



# An overview of emerging pattern mining in supervised descriptive rule discovery: taxonomy, empirical study, trends, and prospects

A.M. García-Vico,<sup>1</sup> C.J. Carmona,<sup>2,3</sup> D. Martín,<sup>4</sup> M. García-Borroto<sup>4</sup> and M.J. del Jesus<sup>1\*</sup>

Emerging pattern mining is a data mining task that aims to discover discriminative patterns, which can describe emerging behavior with respect to a property of interest. In recent years, the description of datasets has become an interesting field due to the easy acquisition of knowledge by the experts. In this review, we will focus on the descriptive point of view of the task. We collect the existing approaches that have been proposed in the literature and group them together in a taxonomy in order to obtain a general vision of the task. A complete empirical study demonstrates the suitability of the approaches presented. This review also presents future trends and emerging prospects within pattern mining and the benefits of knowledge extracted from emerging patterns. © 2017 The Authors. *WIREs Data Mining and Knowledge Discovery* published by Wiley Periodicals, Inc.

How to cite this article:

*WIREs Data Mining Knowl Discov* 2018, 8:e1231. doi: 10.1002/widm.1231

## INTRODUCTION

Emerging pattern mining (EPM)<sup>1</sup> is a data mining task that searches discriminative patterns whose support increases significantly from one class or dataset to another.

Mined patterns can describe discriminative behavior between classes or emerging trends amongst datasets with respect to a property of interest by means of an understandable representation.<sup>2</sup> In this

way, EPM is halfway between prediction and description because it describes some relationships in data, which is common in unsupervised learning tasks such as clustering, by means of a target variable typically used in classification. In fact, EPM is interesting for researchers in different fields like chemistry,<sup>3,4</sup> disease detection,<sup>5,6</sup> bioinformatics,<sup>7–9</sup> and others<sup>10</sup> for its differentiating potential.

EPM belongs to the supervised descriptive rule discovery (SDRD) framework.<sup>2</sup> Similar tasks in SDRD are subgroup discovery<sup>11,12</sup> and contrast set mining.<sup>13</sup> Thanks to its nature, EPM can be used to describe emerging or differentiating behavior with respect to a property of interest in data. Hence, the main objectives of EPM are the detection of differentiating characteristics between classes and the discovery of emerging trends in time-stamped data.<sup>1,14</sup>

This paper presents an overview of EPM from the descriptive point of view. The main objective is to demonstrate the descriptive capacity of patterns

\*Correspondence to: mijesus@ujaen.es

<sup>1</sup>Department of Computer Science, University of Jaén, Jaén, Spain

<sup>2</sup>Department of Civil Engineering, Languages and Systems Area, University of Burgos, Burgos, Spain

<sup>3</sup>Leicester School of Pharmacy, De Montfort University, Leicester, UK

<sup>4</sup>Department of Artificial Intelligence, Technological University of La Habana, Havana, Cuba

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

extracted by emerging patterns (EPs) algorithms, according to the SDRD framework. In order to achieve this objective, the paper is organized as follows: First, EPM is defined, where different kinds of EPs and the main quality measures used are described. The next section shows a taxonomy where EPs algorithms are grouped into different categories. This review is completed with a wide experimental study over a battery of datasets. The results of this work allow us to determine the suitability of different approaches presented throughout the literature and to determine new research lines in order to create new proposals focused on description. Finally, the trends and prospects of EPM are outlined.

## EMERGING PATTERN MINING

The EPM problem was defined by Dong and Li<sup>1,14</sup> as the task of finding patterns whose support increases significantly from one dataset to another.

