Overview

Overview on evolutionary subgroup discovery: analysis of the suitability and potential of the search performed by evolutionary algorithms

Cristóbal J. Carmona,[1]* Pedro González,[1] María José del Jesus[1] and Francisco Herrera[2,3]

Subgroup discovery (SD) is a descriptive data mining technique using supervised learning. In this article, we review the use of evolutionary algorithms (EAs) for SD. In particular, we will focus on the suitability and potential of the search performed by EAs in the development of SD algorithms. Future directions in the use of EAs for SD are also presented in order to show the advantages and benefits that this search strategy contribute to this task. © 2014 John Wiley & Sons, Ltd.

How to cite this article:
WIREs Data Mining Knowl Discov 2014. doi: 10.1002/widm.1118

INTRODUCTION

The main objective of subgroup discovery (SD), a descriptive data mining technique based on supervised learning,[1] is the discovery of interesting relationships with respect to a specific property of interest to the user, the target variable. SD techniques have been used to address several problems throughout the literature, among which can be highlighted the results in domains such as medicine[2–6] and bioinformatics.[7–10] The main advantage of SD with respect to other techniques in real problems is the obtaining of simple models easily interpretable by experts.

Evolutionary algorithms (EAs)[11] imitate the principles of natural evolution in order to solve search/optimization and learning problems. EAs are very suitable to perform the SD task because they can reflect well the interaction of variables in rule-learning processes and provide wide flexibility in the representation.[12,13] In addition, in the rule extraction process of SD, it is necessary to optimize simultaneously different measures so the use of multiobjective EAs (MOEAs) is well suited.

This article has two objectives. On the one hand it presents a state of the art of the use of EAs in SD. On the other hand the article analyses the suitability and potential of the search performed by EAs in order to apply to the development of SD algorithms. The main goodnesses and advantages of EAs are stated, the lessons learned on the use of this type of search strategy are summarized, and future directions to consider in the use of EAs in SD are discussed.

To do so, the article is organized as follows. First, the positioning of SD in data mining is presented where definition, description language and quality measures for SD are shown in different subsections. Then, the use of EAs in SD can be observed. This section presents the main advantages in the use of EAs for SD task and main features of this type of algorithms presented throughout the literature within SD. In addition, we present a taxonomy proposal for EAs for SD and describe the algorithms presented. Finally, the suitability and potential of the search performed by EAs in the SD task and the conclusions are depicted.

*Correspondence to: ccarmona@ujaen.es

[1]Department of Computer Science, University of Jaen, Jaen, Spain

[2]Department of Computer Science and Artificial Intelligence, University of Granada, Granada, Spain

[3]Faculty of Computing and Information Technology – North Jeddah, King Abdulaziz University, Jeddah, Saudi Arabia

Conflict of interest: The authors have declared no conflicts of interest for this article.

SUBGROUP DISCOVERY

SD extracts descriptive knowledge from data concerning a property of interest. SD together with Contrast Set Mining[14] and Emerging Pattern Mining[15] were grouped within the concept of Supervised Descriptive Rule Induction[16] whose main aim is to understand the underlying phenomena with respect to an objective class. Below is formally described the SD task and two key elements in any SD algorithm, how the extracted knowledge is represented and evaluated (both during and outside the learning process).

Definition

The concept of SD was initially introduced by Kloesgen[17] and Wrobel,[18] and more formally defined by Siebes[19] but using the name Data Surveying for the discovery of interesting subgroups. It can be defined as[20]:

> *'In subgroup discovery, we assume we are given a so-called population of individuals (objects, customers, . . .) and a property of those individuals we are interested in. The task of subgroup discovery is then to discover the subgroups of the population that are statistically "most interesting", i.e., are as large as possible and have the most unusual statistical (distributional) characteristics with respect to the property of interest.'*

The main purpose of SD is the search for relations between different properties or variables with respect to a target variable.

To represent the knowledge, SD employs rules which consist of induced subgroup descriptions. 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 $Target_{value}$.

SD uses descriptive induction through supervised learning which is widely used in classification. However, SD is differentiated with respect to classification techniques because it attempts to describe knowledge by data while a classifier attempts to predict the target value for new data to incorporate in the model. Furthermore, an SD algorithm obtains a simple model composed of individual and interpretable rules, while the classifier is composed of a set of dependant rules which are analyzed overall with respect to accuracy.

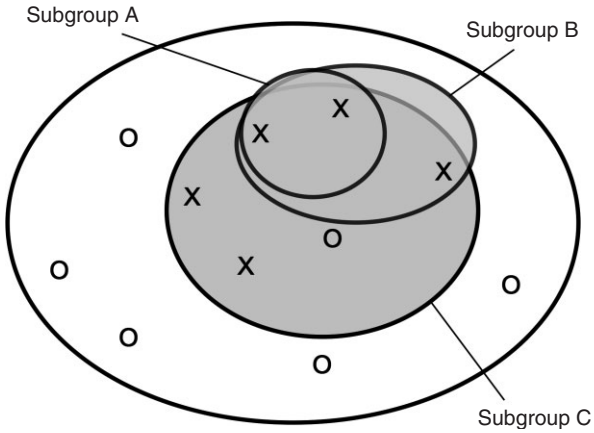

FIGURE 1 | Representation of different subgroups in an example problem.

Figure 1 represents different subgroups in an example problem. As can be observed, *Subgroup A* represents a very specific subgroup because it covers only two examples of the target variable *x*, *Subgroup B* covers one example more. Both subgroups have a high quality since all the examples covered are correctly described. On the other hand, *Subgroup C* is more general because it covers all the examples of the target value *x*. However, *Subgroup C* covers one example of the target value *o* leading to decrement the quality of the subgroup. In summary, different types of subgroups (general and/or precise) can be obtained in SD as also occurs in classification. However, in SD the extraction of general subgroups with good values of precision is really interesting.

The main elements in SD in order to analyze a proposed SD algorithm or to develop a new one are the following[21]: the type of the target variable, the search strategy, the description language and the quality measures. Different types of target variable can be found like binary, nominal or numeric, for example. With respect to the search strategy used there are in the literature different proposals based on beam search like CN2-SD[22] or Apriori-SD,[23] exhaustive search like SD-Map[24] or Merge-SD[25] for example or EAs among others. The proposals based on EAs as search strategy are main focus of this study being analyzed in a specific section. The latter two elements, description language and quality measures are described in the following subsections.

Description Language for SD Rules

Interpretability of the extracted knowledge is an important issue in data mining which determines the quality of both the algorithm used and the results. This is particularly true in SD. In this sense, the knowledge

description language used is one of the most influential factors on the interpretability when designing an SD algorithm.

In SD, different sorts of rules have been used for knowledge representation:

- Boolean rules. Different sorts of Boolean rules have been presented in the literature. They can be classified considering the representation of descriptive variables:
- Pairs of variable/value. This type of rules is used by the algorithms SDIGA,[26] MESDIF[27] and NMEEF-SD.[28] Below, a rule of this type can be observed:

$$R: IF\ X_1 = 3\ \ AND\ \ X_2 = Spain\ \ THEN\ \ Target_{value} \quad (1)$$

- Pairs of variable/interval. These rules are employed by EDER-SD[29] and GP3-SD,[30] for example. Next, an instance of this kind of rules is presented:

$$R: IF\ X_1 = [1,3]\ \ AND\ X_2 = Spain\ \ THEN\ \ Target_{value} \quad (2)$$

- Pairs of variable/value with order relations. Rules with pairs variable/value and order operators. It is used by classical algorithms like CN2-SD[22] and SD.[3] A rule of this type can be observed below:

$$R: IF\ X_1 > 1\ \ AND\ \ X_2 \neq Spain\ \ THEN\ \ Target_{value} \quad (3)$$

- Fuzzy rules. This type of rules, based on fuzzy logic[31] allows to consider uncertainty and to represent the continuous variables in a way close to human reasoning. In this way, fuzzy linguistic labels (LLs) are considered as the values of the continuous variables. The fuzzy set corresponding to each LL can be specified by the user or defined by means of an uniform partition, if the expert knowledge is not available.

