Symposium on Principles of Data Mining and Knowledge Discovery*, volume 1263 of *LNAI*, pages 78–87. Springer, 1997.

[3] N. Lavrac, B. Cestnik, D. Gamberger, and P. A. Flach. Decision Support Through Subgroup Discovery: Three Case Studies and the Lessons Learned. *Machine Learning*, 57(1-2):115–143, 2004.

[4] B. Kavsek and N. Lavrac. APRIORI-SD: Adapting association rule learning to subgroup discovery. *Applied Artificial Intelligence*, 20:543–583, 2006.

[5] M. Atzmueller and F. Puppe. SD-Map - A Fast Algorithm for Exhaustive Subgroup Discovery. In *Proceedings of the 17th European Conference on Machine Learning and 10th European Conference on Principles and Practice of Knowledge Discovery in Databases*, volume 4213 of *LNCS*, pages 6–17. Springer, 2006.

[6] M. J. del Jesus, P. González, F. Herrera, and M. Mesonero. Evolutionary Fuzzy Rule Induction Process for Subgroup Discovery: A case study in marketing. *IEEE Transactions on Fuzzy Systems*, 15(4):578–592, 2007.

[7] M. J. del Jesus, P. González, and F. Herrera. *Fuzzy Sets and Their Extensions: Representation, Aggregation and Models*, volume 220, chapter Subgroup Discovery with Linguistic Rules, pages 411–430. Springer, 2007.

[8] C. J. Carmona, P. González, M. J. del Jesus, and F. Herrera. NMEEF-SD: Nondominated Multi-objective Evolutionary algorithm for Extracting Fuzzy rules in Subgroup Discovery. *IEEE Transactions on Fuzzy Systems*, 18(5):958–970, 2010.

[9] L. A. Zadeh. The concept of a linguistic variable and its applications to approximate reasoning. Parts I, II, III. *Information Science*, 8-9:199–249,301–357,43–80, 1975.

[10] A. E. Eiben and J. E. Smith. *Introduction to evolutionary computation.* Springer, 2003.

[11] C.J. Carmona, V. Ruiz-Rodado, M.J. del Jesus, A. Weber, M. Grootveld, P. González, and D. Elizondo. A fuzzy genetic programming-based algorithm for subgroup discovery and the application to one problem of pathogenesis of acute sore throat conditions in humans. *Information Sciences*, 298:180–197, 2015.

[12] J. R. Koza. *Genetic Programming: On the Programming of computers by Means of Natural Selection.* MIT Press, 1992.

[13] D. Gamberger and N. Lavrac. Expert-Guided Subgroup Discovery: Methodology and Application. *Journal Artificial Intelligence Research*, 17:501–527, 2002.

[14] F. Herrera, C. J. Carmona, P. González, and M. J. del Jesus. An overview on Subgroup Discovery: Foundations and Applications. *Knowledge and Information Systems*, 29(3):495–525, 2011.

[15] M. Atzmueller, F. Puppe, and H. P. Buscher. Towards Knowledge-Intensive Subgroup Discovery. In *Proceedings of the Lernen - Wissensentdeckung - Adaptivität - Fachgruppe Maschinelles Lernen*, pages 111–117, 2004.

[16] C.J. Carmona, P. González, M.J. del Jesus, and F. Herrera. Overview on evolutionary subgroup discovery: analysis of the suitability and potential of the search performed by evolutionary algorithms. *WIREs Data Mining and Knowledge Discovery*, 4(2):87–103, 2014.

[17] N. Lavrac, P. A. Flach, and B. Zupan. Rule Evaluation Measures: A Unifying View. In *Proceedings of the 9th International Workshop on Inductive Logic Programming*, volume 1634 of *LNCS*, pages 174–185. Springer, 1999.

[18] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A.I. Verkamo. Fast discovery of association rules. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and data mining*, pages 307–328. AAAI Press, 1996.

[19] F. Herrera. Genetic fuzzy systems: taxomony, current research trends and prospects. *Evolutionary Intelligence*, 1:27–46, 2008.

[20] G. Venturini. SIA: A Supervised Inductive Algorithm with Genetic Search for Learning Attributes based Concepts. In *Proceedings European Conference on Machine Learning*, volume 667 of *LNAI*, pages 280–296. Springer, 1993.

[21] M. J. del Jesus, P. González, and F. Herrera. Multiobjective Genetic Algorithm for Extracting Subgroup Discovery Fuzzy Rules. In *Proceedings of the IEEE Symposium on Computational Intelligence in Multicriteria Decision Making*, pages 50–57. IEEE Press, 2007.

[22] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In *International Congress on Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, pages 95–100, 2002.

[23] K. Deb, A. Pratap, S. Agrawal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions Evolutionary Computation*, 6(2):182–197, 2002.

[24] D. P. Greene and S. F. Smith. Competition-based induction of decision models from examples. *Machine Learning*, 13(2-3):229–257, 1993.

[25] C. Ferri, J. Hernández-Orallo, and R. Modroiu. An experimental comparison of performance measures for classification. *Pattern Recognition Letters*, 30(1):27–38, 2009.

[26] K. S. Leung, Y. Leung, L. So, and K. F. Yam. Rule Learning in Expert Systems Using Genetic Algorithm: 1, Concepts. In K. Jizuka, editor, *Proc. of the 2nd International Conference*

on Fuzzy Logic and Neural Networks, pages 201–204, 1992.

[27] M. L. Wong and K. S. Leung. *Data Mining using Grammar Based Genetic Programming and Applications*. Kluwer Academics Publishers, 2000.

[28] A. Asuncion and D. J. Newman. UCI Machine Learning Repository, 2007.

[29] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics*, 1:80–83, 1945.