Formally, let  $I = \{i_1, i_2, \dots, i_n\}$  be a set of selectors.<sup>15</sup> A selector is a relational statement (Attribute # Set), where *Set* is a value or a set of values that belongs to the domain of the feature *Attribute*, and # is a relational operator which could be =, ≠, ∈, ∉, >, <, ≥, ≤. A logical complex (l-complex)<sup>15</sup> is a specific type of pattern which is a conjunction of selectors. An example *E* contains an l-complex, or the l-complex covers *E*, if *E* satisfies all the selectors of the l-complex. An EP *x* is an l-complex whose growth rate (GR)<sup>1</sup> is higher than a given threshold  $\rho \geq 1$ . This GR is defined as:

$$GR(x) = \begin{cases} 0, & \text{If } Sup_{D_1}(x) = Sup_{D_2}(x) = 0, \\ \infty, & \text{If } Sup_{D_2}(x) \neq 0 \wedge Sup_{D_1}(x) = 0, \\ \frac{Sup_{D_2}(x)}{Sup_{D_1}(x)}, & \text{another case,} \end{cases} \quad (1)$$

where,  $Sup_{D_1}(x)$  is the support of *x* in the first dataset ( $D_1$ ) and  $Sup_{D_2}(x)$  is the support with respect to the second dataset ( $D_2$ ). Supports are calculated as

$$Sup_{D_1}(x) = \frac{count_{D_1}(x)}{|D_1|}$$

and

$$Sup_{D_2}(x) = \frac{count_{D_2}(x)}{|D_2|},$$

where,  $count_{D_i}(x)$  is the number of examples covered by *x* in dataset *i* and  $|D_i|$  is the number of examples in dataset *i*.

The EPM was conceived from different points of view in order to describe:

- Differentiating characteristics between classes or datasets.
- Emerging trends in time-stamped datasets.

As an illustrative example, two EPs were taken from the Mushroom dataset available at the UCI repository<sup>16</sup>. This dataset contains two values for the class (edible and poisonous)<sup>1</sup>:

$$X = \{(Odor = none), (G.Size = broad), (Ring.Number = one)\}$$

$$Y = \{(Bruises = no)(G.Spacing = close), (Veil.Color = white)\}$$

The results obtained for each EP are presented in Table 1. The pattern *X* is an EP from Poisonous to Edible with a  $GR = \infty$ , while the other EP has a  $GR = 21.4$  from Edible to Poisonous. This means, on the one hand, that instances matching pattern *X* only appear in the edible class. On the other hand, instances matching pattern *Y* are 21.4 times more likely to appear in class poisonous than in edible. As can be observed, both patterns are simple and describe high discriminative characteristics for each class.

These patterns can be represented as rules in the form:

$$P : \text{Cond} \rightarrow \text{Class}$$

where, *Cond* is a conjunction of selectors and *Class* is the value of the target variable to analyze. The previous patterns could be represented as follows:

$$\text{pattern}(X) : \text{If}(\text{Odor} = \text{none}) \wedge (\text{G.Size} = \text{broad}) \wedge (\text{Ring.Number} = \text{one}) \rightarrow \text{Edible}$$

$$\text{pattern}(Y) : \text{If}(\text{Bruises} = \text{no}) \wedge (\text{G.Spacing} = \text{close}) \wedge (\text{Veil.Color} = \text{white}) \rightarrow \text{Poisonous}$$

**TABLE 1** | Results Obtained for Different Emerging Patterns (EPs) in the Mushroom Dataset

EP	Sup <sub>Poisonous</sub>	Sup <sub>Edible</sub>	GR
X	0.000	0.639	∞
Y	0.814	0.038	21.4

It is important to note that, by definition, patterns obtained through EPs algorithms do not satisfy the downward closure property,<sup>17</sup> which states that subsets of frequent patterns are also frequent. EPM represents an NP-Hard problem with respect to the number of variables<sup>18</sup> due to the nonapplicability of such a property. Notice that the  $X$  pattern presented in the previous example is an EP, while its sub-pattern  $X = \{(\text{Odor} = \text{none})\}$ , however, is not an EP. In fact, none of its singleton sub-patterns are EPs, so the downward closure property cannot be applied anyway. The growth rate is based on the change proportion between supports and not for the current support. Therefore, a specific pattern, i.e., with high number of selectors and generally with low support, could obtain a higher change proportion than a general pattern, which has a lower number of selectors and generally has higher support. Thus, it is not possible to achieve an exact solution in a reasonable time. However, throughout the literature EPM uses the downward closure property to speed up algorithms as a pruning method, since low-support EPs are considered irrelevant.