Both crisp and fuzzy rules can have canonical representation—a conjunction of variable-value pairs with order relations, equalities or intervals—or the disjunctive normal form (DNF) representation[32] allowing the use of more than one value for each variable. Eq. (4) represents an example of canonical crisp rule and Eq. (5) an instance for DNF fuzzy rule:

$$R: IF\ X_1 = 3\ \ AND\ \ X_2 = Spain\ \ THEN\ \ Target_{value} \quad (4)$$

$$R: IF\ X_1 = \left(LL_1^1\ OR\ LL_1^2\right)\ \ AND\ \ X_6 = \left(LL_6^2\right)\ \ THEN\ \ Target_{value} \quad (5)$$

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. These variables can be categorical or numerical.
- $T = \{Target_{value} / j = 1, \ldots, n_c\}$ is a set of values for the target variable and n_c is the number of values.
- $LL_{n_v}^{l_{n_v}}$ is the linguistic label number l_{n_v} of the variable n_v.

As can be observed, a canonical rule is formed by variable-value pairs (with only one value per variable) where all pairs are connected through the *AND* operator. Otherwise, DNF rules can include several values for each variable. In order to represent this property in the representation, the operator *OR* is employed.

Quality Measures Used in SD

A wide number of quality measures have been presented in the SD literature both to guide the search process in order to find the best SD rules and to measure the quality of the SD rule set finally obtained.[17,22] In addition, a complete description about these quality measures can be found in.[1] The most common quality measures used in SD can be classified by their main objective such as:

- Complexity measures, related to the interpretability of the subgroups, i.e., to the simplicity of the knowledge extracted.
- Generality measures, used to quantify the quality of individual rules according to the individual patterns of interest covered.
- Precision measures, showing the precision of the subgroups.

TABLE 1 | Classification of the Quality Measures Used in Subgroup Discovery

Quality measure	C	G	P	I
Number of rules	X			
Number of variables	X			
Coverage[22]		X		
Support[22]		X		
Confidence[33]			X	
Precision measure Q_c[3]			X	
Precision measure Q_g[17]			X	
Q_g-Weight[3]			X	
Interest[34]				X
Novelty[18]				X
Lift[35]				X
Significance[17]				X
Sensitivity[17]		X	X	
False alarm[3]		X	X	
Specificity[17]		X	X	
Unusualness[36]		X	X	X
Piatetstky-Shapiro[37]		X	X	X

C, complexity; G, generality; I, interest; P, precision.

- Interest measures, intended for selecting and ranking patterns according to their potential interest to the user.
- Hybrid, that attempt to obtain a good trade-off between different objectives.

Table 1 summarizes the *Quality measures* most used in SD[1] and their main characteristics.

According to the SD concept the obtaining of interesting, simple and interpretable subgroups, covering the majority of the examples of the interest property (target variable) is desirable. Considering this definition and the analysis of the different quality measures used in the literature, we propose three guidelines in order to establish the type of measure more suitable, to guide the search process and to analyze the quality of the subgroups obtained by any SD algorithm:

- *Interpretability*. An SD proposal must obtain few rules containing a low number of variables in the antecedent part in order to help to the experts to understand and use the extracted knowledge, i.e., simple and interpretable subgroups are preferred in SD task.
- *Relation sensitivity-confidence*. An SD algorithm must obtain results with a good precision, where the majority examples covered belong to the target variable, i.e., it must achieve the best possible relation between sensitivity and confidence. Both quality measures are primordial in order to provide subgroups to the experts that cover the higher number of described correctly examples. It is difficult for the algorithms to achieve this compromise due to the loss suffered by a measure when trying to increase the other.
- *Novelty*. An SD model must contribute novel knowledge, providing the experts with information in order to describe unusual and interesting behavior within the data. This objective could be measured with a wide number of quality measures as novelty, interest or significance, among others. Nevertheless, it is important to highlight the utility of the unusualness to measure this objective because it contributes with generality and confidence to the problem. Moreover, this quality measure is widely used in the specialized bibliography.

It can be considered that the main purpose of an SD algorithm is to find a good trade-off between these three guidelines because this lead to the obtaining of good results in a wide number of quality measures and not only in the ones used in the search process.

EVOLUTIONARY ALGORITHMS FOR SD

SD is a rule-learning process that can be seen as an approximation problem in which the objective is the learning of the parameters of the model. In this task, the search space can be very complex and the search strategy used becomes a key factor. The search in this complex search space can be performed using different strategies, such as bean search, exhaustive search or EAs. The use of EAs is very well suited because these algorithms perform a global search in the space while other systems based on decision trees divide the search space enabling the not detection of overlapping rules for the same class.

EAs are stochastic algorithms for optimizing and searching based on the natural evolution process. These algorithms were introduced by Holland.[38] Different computational paradigms can be found within EAs: genetic algorithms,[38,39] evolution strategies,[40] evolutionary programming[41] and genetic programming.[42] It is interesting to remark the use of the evolutionary rule-based systems[43] which use sets of rules as knowledge representation highlighting the use of this type of algorithms for efficient searches

over complex search spaces[44] because it has a great potential as a search tool, allowing the inclusion of domain knowledge and the obtaining of better rules. A comparison between evolutionary versus nonevolutionary methods was performed in[13] and the encouraging performance of the evolutionary rule-based systems in comparison with the nonevolutionary algorithms was shown. In addition, a review about the good use of genetic programming for classification can be observed in[12] where main advantages with respect to the facility in the representation in different problems, application in different task of data mining, adaptability to the problems and so on.

This section presents the main features of EAs and a taxonomy divided into the use of one and other EAs, allowing the aggregation of the objectives in a case and the use of different objectives in the other, and in the use of post-processing methods in order to improve the rules obtained are presented.

Main Features of EAs Used in SD

The knowledge is represented in SD through rules as we have previously mentioned. Each rule is divided into two parts, antecedent and consequent. In this way, EAs for rule-learning process can consider or not to represent the consequent within the rule, i.e., two codifications can be used: to codify the complete rule or codify only the antecedent. Moreover, different options can be taken when codifying the consequent because the value of the target variable could be fixed in a value or could be determined to the evolutionary process in order to cover most of the examples of the search subspace delimited by the antecedent.

According to this, EAs for rule-learning process can use different schemes of representation:

1. '*Chromosome = Rule*' approach, in which each individual codifies a single rule and the whole rule set is provided by combining several individuals in the population. Three categories are considered:

- The Michigan approach is usually called learning classifier systems.[45] This approach are rule-based, message-passing systems that employ reinforcement learning and an EA to learn rules that guide their performance in a given environment. New rules are detected through the EA and bad ones are replaced via competition in the evolutionary process.
- Iterative Rule-Learning (IRL) approach. In the evolutionary process the chromosomes compete where the best rule per run is chosen. The global solution is formed by the best rules obtained when the algorithm is run multiple times. SIA[46] is a proposal that follows this approach.
- The genetic cooperative-competitive learning (GCCL) approach,[47] in which the complete population or a subset of it encodes the rules. In this scheme the chromosomes compete and cooperate simultaneously.