### Context of Emerging Pattern Mining

Traditionally, in data mining there are two clearly defined approaches: supervised and unsupervised learning. In general, supervised learning refers to tasks such as classification,<sup>19</sup> regression,<sup>19</sup> and temporal series analysis and classification.<sup>20</sup> Their main aim is to predict the value of a property of interest in new incoming instances. On the other hand, unsupervised learning is referred to tasks such as summarization<sup>21</sup> or association rule mining.<sup>22</sup> In this case, the aim is to describe relationships in data according to different properties such as support or confidence, and there is no interest property.

EPM attempts to build a descriptive model of data with respect to a property of interest, i.e., using supervised learning. Therefore, it is somewhere between prediction and description, because it describes some relationships between data with respect to an interest property or class. This model is built by patterns with specific characteristics in order to describe emerging or discriminative behavior in data. These descriptions should be general, simple, and as precise as possible in order to be comprehensible for the experts. EPM belongs to the SDRD framework, which includes similar tasks such as subgroup discovery<sup>11,12</sup> and contrast set mining.<sup>13</sup> The main difference between EPM and subgroup discovery is that while subgroup discovery finds unusual distributions with respect to the property of interest, EPM

finds relationships in data with respect to the possible values of the target variable. On the other hand, the difference with contrast set mining is that while contrast set mining tries to find patterns with high difference of support, EPM tries to find patterns with a high growth rate, which allows us to describe emerging tendencies in data.

### Types of Emerging Patterns

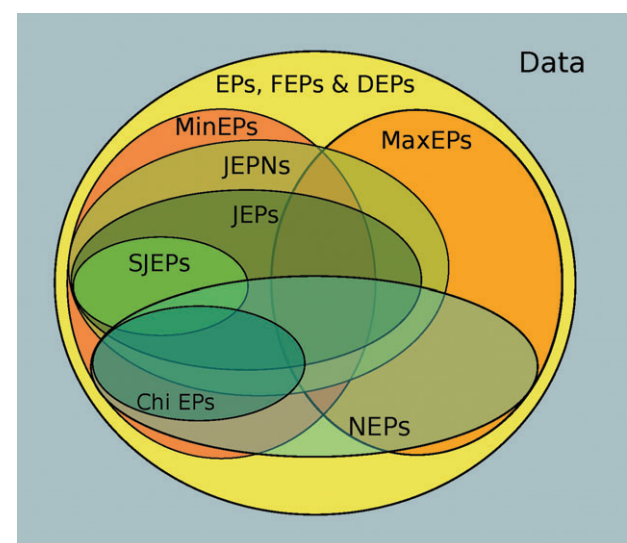
EPM is considered an NP-Hard problem with respect to the number of variables. To approximate this problem efficiently, the authors have attempted to reduce the number of patterns extracted. They try to take only those patterns which describe some specific and interesting relationships between variables. This yields a more interesting and simpler set of patterns which should be as precise as possible. These types of patterns allow the description of a problem in an easy and direct way.

Figure 1 presents the relationships between the types of EPs most used throughout the literature, which are summarized below:

- *Jumping emerging patterns* (JEPs). These are EPs with a growth rate equal to infinity, i.e., the JEP covers examples for a single class. They have a great differentiator character between classes<sup>23</sup>:

$$\text{JEPs} = \{P \mid \text{GR}(P) = \infty\} \quad (2)$$

- *Minimal emerging patterns* (MinEPs). A MinEP is defined as an EP whose sub-patterns are not



**FIGURE 1** | Relationships among different types of emerging patterns (EPs).

EPs any more. These kinds of patterns are the most general EPs. They are interesting for description because, in general, they contain a low number of variables<sup>24,25</sup>:

$$\text{MinEPs} = \{P \mid \text{GR}(P) \geq \rho \wedge \nexists S \subset P / \text{GR}(S) \geq \rho\} \quad (3)$$

- *Maximal emerging patterns* (MaxEPs). This type of pattern is the opposite of the minimal. A MaxEP is an EP whose super-patterns are not EPs any more. This produces very specific patterns, which are very precise<sup>18</sup>:

$$\text{MaxEPs} = \{P \mid \text{GR}(P) \geq \rho \wedge \nexists Q \supset P / \text{GR}(Q) \geq \rho\} \quad (4)$$

- *Essential jumping emerging patterns* (eJEPs). These are also known as *strong jumping emerging patterns* (SJEPs). This type of pattern is the intersection of the MinEPs set with the JEPs set<sup>24,25</sup>:

$$\text{eJEPs} = \{P \mid \text{GR}(P) = \infty \wedge \nexists S \subset P / \text{GR}(S) = \infty\} \quad (5)$$

These kinds of patterns are easy to understand and have high predictive power. They have been widely used throughout the literature due to their descriptive properties.

- *Noise-Tolerant emerging patterns* (NEPs). Also known as *constrained emerging patterns* (CEPs). This type is very close to JEPs but it allows us to obtain precise patterns in noisy environments. Formally, a NEP from  $D_1$  to  $D_2$  must satisfy<sup>25,26</sup>:

$$\text{NEPs} = \{P \mid \text{Sup}_{D_1}(P) \leq \delta_1 \wedge \text{Sup}_{D_2}(P) \geq \delta_2\}, \quad (6)$$

where, normally,  $\delta_2 \gg \delta_1$  in order to obtain patterns with a high GR.

- *JEPs with negation* (JEPNs). They are an extension of the JEPs subset. These kinds of patterns allow the representation of negated values, i.e., a negated value represents the idea that such a value does not appear in an example. Let  $\mathcal{I}$  denote the set of existing selectors and  $\bar{\mathcal{I}} = \{\bar{i}\}_{i \in \mathcal{I}}$  the set of selectors with negated values. The new set of possible selectors in a JEPN is defined as  $\mathcal{P} = \{p \subseteq \mathcal{I} \cup \bar{\mathcal{I}} \mid \forall i \in \mathcal{I} \ i \in p \rightarrow \bar{i} \notin p\}$ . Thus, JEPNs are defined as<sup>27</sup>:

$$\text{JEPNs} = \{P \mid \text{GR}(P) = \infty\}, \quad (7)$$

where,  $P = \{x_1, x_2, \dots, x_n \mid x_n \in \mathcal{P}\}$ . An example of a JEPN could be the following pattern:

$$X = \{(\text{Odor} = \overline{\text{none}}), (\text{G.Size} = \text{broad}), \\ (\text{Ring.Number} = \text{one})\},$$

which, represents those patterns without the value *none* in the variable *Odor*, e.g., *Odor* = almond, and contains (G.Size = broad) and (Ring.Number = one).

- *Chi emerging patterns* (Chi EPs). These are similar to NEPs but introduce a  $\chi^2$  statistical test in order to improve the descriptive capacity of the pattern. Chi EPs extracted are also considered as significant. A pattern is considered Chi EP when<sup>28,29</sup>:
  - $\text{Sup}(P) \geq \xi$ , where  $\xi$  is the minimum support threshold.
  - $\text{GR}(P) \geq \rho$ , where  $\rho$  is the minimum growth rate threshold.
  - $\nexists S \subset P \mid (\text{Sup}(S) \geq \xi) \wedge (\text{GR}(S) \geq \rho) \wedge (\text{Strength}(S) \geq \text{Strength}(P))$ , where  $\text{Strength}(P) = \frac{\text{GR}(P)}{\text{GR}(P)+1} \text{Sop}_2(P)$ .
  - $|P| = 1 \vee |P| > 1 \wedge \forall S(S \subset P \wedge |S| = |P| - 1) \Rightarrow \text{chi}(P, S) \geq \eta$ , where  $\eta = 3.84$  is the minimum threshold for the  $\chi^2$  test and  $\text{chi}(P, S)$  is calculated through a contingency table.