2. '*Chromosome = Set of rules*' approach, also called the Pittsburgh approach, in which each individual represents a set of rules.[48] In this case, a chromosome evolves a complete set of rules and they compete among them along the evolutionary process.

On the other hand, the choice of the representation determines largely the operation of the algorithm and it is fundamental to solve the problem. Different representations for the codification of the chromosomes can be observed such as:

- Binary codification is based on bits chains where for each gene we can use different number of bits.
- Integer/Real codification where the variables of the problem are associated with only one gen which has the concrete value within the interval.
- Tree structure is traditionally employed in genetic programming paradigm.[12,42]

EAs are widely applied in knowledge discovery. In SD it is necessary to simultaneously optimize different measures, which are in conflict with each other. SD task is so a multiobjective optimization problem. EAs can address this problem in different ways. On the one hand, the monoobjective EAs[49] face the problem as the optimization of a single measure, usually aggregating the different measures into a single one. On the other hand, MOEAs[50,51] attempt to optimize the different measures at the same time.

Monoobjective Evolutionary SD

These algorithms are based on the utilization of EAs with a single objective as fitness function. In this way, the search process is oriented in only one direction for one objective or for the aggregation of different objectives in one expression.

Different approaches can be observed in the bibliography based on an evolutionary fuzzy system (SDIGA[26]), on a hierarchical decision rules approach (EDER-SD[29]), on an IRL approach (GAR-SD[52]) and

```
BEGIN
Set of rules is empty
repeat
  repeat
    Generate P(0)
    Evaluate P(0)
    repeat
      Include the best individual in P(nGen+1)
      Complete P(nGen+1): Crossover and Mutation of individuals from P(nGen)
      Evaluate P(nGen+1)
    until Number of evaluations is not reached
    Obtain the best rule R
    Local search R
    if Confidence(R) ≥ Minimum Confidence and R represents new examples then
      Set of rules ⋃ R
      Mark the set of examples covered by R
    end if
    Set of rules ⋃ R
  until Confidence(R) ≤ Minimum Confidence and R not represents new examples
until Not Target value
END
```

FIGURE 2 | Operation scheme of SDIGA algorithm.

based on genetic programming (GP3-SD[30]). These algorithms are presented below.

SDIGA.[26] It is an evolutionary fuzzy system[53] because it uses a knowledge representation fuzzy rules and evolutionary computation as a learning process. It is interesting to remark that SDIGA searches for rules for each value of the target variable, i.e., the consequent is not represented in the chromosome but is fixed.

This algorithm follows the IRL approach where the solution of each iteration is the best individual obtained and the global solution is formed by the best individuals obtained in the different runs. The representation of the individuals is performed through the '*Chromosome* = *Rule*' approach and the core of SDIGA is an EA using a post-processing step based on a local search. This hybrid algorithm extracts one simple and interpretable fuzzy rule with an adequate level of support and confidence. The algorithm model can use fuzzy canonical or DNF rules with a predefined set of LLs.

This algorithm is included in an iterative process for the extraction of different rules. In this way, algorithm marks examples cover for rules to prevent a new rule being obtained which covers exactly the same examples in the following runs. Algorithm produces rules while the generated rules reach a minimum level of confidence and give information on areas of the search space in which there are examples not described by the rules generated in previous iterations. The rule is improved in a post-processing phase throughout a hill-climbing process, which modifies the rule in order to increase the degree of support.

The fitness is an aggregation function where the selection of the quality measures like coverage, significance, unusualness, accuracy, sensitivity, crisp support, fuzzy support, confidence, and fuzzy confidence is determined by the user. The number of objectives within the weighted aggregation function is between 1 and 3.

SDIGA is implemented in the KEEL software tool[54,55] and its operation scheme can be observed in Figure 2. This algorithm has been applied in order to search for unusual relationships in different real-world problems such as:

- Marketing, which was analyzed in.[26] The main objective of this article was to extract conclusions from the information on previous trade fairs to determine the relationship between the trade fair planning variables and the success of the stand. SDIGA was applied in order to extract information of interest about each of the three efficiency groups of stands: low, medium, and high efficiency.
- Medicine, for the discovery and description of patients patterns in a psychiatric emergency department.[2] In this work were presented rules describing relationships between the different variables stored for each patient and the arrival

```
BEGIN
Set of rules is empty
while Instances for Target value are not covered do
   Generate P(0)
   Evaluate P(0)
   repeat
      Generate P(nGen+1): Select and Recombination from P(nGen)
      Evaluate P(nGen+1)
   until Number of generations is reached
   Obtain the best rule R
   Modify the weight from R
   Set of rules ⋃ R
end while
END
```

FIGURE 3 | Operation scheme of EDER-SD algorithm.

time divided in different periods: day, afternoon, and night.

- E-learning, which was analyzed in different works.[56,57] The main objective was to determine behavior patterns for the students in e-learning platforms at the University of Cordoba. Both articles analyze possible relations between the usage of complementary activities of a course and the final marks obtained by the students. The final mark is used as the variable to characterize, using the different marks to divide the data into classes and codifying them as values of the consequent of the rules.

EDER-SD.[29] This EA attempts to characterize the minority class of a problem. Specifically, the algorithm describes interesting relations on imbalanced data that generates rules for the target variable. This algorithm is a modification of the HIDER (HIerarchical DEcision Rules) algorithm.[58] It is a sequential covering EA that produces a hierarchical set of rules, i.e., an instance will be classified by the ith rule if it does not match the conditions of the $(i-1)$th precedent rules. The rules are sequentially obtained until the search space is totally covered.

EDER-SD generates only individuals for the target variable value corresponding to the minority class. In addition, this algorithm penalizes an example when it is covered by a rule, as CN2-SD[22] and Apriori-SD[23] do. This penalization instead of removing the examples produces the obtaining of more complex rules because they have more conditions, i.e., the precision of the rules is increased. Therefore, this algorithm generates pyramidal rules, where the first rule (with fewer conditions) covers a large number of instances (high support) but with low precision. The rest of the rules in the pyramid keep adding conditions decreasing the support but increasing the precision.

The representation of the individuals is performed through the '*Chromosome* = *Rule*' approach, where a tuple of real values represent the interval of the ith attribute. This algorithm, instead of SDIGA,[26] represents the target variable in the chromosome.

This algorithm is a monoobjective algorithm and it allows the use of a wide number of the metrics used in SD such as unusualness, accuracy, sensitivity, or significance among others. The operation scheme of the algorithm EDER-SD can be observed in Figure 3. The good behavior of this algorithm has been shown in a real-problem with several datasets in order to generate models for defect classification. Results obtained show that this algorithm generates understandable and useful models that can be used by project managers or quality assurance personnel to guide the testing effort and improve the quality of software development projects.

GAR-SD.[52] This algorithm is based on IRL approach and searches for rules for each value of the target variable. GAR-SD is able to operate with continuous variables (through the use of intervals) and discrete ones. The process of searching for different rules is initiated with the generation of the initial population according to examples in the dataset that have not been covered previously by individuals. In addition, it employs a predefined support indicating the minimum percentage of examples covered. With respect to the initialization, continuous attributes are generated with a reduced interval while discrete attributes start with a single value.

The representation of the individuals is performed through the '*Chromosome* = *Rule*' approach, where a tuple of real values represent both the interval of the ith continuous attribute and the value of discrete variables. These rules are codified through individuals with variable length in which the consequent is not included because it is fixed for each run.

```
BEGIN
Set of rules is empty
repeat
  while Instances not covered < Support do
    nGen ← 0
    generate the initial population P(0)
    repeat
      Evaluate P(nGen)
      Generate P(nGen+1): Select, Crossbreeding and Mutation from P(nGen+1)
    until Number of generations is reached
    Set of rules ⋃ Best rule from P(nGen)
    Adjust bounds of set of rules
  end while
until Not Target value
END
```

FIGURE 4 | Operation scheme of GAR-SD algorithm.