Despite the efforts of researchers in extracting only those types of EPs that are relevant, most of these kinds of patterns can be extracted at once with a single execution of a mining algorithm. This is because these patterns are interrelated. Figure 1 shows the schema of these relationships. At the top level, there are classical EPs.

Inside the EPs there are some interesting groups:

- *MinEPs and MaxEPs*. As mentioned previously, MaxEPs are the opposite of MinEPs and they form two large groups. On the one hand, MinEPs are the most general EPs. Their properties have been widely used throughout the literature due to an easier mining and good results when joined with JEPs. On the other hand, MaxEPs are very specific patterns with normally higher precision than MinEPs. However, mining this kind of pattern is computationally expensive.
- *The 'Jumping' group*. In this group there are the JEPNs, the JEPs, and the SJEPs. As can be seen, one is a subset of the other. The most

general is the JEPNs due to the introduction of the negated values. After that, JEPs are JEPNs with only positive values. And finally, SJEPs are minimal JEPs, and thus are also a subset of the MinEPs. Moreover, some JEPs and JEPNs can be MaxEPs. However, the number of patterns with this condition is less than those JEPs or JEPNs that are minimal.

- *The noise tolerant group.* NEPs and Chi EPs belong to this group. The NEPs is a wide set of patterns. In this set we can find some patterns that are JEPNs, JEPs, or even SJEPs. Additionally, the number of NEPs that can be maximal is similar to those that are minimal because more MaxEPs can fit with the conditions of NEPs. In fact, NEPs are more likely to be MaxEPs than MinEPs because MaxEPs tend to have higher GR due to their precision. On the other hand, Chi EPs is a subset of NEPs that tends to be minimal due to the  $\chi^2$  test joined with condition 3.

### Quality Measures

EPM has been explicitly defined with respect to the GR ((1)) measure, so it is one of the most important measures of the task.<sup>1</sup> However, it is necessary to analyze the EPs from a new perspective. Knowledge obtained from these patterns allows the experts to learn the main discriminative characteristics between classes, or emerging trends in data, easily. In fact, this is more interesting when the amount of data grows, because it is more difficult to obtain a general and precise description of data in these environments.<sup>30</sup> Therefore, an analysis of algorithms and patterns is necessary in order to measure characteristics that fulfill these requirements of knowledge. These requirements are measured by means of the precision, generality and interest of the patterns extracted.<sup>2</sup> This assertion is key in order to bring precise and concise knowledge to the experts.

EPs were conceived for analysis between datasets and, for extension, between classes of a dataset. However, the number of these classes could be greater than 2, i.e., multiclass datasets. An *One-vs-All* (OVA)<sup>31</sup> decomposition of the problem is necessary in order to deal with multiclass problems. This considers the positive class as the one represented in the consequent part of the pattern and the negative for the remaining classes, regardless of the number of classes contained in the dataset.

It is necessary to show the confusion matrix obtained for one pattern in order to determine its

quality.<sup>32</sup> Table 2 represents a confusion matrix for a pattern, where

- $p$  is the number of examples correctly covered.
- $\bar{p}$  is the number of examples not covered by the class.
- $n$  is the number of examples incorrectly covered.
- $\bar{n}$  is the number of examples not covered by the nonclass.
- $p + n$  is the number of examples covered by the pattern.
- $\bar{p} + \bar{n}$  is the number of examples not covered by the pattern.
- $\text{Pos} = p + \bar{p}$  is the number of examples for the actual class.
- $\text{Neg} = n + \bar{n}$  is the number of examples for the remaining classes.
- $T = \text{Pos} + \text{Neg}$  is the number of examples for the whole dataset.

For EPM, there are no measures that can estimate the descriptive characteristics mentioned previously. However, for the SDRD framework there do exist several measures for this objective. Then, it is possible to measure the descriptive characteristics of EPs according to the SDRD framework.<sup>33</sup> A comparative study of the behavior of several quality measures was presented in.<sup>34</sup> In this study, the measures were classified according to their main objective and grouped following the Pearson correlation value. Following the classification performed in this study, the most relevant measures for EPM are summarized below.

### Quality Measures Based on Precision

Quality measures that belong to this group quantify the precision of a given pattern. In EPM, the acquisition of patterns with high precision is important in order to obtain accurate knowledge. The most relevant quality measures within this group are presented below:

**TABLE 2** | Confusion Matrix for a Pattern

True condition	Predicted condition	
	Positive	Negative
Positive	$p = tp$	$\bar{p} = fn$
Negative	$n = fp$	$\bar{n} = tn$

- *Growth rate* (GR). This is the measure that defines an EP. It calculates the ratio between the support of the pattern in the positive class and the support in the negative. It is interpreted as the discriminative power of a pattern<sup>1</sup>:

$$GR(P) = \begin{cases} 0, & \text{If } \frac{p}{Pos} = \frac{n}{Neg} = 0, \\ \infty, & \text{If } \frac{p}{Pos} \neq 0 \wedge \frac{n}{Neg} = 0, \\ \frac{p \cdot Neg}{Pos \cdot n}, & \text{another case.} \end{cases} \quad (8)$$

Where  $Pos = p + \bar{p}$  and  $Neg = n + \bar{n}$ .

- *Confidence* (Conf). This is defined as the ratio of the predictive capacity of a pattern for the positive class, i.e., examples correctly covered among all examples covered<sup>35</sup>:

$$Conf(P) = \frac{p}{p+n} \quad (9)$$

This measure can be interpreted as the accuracy of the pattern with respect to the examples it covers. It only measures precision.

### Quality Measures Based on Gain Accuracy and Information

The quality measures of this group are considered as hybrids because they have different objectives: to improve the generality, precision, and interest of the pattern. The most relevant quality measures are presented below:

- *Weighted relative accuracy* (WRAcc). Also known as unusualness in the literature. It measures the trade-off between generality and accuracy gain<sup>36</sup>:

$$WRAcc(P) = \frac{p+n}{T} \left( \frac{p}{p+n} - \frac{Pos}{T} \right) \quad (10)$$

Where  $T = Pos + Neg$ . This value represents the balance between the coverage of the pattern,  $\left(\frac{p+n}{T}\right)$  and its accuracy gain, i.e., the comparison of the accuracy obtained with respect of a naive classification by considering all examples as positive  $\left(\frac{p}{p+n} - \frac{Pos}{T}\right)$ . It is normalized because the domain of this measure depends on the class analyzed.<sup>37</sup> A high value can be interpreted as a high balance between the generality of the pattern and its confidence.

- *False positive rate* (FPR). This measures the percentage of examples incorrectly covered with

respect to the total amount of negative examples<sup>38</sup>:

$$FPR(P) = \frac{n}{Neg} \quad (11)$$

It can be interpreted as the specificity of the pattern, and normally this value must be minimized. This measure can measure both precision and generality.

- *True positive rate* (TPR). Measures the percentage of examples correctly covered with respect to the total number of positive examples<sup>39</sup>:

$$TPR(P) = \frac{p}{Pos} \quad (12)$$

This can be interpreted as the sensitivity of the pattern. It combines precision and generality related to the class.

### Quality Measures Based on Interpretability

In SDRD, interpretability is defined as the ease of analysis of the results by the experts. These measures are normally related to the complete set of patterns extracted and are calculated through:

- *Number of patterns* (# Patterns). This measure indicates the number of patterns extracted from a given problem.
- *Number of variables of the antecedent* (# Variables). This measure indicates to the experts the complexity of the patterns extracted. The number of variables for a set of patterns is computed as the average of the variables for each pattern of that set.

In SDRD, a low number of patterns and variables ease the analysis.

## TAXONOMY FOR EPM ALGORITHMS

This section outlines a taxonomy for EPM algorithms. It allows the identification of the strategies followed by the algorithms in order to extract those interesting subtypes of EPs presented previously. The taxonomy has been divided into four different groups of EPs algorithms. Each group contains methods whose search strategies are similar. In addition, it presents a brief description of each algorithm focused on their advantages and drawbacks with respect to descriptive induction. As a summary, Table 3 shows the main algorithms for EPM developed under this classification.