GAR-SD employs a weighted aggregation function with six objectives where there are three quality measures for SD, namely support, confidence, and significance. The remaining quality measures are related to the variables which form the rule as amplitude_C, amplitude_D, and re-covering. Moreover, the evolutionary process is guided by a modification of the uniform crossover operator where in the replication of the population the elitism concept is employed. In addition, the algorithm uses the mutation operator and finally an adjustment function of the intervals is presented.

The validity of the algorithm has been analyzed with benchmarks obtained from the UCI Repository of machine learning databases.[59]

GP3-SD[30] is an iterative algorithm that mines subgroups for each value of the target attribute. The main objective is to discover the best subgroups (induced as rules in the form of a derivation tree) for each target attribute. This approach is based on a genetic programming algorithm for mining association rules[60] where the individuals are represented by means of tree structures with antecedent and consequent.

One of the most important aspects of this algorithm is the selection procedure because it works as an elitist selection allowing some of the best individuals discovered and more representative from the current generation to carry over to the next, unaltered.

At the end of the iterative process, GP3-SD runs a final stage to increase or decrease the width of the intervals (in continuous domains) in order to discover a high quality intervals satisfying more instances or prompting to a more reliable rule.

It uses genetic standard operators for genetic programming like mutation and crossover. However,

```
BEGIN
Set of rules is empty
repeat
  Generate P(0)
  Evaluate P(0)
  repeat
    Copy best individuals in P(nGen+1)
    Generate P(nGen+1)
    Evaluate P(nGen+1)
    Update genetic operator probabilities
  until Number of generations is reached
until Not Target value
Obtain best individuals
Optimise numerical conditions
Set of rules ⋃ Best individuals from P(nGen)
END
```

FIGURE 5 | Operation scheme of GP3-SD algorithm.

there is an interesting feature with respect to these operators because the probability of applying them is adjusted based on the population requirements. With respect to the fitness function, this algorithm uses a weighted aggregation function considering support and confidence (Figure 4).

The operation scheme of this algorithm can be observed in Figure 5. At the end of the process, the algorithm optimizes numerical features (if they are present in the problem) codified through intervals with an increase or decrease of their width.

GP3-SD has been validated with benchmarks obtained from the UCI Repository of machine learning databases.[59]

Table 2 summarizes the most important elements used for each evolutionary approach.

TABLE 2 | Comparison of Properties for Monoobjective Evolutionary Algorithm for Subgroup Discovery

Feature	SDIGA[26]	EDER-SD[29]	GAR-SD[52]	GP3-SD[30]
Evolutionary algorithm	Generational with IRL approach	Generational with IRL approach	Generational with IRL approach	Generational with Elitism
Knowledge representation	Canonical or DNF fuzzy rules	Canonical rules with intervals	Canonical rules with intervals	Canonical rules with intervals
Individual coding	Chromosome = Rule	Chromosome = Rule	Chromosome = Rule	Tree structure = Rule
Quality measures used as objectives	Aggregation function with different measures among: Coverage, significance, unusualness, accuracy, sensitivity, support and confidence	Accuracy, sensitivity, significance, f-measure, lift or unusualness	Aggregation function with different measures: support, confidence, significance, amplitude, amplitude and re-covering	Aggregation function with different measures: support and confidence
Operators	Biased mutation, two point crossover and local search	Recombination and real-code crossover	Uniform crossover, mutation and adjustment of intervals	Standard crossover and mutation

Multiobjective Evolutionary SD

The simultaneous optimization of different objectives is desirable in SD. The improvement through an aggregation function induces a search only in a trade-off zone between all the quality measures used. Moreover, the analysis of weights used difficult this search process. However, a multiobjective optimization is more suitable for SD because the goal is to find the decision vectors, which correspond to objective vectors and can not be improved in a dimension without degrading one-another, which is called optimal Pareto front.[51]

Throughout the bibliography different multiobjective approaches for SD have been presented: MESDIF[27] and NMEEF-SD[28] based on evolutionary fuzzy systems. These approaches are presented, below:

MESDIF.[27] This MOEA is an evolutionary fuzzy system based on the SPEA2 approach.[61] It applies the concepts of elitism in the rule selection (using a secondary or elite population) and the search for optimal solutions in the Pareto front. In order to preserve the diversity at a phenotypic level the algorithm uses a niches technique which considers the proximity in values of the objectives and an additional objective based on novelty to promote rules which give information on examples not described by other rules of the population.

The rule induction process obtains rules with high predictive accuracy and which are comprehensible and interesting. In this proposal, the user can choose between a wide number of quality measures (coverage, significance, unusualness, accuracy, sensitivity, support, and confidence) to maximize all the defined objectives.

One of the most important aspects of MESDIF is the obtention of results for all the values of the target variable. It returns the individuals of the elite population for each value, whose size is defined by the user.

The algorithm uses the '*Chromosome = Rule*' approach. The MOEA discovers fuzzy rules whose consequent is prefixed to one of the possible values of the target feature. Therefore, all the individuals of the population are associated with the same value of the target variable, and so the chromosome only represents the antecedent of the rule.

MESDIF is implemented in KEEL[54,55] and its operation scheme can be observed in Figure 6. This algorithm was applied in real-world problems such as marketing,[62] medicine,[2] and e-learning.[56,57]

NMEEF-SD.[28] It is a multiobjective evolutionary fuzzy system based on NSGA-II.[63] NMEEF-SD codifies each candidate solution according to the '*Chromosome = Rule*' approach, where only the antecedent is represented in the chromosome and the consequent is prefixed to one of the possible values of the target feature in the evolution. Therefore, the algorithm must be executed as many times as the number of different values the target variable contains. With respect to the representation of the rules NMEEF-SD can use canonical or DNF rules.

As the general objective of NMEEF-SD is to obtain a set of rules, which should be general and accurate, the algorithm includes components which enhance these characteristics. In particular, diversity is enhanced in the population using a new operator to perform a re-initialization based on coverage,

```
BEGIN
Set of rules is empty
repeat
   Initialise counters
   Generate an initial population P(0) and create an empty elite population P'(0).
   repeat
      Calculate Fitness for P(nGen)
      Copy non-dominated individuals in P'(nGen)
      Generate P(nGen+1): Select, Crossover and Mutation from P(nGen)
   until Number of evaluations is reached
   Set of rules ⋃ Non-dominated individuals P(nGen)
until Not Target value
```

FIGURE 6 | Operation scheme of MESDIF algorithm.

```
BEGIN
Set of rules is empty
repeat
   Generate P(0)
   repeat
      Generate offspring Q(nGen) through operators in P(nGen)
      Join P(nGen) and Q(nGen) in R(nGen)
      Generate all non-dominated fronts from R(nGen)
      if the Pareto front evolves then
         Complete P(nGen+1) with fronts in order
      else
         Apply Re-initialisation based on coverage in P(nGen+1)
      end if
   until Number of evaluations is reached
   Set of rules ⋃ Pareto front P(nGen)
until Not Target value
END
```

FIGURE 7 | Operation scheme of NMEEF-SD algorithm.

in addition to a niching technique (the crowding distance in the selection operator). On the other hand, in order to promote generalization, as well as the objectives considered in the evolutionary approach, the algorithm includes operators of biased initialization and biased mutation. Finally, to ensure accuracy, in addition to the objectives NMEEF-SD returns as its final solution those rules which reach a predetermined confidence threshold.