**TABLE 3** | Classification of the Main Algorithms for Emerging Pattern Mining (EPM) Developed in the Literature

Border-based algorithms	Tree-based algorithms
MBD-LLBorders <sup>1</sup>	Tree-based JEP-C <sup>40</sup>
JEPProducer <sup>41</sup>	BCEP <sup>42</sup>
DeEPS <sup>43</sup>	iEPMiner <sup>44</sup>
Decision tree-based algorithms	StrongJEP-C <sup>25</sup>
LCMine <sup>45</sup>	Top-K <sup>46</sup>
CEPMine <sup>47</sup>	DGCP <sup>48</sup>
EP RandomForest <sup>49</sup>	Evolutionary fuzzy system-based algorithms
FEPM <sup>50</sup>	EvAEP <sup>51</sup>

### Border-Based Algorithms

The concept of border was introduced within the first definition of EPM by Dong and Li<sup>1,14</sup> as an efficient method of extraction of EPs. The border concept allows a huge number of EPs to fit into a compact representation. A border is defined as a pair  $\langle \mathcal{L}, \mathcal{R} \rangle$ , where  $\mathcal{L}$ , called left-border, is a set of MinEPs and  $\mathcal{R}$ , called right-border, is a set of MaxEPs. This pair is considered a valid border when:

1.  $\mathcal{L}$  and  $\mathcal{R}$  are both *antichains*. A set of patterns  $S$  is an *antichain*, if  $\forall X, Y \in S, X \not\subseteq Y \wedge Y \not\subseteq X$ .
2. Each element of  $\mathcal{L}$  is a subset of some element of  $\mathcal{R}$ , and each element of  $\mathcal{R}$  is a superset of some element of  $\mathcal{L}$ .

These conditions allow the border to represent a convex space within the EP space where those patterns between  $\mathcal{L}$  and  $\mathcal{R}$ , i.e., those  $S$  such that  $X \subseteq S \subseteq Y$  where  $X \in \mathcal{L}$  and  $Y \in \mathcal{R}$ , are all EPs. Most of the methods within this approach use the Border-Diff procedure as their core function in order to include the borders that represent all EPs whose GR is greater than the threshold.

Figure 2 presents the general schema of border-based algorithms. First, it extracts the

**BEGIN**

Get the borders that represent patterns of  $D_1$

Get the borders that represent patterns of  $D_2$

Apply Border-Diff on each border of  $D_2$  with respect to each border of  $D_1$  and vice versa

RETURN The borders obtained by Border-Diff

**END**

**FIGURE 2** | General schema of a border-based algorithm.

borders representation of patterns that fulfill some constraints on each dataset. EPs are those patterns that are in the difference set of these borders. In order to calculate this difference efficiently, the Border-Diff procedure is executed. It handles only its borders without generating any candidate set. Within this group are the initial algorithms developed for the task.

The most important algorithms in this group are:

- *MBD-LLBorder*.<sup>1</sup> This is the pioneering algorithm for the extraction of EPs. It extracts the large borders that represent the patterns whose support is greater than a given threshold on both datasets using the Max-Miner algorithm.<sup>52</sup> After that, it performs the Border-Diff procedure several times in order to obtain the difference of these large borders. All patterns with a GR greater than a given threshold are returned.
- *JEPProducer*.<sup>41</sup> This method obtains the border representation of all JEPs. First, it obtains the horizontal border of each dataset. A horizontal border is represented as  $\langle \{\emptyset\}, \{\mathcal{R}_1\} \rangle$  and it contains all nonzero support patterns. Then, in order to obtain JEPs it executes the Border-Diff routine to get the border which represents the difference of these sets represented by the horizontal borders. It also contains incremental operators in order to keep the JEPs border representation without generating the horizontal borders when new instances arrive.
- *DeEPS*.<sup>43</sup> This method is based on lazy learning, i.e., the training data is the model itself. For each new test instance, those selectors that do not belong to the test instance are removed from the model. After that, the *JEPProducer* procedure is applied on this reduced dataset in order to obtain the border-based representation of a set of JEPs for each class that covers the test instance. The resulting set of EPs is obtained by repeating this procedure for each test instance and then removing duplicates. This

is currently the reference algorithm in this group for EPM.