NMEEF-SD allows to choose between two and three quality measures as objectives of the evolutionary process in order to obtain relevant subgroups, between: coverage, significance, unusualness, accuracy, sensitivity, support, and confidence. It is also implemented in KEEL[54,55] and the operation scheme can be observed in Figure 7. Recent applications to real problems with an EA for SD have been analyzed through this algorithm in different problems such as:

- E-learning,[57] where a description of possible relationships between the use of the e-learning platform and marks obtained by the students were analyzed. NMEEF-SD obtained a comprehensive set of subgroups employing a low number of variables with the highest unusualness. Subgroups obtained allowed to the teachers to take decisions about course activities to improve the performance of their students.
- E-commerce,[64] where the main objective of this article was to analyze the usage of customers in a website based on the sell of olive oil in order to improve its design. Conclusions obtained have helped to the webmaster team to improve the design of the website.
- Bioinformatic,[7] which presents an interesting study in this domain in order to find interpretable knowledge in the Influenza A virus problem and describe unusual behavior in several subtypes of this virus. The results of NMEEF-SD offer the community a new point of view in the analysis of the Influenza A virus by its interpretability, which obtains simple rules to represent different subtypes of the virus.
- Concentrating photovoltaic technology was analyzed in.[65] This technology is an alternative to the conventional photovoltaic for electric generation. It produces electricity in a cheaper way by means of high efficiency multi-junction solar cells. The main objective was to describe the main external variables which improve the performance of the solar cells. The results confirmed some relationships between atmospheric variables and maximum power as well as new knowledge.

Post-processing Approaches for Optimizing Subgroups

In a data mining process, it is possible to improve the rules obtained with the adjustment of different parameters of the algorithm. For example, it is possible to select some rules, modify intervals, adjust fuzzy labels, improve rules, and so on. This is also true in SD, where it is possible to improve the results by analyzing the rules and performing different adjustments.

TABLE 3 | Comparison of Properties for Multiobjective Evolutionary Algorithm for Subgroup Discovery

Feature	MESDIF	NMEEF-SD
Evolutionary algorithm	MOEA based on SPEA2	MOEA based on NSGA-II
Knowledge representation	Canonical or DNF fuzzy rules	Canonical or DNF fuzzy rules
Individual coding	Chromosome = Rule	Chromosome = Rule
Quality measures used as objectives	Selected by the user among: Coverage, significance, unusualness, accuracy, sensitivity, support, and confidence	Selected by the user among: Coverage, significance, unusualness, accuracy, sensitivity, support, and confidence
Selection	Tournament	Tournament
Operators	Biased mutation and two point crossover	Biased mutation, two point crossover and re-initialization based on coverage

This section presents two post-processing approaches for optimizing subgroups. On the one hand, the improvement of subgroups obtained by any SD algorithm based on fuzzy logic is performed through the use of the 2-tuples linguistic representation.[66] This adjustment improves the extracted knowledge without modify the linguistic representation of the rules. On the other hand, subgroups are improved with an approach based on the search for exceptions within subgroups obtained by any SD algorithm. This approach combines initial subgroups with their exceptions in order to improve the knowledge extracted.

Genetic lateral tuning for SD

Some SD algorithms use fuzzy logic[31] to codify continuous variables which are considered linguistic, and the fuzzy sets corresponding to the LLs can be specified by the user or defined by means of a uniform partition for example if the expert knowledge is not available. Therefore, the definition of the optimal fuzzy sets is a complex process. In a post-processing stage the definition of this fuzzy sets can be improved, for example through a lateral displacement of the LLs which is known as the 2-tuples linguistic representation.[66]

This approach was presented in[67] and allows the lateral displacement of the labels considering only one parameter (slight displacements to the left/right of the original membership functions). This involves a simplification of the search space that eases the derivation of optimal models. Furthermore, this process of contextualizing the membership functions enables them to achieve a better covering degree while maintaining the original shapes, which results in accuracy improvements without a loss in the interpretability of the fuzzy labels. In the specialized literature, the 2-tuples representation has been used to tackle different problems as regression[68] or classification.[69,70]

The symbolic translation of a linguistic term is a number within the interval $[-0.5, 0.5)$ that expresses the domain of a label when it is moving between its two lateral labels. Let us consider a set of labels S representing a fuzzy partition. Formally, we have the pair, $(s_i,\alpha_i), s_i \in S, \alpha_i \in [-0.5, 0.5)$.

This approach employs the lateral tuning through the global tuning of the semantics. The tuning follows a major interpretability and it is suitable for SD task because this property is important for SD algorithms. It is applied to the level of linguistic partition, where the pair (s_i, α_i) takes the same tuning value in all the rules where it is considered. Therefore, it uses a real representation with two values for variable. The proposal uses the CHC algorithm[71] in order to design the learning method. It is a genetic algorithm that presents a good trade-off between exploration and exploitation, making it a good choice in problems with complex search spaces.

Multiobjective Evolutionary Fuzzy System for Detection of Exceptions in Subgroups

An exception is a rule with incorrectly described examples corresponding to a different value from the target variable of the subgroup. The main idea of this post-processing approach is to find exceptions within subgroups in order to provide an improvement in the precision of subgroups and new knowledge for experts. In this way, this approach is applied for each subgroup obtained previously for any SD algorithm. Then, a set of exceptions are obtained for each subgroup, and finally a modified subgroup formed by initial subgroups and their exceptions are shown.

The approach based on this methodology is called MEFES and was presented in.[65] It uses a MOEA and fuzzy logic to tackle the detection of exceptions in subgroups. It is able to work in fuzzy and/or crisp domains, obtaining modified subgroups which are formed by initial subgroups and their exceptions.

The objective of this post-processing approach is to find exceptions within each input subgroup—i.e., examples incorrectly described corresponding to a different value of the target variable—in order to provide

an improvement in the accuracy of the SD models, because errors made by the model in the description of examples are corrected, and new knowledge for the experts is provided, and new spaces in the data with unusual behavior are delimited and described.

MEFES uses an integer representation scheme with as many genes as variables contained in the dataset, without considering the target variable. It is able to works with categorical and/or continuous variables, depending on the problem. The codification is performed according to the '*Chromosome* = *Rule*' approach,[53] where only the antecedent is represented in the chromosome. The value of the target variable for the individuals is considered to be the opposite value of the initial subgroup, because the main idea is to search for exceptions within the subgroup.

The core of the algorithm is performed through a MOEA following the NSGA-II approach.[63] In this way, MEFES uses nondominated sort and crowding distance as operators of the algorithm and specific operators as oriented initialization, oriented mutation, and oriented re-initialization based on coverage. The search for exceptions within subgroups is performed through two quality measures as objective vectors of the algorithm: confidence and sensitivity.

This approach was also applied to the problem of concentrating photovoltaic technology.[65] The results were analyzed by experts in this technology and subgroups with exceptions presented new interesting and relevant knowledge because some rare cases were detected.

SUITABILITY AND POTENTIAL OF THE SEARCH PERFORMED BY EAS

Considering the distinguishing features of the SD task and the current state of EA-based SD algorithms, some important questions about the suitability and potential of this type of algorithms are highlighted. To do so, this section presents a complete analysis of the advantages in the use of EAs for SD organized in different sections. Firstly, the justification on the use of EAs in the SD task is presented, describing the main advantages and benefits of this type of algorithms. Next, lessons learned are shown respect to the use of EAs for SD. Finally, the challenges to be addressed in the future are depicted.

Why Using EAs in SD

The process of obtaining descriptive rules using supervised induction in SD must:

- discover relationships among descriptive variables and target one,
- represent this knowledge in a simple, understandable way using rules and
- optimize different and usually conflicting quality measures,

including all the available domain knowledge.

SD is a complex problem that can be formulated from the optimization point of view coding rules as a parameter structure and searching the parameters values that give us the optimum for a concrete fitness function. In this scenario the EAs are a very suitable tool in order to design an SD algorithm. They are global search techniques with the ability to explore a large search space for suitable solutions only requiring a performance measure. They have shown a good behavior in complex optimization problems.[72]

The assortment of SD rules includes rules with discrete variables, rules with inequality operators, rules with intervals, fuzzy rules (and combinations among them) and depending on the kind of rule to use, the configuration of the search space will be different. It is interesting to remark that EAs allow a wide flexibility in the representation. Likewise, the design of genetic operators allows to reflect in a suitable way the interaction among variables which is a key factor in the rule learning processes. Moreover the generic code structure and independent performance features of EAs make them suitable candidates to incorporate *a priori* knowledge. For SD rule learning this knowledge can be expressed as definitions or restrictions for intervals, fuzzy sets corresponding to linguistic variables, fuzzy membership function parameters, number of rules, or genetic operators that bias the search using knowledge related to SD task.

As previously mentioned, the main purpose of SD algorithms is to find a good trade-off between interpretability, relation between sensitivity-confidence and novelty. This complex purpose has been addressed with classical methods to handle multiobjective optimization problems, usually the weighted sum method that scales a set of objectives into a single objective by multiplying each one with a user-supplied weight describing his preference.[26,30] These approaches look for the best solution representing only one preference. As it is well known, EAs have the ability to find multiple optimal solutions in one single run. It makes EAs into a good approach to multiobjective optimization problems. MOEAs[50,51] can be taken into account multiple goals within the same optimization process. They generate a set of nondominated solutions that represent a trade-off among objectives instead of a single one. The use of MOEAs is a suitable option for the multiobjective optimization problem that SD task presents.

Sometimes, it is necessary to optimize previously obtained extracted knowledge in SD, such as simplify rule set, tune intervals, change fuzzy partitions used in variables or detect exceptions in rules among others. These problems can also be formulated as optimization problems approachable by EAs.

Lessons Learned

EAs are used as a global search technique. They are very suitable to explore large search spaces and to find adaptable solutions. The contribution of this type of algorithms to the SD task can be summarized in the following main properties:

1. Chromosome = Rule is probably the more natural representation for extracting rules in SD. It should be noted that the main objective of SD is to obtain rules to represent independent chucks of knowledge. An evolutionary search process where an individual is a rule allows to explore in an efficient and effective way the complex search space.
2. There are not Chromosome = Set of rules or Cooperative-Competitive approaches in the literature. This could be related to the individual character of the SD rules. Nevertheless, the development of proposals based on these approaches (in special cooperative-competitive for the consideration of a more limited search space) could be analyzed. This approach would allow to consider rules with overlapping between antecedents and different consequents in a suitable way. In addition, it could allow to obtain both rules and exceptions in a single stage.
3. Some proposals codify both antecedent and consequent in the search process. This codification allows to determine the value of the consequent in order to obtain rules so it gives a higher flexibility in the learning process. However, rules for all values of the target variable may not be obtained. In certain situations where experts need to obtain information only for some values of the target variable, it would be necessary to use algorithms that fix the consequent value. As a consequence, the use of a prefixed consequent in the learning process reduces the search space and could improve the learning process in an efficient and effective way. Definitely, codification of individuals is associated to the necessity of the experts.
4. There is no consensus about the most suitable type of rule for the SD task (interval, fuzzy, categorical, and DNF). The election usually relies on the particular preferences for the problem. In problems with a continuous domain (not previously discretized) the learning process is more complex but the EAs provide a suitable framework.

New Challenges on the Use of EAs for SD

There are some potential areas of future research regarding the use of EAs for SD. Some of them are the following:

1 *Overlapping and interaction among SD rules*. So far the cooperation between rules has not been taken into account in SD. Algorithms evaluate individually rules during the search process. At the end averages of the quality measures of the set of individual rules are obtained. Thus obtaining overlapping rules is not prevented or detected, except for some approaches that weights the examples already covered.

The analysis of overlapping rules in SD can be interesting in some cases:

- Two rules with the same consequent and overlapped antecedents could be considered as two different points of view to describe the value of the target variable. This can provide useful knowledge to the expert on the problem. The development of EAs for SD that considers this overlapping could be relevant.
- Rules with different consequents and overlapped antecedents mean overlapping between classes. If classes are overlapped it might be interesting to develop EAs to pay special attention to the description of the minority class. In the special case when an antecedent includes the other, we are talking about exceptions. There is a post-processing approach for the extraction of exceptions from SD rules,[65] but an approach addressing the learning of general rules and exceptions together would be interesting. Moreover, this situation is possible to analyze it from the point of view of the multilabel classification problem.
- The interaction among rules has not been considered yet in SD algorithms due to the individual evaluation of the SD rules. As the common use of SD algorithms is the obtaining of a set of simple rules that provide knowledge in a classification problem, it would be interesting

to develop SD algorithms considering the interaction among rules.

2. *SD algorithms for regression problems*. In regression problems the description of the relation among descriptive variables and one variable of interest (in this case, continuous) could be interesting for the expert. The EAs for SD presented throughout the literature work only with discrete/categorical target variables and there is only one nonevolutionary algorithm, SD-Map ⋆,[73] that addresses this problem.

The learning of SD rules for the description of regression problems must face a more complex search space in which the dependences among descriptive features and continuous target variable must by analyzed as well as the proper way to manage the target variable. Again, for this complex problem, the adaptability on the representation and operators and the robustness capacity of EAs make them a good choice to design SD algorithms for regression problems.

3. *Scalability issues in SD*. When applying SD to real-world problems, these usually are large or high dimensional datasets. The former is related to the number of instances and the later with the number of features. There are two typical possibilities to face it[74]: redesigning the algorithm to run efficiently with huge input datasets or reducing the size of the data before the application of SD algorithm without changing the result drastically. In the specialized bibliography different sampling and filter algorithms are proposed with the aim to reduce the number of instances before applying the SD algorithms.[75–77] Until now there are no studies that attempt to reduce the number of variables for SD task or to approach the large and high dimensional component through a multiobjective EA. The analysis of estimations for the quality measures based on a reduced subset of the pattern problems could be interesting.

4. *SD algorithms for multilabel classification problems*. Multilabel classification[78] can be seen as a generalization of binary and multiclass classifications. It does not impose an *a priori* limit on the number of elements that the set of outputs can hold. The extraction of SD rules for this kind of problems imposes a set of additional difficulties and restriction to the search process. It would be interesting to develop new SD algorithms for this type of problems.

This overview has presented an analysis about the evolutionary extraction of descriptive rules in SD. A complete analysis of different approaches, type of rules, codification schemes, genetic operators, and quality measures has been performed. Moreover, it has been shown the development carried out of EA-based processes for generating SD rules and for optimizing SD rules obtained in a previous stage. On the other hand, main guidelines that should be satisfied by any SD approach in order to facilitate the selection of quality measures more adapted both to evaluating the subgroups discovered and to guide the search process have been presented.

Considering the distinguishing features and the state of the art for SD task, the advantages, and benefits that the use of EAs for this data mining task have been shown.

ACKNOWLEDGMENTS

This work was partially supported by the Spanish Ministry of Economy and Competitiveness under projects TIN2012-33856 (FEDER Founds), the Spanish Ministry of Education and Science under Projects TIN2010-15055 and TIN2011-28488, and by the Andalusian Research Plans TIC-3928, P11-TIC-7765, and P10-TIC-6858 (FEDER Founds).

REFERENCES

1. Herrera F, Carmona CJ, González P, del Jesus MJ. An overview on subgroup discovery: foundations and applications. *Knowl Inform Syst* 2011, 29:495–525.
2. Carmona CJ, González P, del Jesus MJ, Navío M, Jiménez L. Evolutionary fuzzy rule extraction for subgroup discovery in a psychiatric emergency department. *Soft Comput* 2011, 15:2435–2448.
3. Gamberger D, Lavrac N. Expert-guided subgroup discovery: methodology and application. *J Artif Intell Res* 2002, 17:501–527.

4. Gamberger D, Lavrac N. Active subgroup mining: a case study in coronary heart disease risk group detection. *Artif Intell Med* 2003, 28:27–57.
5. Gamberger D, Lavrac N, Krstaic A, Krstaic G. Clinical data analysis based on iterative subgroup discovery: Experiments in brain ischaemia data analysis. *Appl Intell* 2007, 27:205–217.
6. Lavrac N, Cestnik B, Gamberger D, Flach PA. Decision support through subgroup discovery: three case studies and the lessons learned. *Mach Learn* 2004, 57:115–143.
7. Carmona CJ, Chrysostomou C, Seker H, del Jesus MJ. Fuzzy rules for describing subgroups from Influenza A virus using a multi-objective evolutionary algorithm. *Appl Soft Comput* 2013, 13:3439–3448.
8. Gamberger D, Lavrac N, Zelezny F, Tolar J. Induction of comprehensible models for gene expression datasets by subgroup discovery methodology. *J Biomed Inform* 2004, 37:269–284.
9. Schmidt J, Hapfelmeier A, Mueller M, Perneczky R, Kurz A, Drzezga A, Kramer S. Interpreting PET scans by structured patient data: a data mining case study in dementia research. *Knowl Inform Syst* 2010, 24:149–170.
10. Trajkovski I, Zelezny F, Lavrac N, Tolar J. Learning relational descriptions of differentially expressed gene groups. *IEEE Trans Syst Man Cybern B Cybern* 2008, 38:16–25.
11. Bäck T, Fogel D, Michalewicz Z. *Handbook of Evolutionary Computation*. Bristol, UK: Oxford University Press; 1997.
12. Espejo P, Ventura S, Herrera F. A survey on the application of genetic programming to classification. *IEEE Trans Syst Man Cyber C Appl Rev* 2010, 40:121–144.
13. Fernández A, García S, Luengo J, Bernadó-Mansilla E, Herrera F. Genetics-based machine learning for rule induction: State of the art, taxonomy, and comparative study. *IEEE Trans Evol Comput* 2010, 14:913–941.
14. Bay S, Pazzani M. Detecting group differences: mining contrast sets. *Data Min Knowl Disc* 2001, 5:213–246.
15. Dong G, Li J, Wong L. The use of emerging patterns in the analysis of gene expression profiles for the diagnosis and understanding of diseases. In: *New Generation of Data Mining Applications*. Xxx: John Wiley & Sons; 2005, 331–354.
16. Kralj-Novak P, Lavrac N, Webb GI. Supervised descriptive rule discovery: a unifying survey of contrast set, emerging pattern and subgroup mining. *J Mach Learn Res* 2009, 10:377–403.
17. Kloesgen W. Explora: a multipattern and multistrategy discovery assistant. In: *Advances in Knowledge Discovery and Data Mining*. Menlo Park, CA: American Association for Artificial Intelligence; 1996, 249–271.
18. Wrobel S. 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*, 78–87. Trondheim, Norway: Springer, 1997.
19. Siebes A. Data surveying: foundations of an inductive query language. In: *Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining*, 269–274. Quebec, Canada: AAAI Press, 1995.
20. Wrobel S. Relational data mining. In: *Inductive Logic Programming for Knowledge Discovery in Databases*. Berlin: Springer; 2001, 74–101.
21. Atzmueller M, Puppe F, Buscher HP. Towards knowledge-intensive subgroup discovery. In: *Proceedings of the Lernen – Wissensentdeckung – Adaptivität – Fachgruppe Maschinelles Lernen*, 111–117, 2004.
22. Lavrac N, Kavsek B, Flach PA, Todorovski L. Subgroup discovery with CN2-SD. *J Mach Learn Res* 2004, 5:153–188.
23. Kavsek B, Lavrac N. APRIORI-SD: adapting association rule learning to subgroup discovery. *Appl Artif Intell* 2006, 20:543–583.
24. Atzmueller M, Puppe F. 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*, 6–17. Berlin, Germany: Springer, 2006.
25. Grosskreutz H, Rueping S. On subgroup discovery in numerical domains. *Data Min Knowl Disc* 2009, 19:210–216.
26. del Jesus MJ, González P, Herrera F, Mesonero M. Evolutionary fuzzy rule induction process for subgroup discovery: a case study in marketing. *IEEE Trans Fuzzy Syst* 2007, 15:578–592.
27. del Jesus MJ, González P, Herrera F. Multiobjective genetic algorithm for extracting subgroup discovery fuzzy rules. In: *Proceedings of the IEEE Symposium on Computational Intelligence in Multicriteria Decision Making*, 50–57. Honolulu, US: IEEE Press, 2007.
28. Carmona CJ, González P, del Jesus MJ, Herrera F. NMEEF-SD: non-dominated multi-objective evolutionary algorithm for extracting fuzzy rules in subgroup discovery. *IEEE Trans Fuzzy Syst* 2010, 18:958–970.
29. Rodriguez D, Ruiz R, Riquelme JC, Aguilar-Ruiz JS. Searching for rules to detect defective modules: a subgroup discovery approach. *Inform Sci* 2012, 191:14–30.
30. Luna JM, Romero JR, Romero C, Ventura S. Discovering subgroups by means of genetic programming. In: *Proceedings of the 16th European Conference on Genetic Programming*, 121–132. Vienna, Austria: LNCS, 2013.
31. Zadeh LA. The concept of a linguistic variable and its applications to approximate reasoning. Parts I, II, III. *Inform Sci* 1975, 8–9:199–249, 301–357, 43–80.

32. Xiong N, Litz L, Ressom HW. Learning premises of fuzzy rules for knowledge acquisition in classification problems. *Knowl Inform Syst* 2002, 4:96–111.
33. Agrawal R, Mannila H, Srikant R, Toivonen H, Verkamo AI. Fast discovery of association rules. In: Fayyad UM, Piatetsky-Shapiro G, Smyth P, Uthurusamy R, eds. *Advances in Knowledge Discovery and Data Mining*. Menlo Park, CA: AAAI Press; 1996, 307–328.
34. Noda E, Freitas AA, Lopes HS. Discovering interesting prediction rules with a genetic algorithm. *IEEE Cong Evol Comput* 1999, 2:1322–1329.
35. Brin S, Motwani R, Ullman JD, Tsur S. Dynamic itemset counting and implication rules for market basket data. In: *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data*, 255–264. Arizona, US; ACM Press, 1997.
36. Lavrac N, Flach PA, Zupan B. Rule evaluation measures: a unifying view. In: *Proceedings of the 9th International Workshop on Inductive Logic Programming*, volume 1634 of *LNCS*, 174–185. Bled, Slovenia: Springer, 1999.
37. Grosskreutz H, Rueping S, Wrobel S. Tight optimistic estimates for fast subgroup discovery. In: *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 440–456, 2008.
38. Holland JH. *Adaptation in Natural and Artificial Systems*. Cambridge, MA: University of Michigan Press; 1975.
39. Goldberg DE. *Genetic Algorithms in Search, Optimization and Machine learning*. Boston, MA: Addison-Wesley Longman Publishing Co., Inc; 1989.
40. Schwefel HP. Sixth-generation computer technology series. In: *Evolution and Optimum Seeking*. xxx: Wiley; 1995.
41. Fogel DB. *Evolutionary Computation – Toward a New Philosophy of Machine Intelligence*. Piscataway, NJ: IEEE Press; 1995.
42. Koza JR. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press; 1992.
43. Freitas AA. A survey of evolutionary algorithms for data mining and knowledge discovery. In: *Advances in Evolutionary Computation*. Berlin: Springer; 2003, 819–845.
44. Eshelman LJ, Schaffer JD. Real-coded genetic algorithms and interval-schemata. In: *Foundations of Genetic Algorithms 2*. Colorado, US: Kaufmann Publishers; 1993, 187–202.
45. Holland JH, Reitman JS. Cognitive systems based on adaptive algorithms. In: Waterman DA, Hayes-Roth F, eds. *Pattern Directed Inference Systems*. New York, NY: Academic Press; 1978, 313–329.
46. Venturini G. SIA: a supervised inductive algorithm with genetic search for learning attributes based concepts. In: *Proceedings European Conference on Machine Learning*, volume 667 of *LNAI*, 280–296. Vienna, Austria: Springer, 1993.
47. Greene DP, Smith SF. Competition-based induction of decision models from examples. *Mach Learn* 1993, 13:229–257.
48. Smith SF. A learning system based on genetic adaptive algorithms. PhD Thesis, Pittsburgh, PA, USA, 1980.
49. Eiben AE, Smith JE. *Introduction to Evolutionary Computation*. Berlin: Springer; 2003.
50. Coello CA, Lamont GL, van Veldhuizen DA. Genetic and evolutionary computation. In: *Evolutionary Algorithms for Solving Multi-Objective Problems*. 2nd ed. Berlin: Springer; 2007.
51. Deb K. *Multi-Objective Optimization using Evolutionary Algorithms*. New York, NY: John Willey & Sons; 2001.
52. Pachón V, Mata J, Domínguez JL, MaÃ'a MJ. A multi-objective evolutionary approach for subgroup discovery. In: *Proceedings of the 6th International Conference on Hybrid Artificial Intelligence Systems*, volume 6671 of *LNAI*, 271–278, 2011.
53. Herrera F. Genetic fuzzy systems: taxomony, current research trends and prospects. *Evol Intell* 2008, 1:27–46.
54. Alcalá-Fdez J, Fernández A, Luengo J, Derrac J, García S, Sánchez L, Herrera F. KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *J Mult-Valued Log Soft Comput* 2011, 17:255–287.
55. Alcalá-Fdez J, Sánchez L, García S, del Jesus MJ, Ventura S, Garrell JM, Otero J, Romero C, Bacardit J, Rivas VM, et al. KEEL: a software tool to assess evolutionary algorithms for data mining Problems. *Soft Comput* 2009, 13:307–318.
56. Carmona CJ, González P, del Jesus MJ, Romero C, Ventura S. Evolutionary algorithms for subgroup discovery applied to e-learning data. In: *Proceedings of the IEEE International Education Engineering*, 983–990, 2010.
57. Carmona CJ, González P, del Jesus MJ, Ventura S. Subgroup discovery in an e-learning usage study based on Moodle. In: *Proceedings of the International Conference of European Transnational Education*, 446–451, 2011.
58. Aguilar-Ruiz JS, Riquelme JC, Toro M. Evolutionary learning of hierarchical decision rules. *IEEE Trans Syst Man Cyber B Cyber* 2003, 33:324–331.
59. Asuncion A, Newman DJ. UCI Machine Learning Repository; 2007.
60. Luna JM, Romero JR, Ventura S. Design and behavior study of a grammarguided genetic programming algorithm for mining association rules. *Knowl Inform Syst* 2012, 32:53–76.
61. Zitzler E, Laumanns M, Thiele L. 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*, 95–100, 2002.

62. Berlanga FJ, del Jesus MJ, González P, Herrera F, Mesonero M. Multiobjective evolutionary induction of subgroup discovery fuzzy rules: a case study in marketing. In: *Proceedings of the 6th Industrial Conference on Data Mining*, volume 4065 of *LNCS*, 337–349. Leipzig, Germany: Springer, 2006.
63. Deb K, Pratap A, Agrawal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 2002, 6:182–197.
64. Carmona CJ, Ramírez-Gallego S, Torres F, Bernal E, del Jesus MJ, García S. Web usage mining to improve the design of an e-commerce website: OrOliveSur.com. *Expert Syst Appl* 2012, 39:11243–11249.
65. Carmona CJ, González P, García-Domingo B, del Jesus MJ, Aguilera J. MEFES: an evolutionary proposal for the detection of exceptions in subgroup discovery. An application to concentrating photovoltaic technology. *Knowl-Based Syst* 2013, 54:73–85.
66. Herrera F, Martínez L. A 2-tuple fuzzy linguistic representation model for computing with words. *IEEE Trans Fuzzy Syst* 2000, 8:746–752.
67. Carmona CJ, González P, Gacto MJ, del Jesus MJ. Genetic lateral tuning for subgroup discovery with fuzzy rules using the algorithm NMEEF-SD. *Int J Comput Intel Syst* 2012, 5:355–367.
68. Alcalá R, Ducange P, Herrera F, Lazzerini B, Marcelloni F. A multi-objective evolutionary approach to concurrently learn rule and data bases of linguistic fuzzy rule-based systems. *IEEE Trans Fuzzy Syst* 2009, 17:1106–1122.
69. Alcalá R, Alcalá-Fdez J, Herrera F. A proposal for the genetic lateral tuning of linguistic fuzzy systems and its interaction with rule selection. *IEEE Trans Fuzzy Syst* 2007, 15:616–635.
70. Robles I, Alcalá R, Benítez JM, Herrera F. Evolutionary parallel and gradually distributed lateral tuning of fuzzy rule-based systems. *Evol Intell* 2009, 2:5–19.
71. Eshelman LJ. The CHC adaptive search algorithm: how to have safe search when engaging in nontraditional genetic recombination. In: Rawlings GJE, eds. *Foundations of Genetic Algorithms*. Burlington, Massachusetts: Morgan Kaufman; 1991, 265–283.
72. Konar A. *Computational Intelligence: Principles, Techniques and Applications*. New York: Springer-Verlag New York, Inc.; 2005.
73. Atzmueller M, Lemmerich F. Fast subgroup discovery for continuous target concepts. In: *Proceedings of the 18th International Symposium on Methodologies for Intelligent Systems*, volume 5722 of *LNAI*, 35–44. Prague, Czech Republic: Springer, 2009.
74. Domingo C, Gavaldá R, Watanabe O. Adaptive sampling methods for scaling up knowledge discovery algorithms. *Data Min Knowl Disc* 2002, 6:131–152.
75. Cano JR, Herrera F, Lozano M, García S. Making CN2-SD subgroup discovery algorithm scalable to large size data sets using instance selection. *Expert Syst Appl* 2008, 35:1949–1965.
76. Scheffer T, Wrobel S. Finding the most interesting patterns in a database quickly by using sequential sampling. *J Mach Learn Res* 2002, 3:833–862.
77. Scholz M. Knowledge-based sampling for subgroup discovery. In: *International Seminar on Local Pattern Detection*, volume 3539 of *LNAI*, 171–189. Berlin: Springer, 2005.
78. Tsoumakas G, Katakis I, Vlahavas I. Mining multi-label data. In: Maimon O, Rokach L, eds. *Data Mining and Knowledge Discovery Handbook*. Berlin: Springer; 2010, 667–685.