In summary, the border-based strategy is a first approach to handle EPM efficiently. However, the complexity of the search space is exponential to the number of variables in the worst case. For this reason, JEPs are used in order to reduce the search space. Additionally, JEPs, which are very discriminative, are very sensible to noise. The border-based representation makes the EPs obtained by this method hard to comprehend by humans. In fact, the number of patterns obtained when expanding the borders, i.e., the representation of all patterns within the border, is huge if elements of  $\mathcal{L}$  have many fewer variables than those on  $\mathcal{R}$ . So, the descriptive properties of EPs are almost useless in this representation.

### Tree-Based Algorithms

The border-based strategy has an exponential complexity. The algorithms in this group represent the training data in a tree structure in order to efficiently mine EPs. The general mining approach that follows these algorithms is described in Figure 3.

In general, the nodes of the tree consist of a selector and the occurrence counts of the pattern formed by the path from the root to the place of the node. The root of the tree is normally a null node and the sub-tree of each child is called a component tree (CT). The selectors are sorted by a measure, which normally is GR. This keeps the most discriminative selectors closer to the root in order to perform a faster mining. Also, different pruning strategies can be performed in order to obtain only those EPs which are of interest. This strategy has the advantage of a faster mining than border-based methods.

However, this general scheme forces the reinsertion of instances when processing the second and consecutive CTs in order to keep the correct counts of patterns.<sup>25</sup> This normally creates a large bottleneck when mining. The most important algorithms within this group are:

- *Tree-based JEP-C*<sup>40</sup> (TBJEP-C). This is the first method with this approach. In fact, this method can be considered as a hybrid method, because it uses the Border-Diff procedure to obtain JEPs. The aim was to improve the mining time of the JEP-Classifier algorithm<sup>53</sup> which is a classification algorithm on top of JEPProducer. To achieve it, it uses a modification of the FP-Tree data structure<sup>54</sup> to allow the use of labeled data. Each node of the tree contains the elements of the general scheme presented in Figure 3 and a link to another node in the tree with the same selector. The method allows six different ordering methods for the selector when they are introduced in the tree: frequency, GR, inverse GR, hybrid, which is a weighted average between frequency and GR, least probable in the negative class and most probable in the positive class orderings. After all examples are inserted, the mining is performed by a depth-first traversal until a node with zero counts in one class and nonzero counts in the other is found. This node is called the base node. Then, the method collects those negative instances which contain the selector in the root and the selector in the base node by means of the node-links. After that, the JEPs are extracted by implementing the Border-Diff procedure using the JEP in the base node and the

#### BEGIN

$Ra \leftarrow$  Ranking of selectors by a given measure (in general, GR).

$T \leftarrow$  Initialise a null root.

for each example  $E$  do

Sort selectors of  $E$  according to  $Ra$ .

Add  $E$  in  $T$  as nodes in a depth-first way. Update the counter of occurrences if the node already exist in  $T$ .

end for

Mine EPs using some tree traversal strategy and some pruning strategy.

RETURN Set of mined EPs.

END

**FIGURE 3** | General algorithmic schema of a tree-based algorithm.



negative instances collected. The experiments show a speed-up of 2 and 20 times with respect to the border-based algorithm. Moreover, it allows a minimum support-threshold pruning.

- *BCEP*.<sup>42</sup> This method uses the Pattern Tree (P-Tree) structure<sup>24</sup> to mine the complete set of SJEPs. The tree is built by sorting the selectors by GR. The mining strategy is performed by a depth-first traversal finding SJEPs. The tree is pruned by means of a minimum support threshold or when a JEP is found, in order to obtain only SJEPs. After that, the method keeps only the most important SJEPs with a filtering strategy similar to the Token Competition procedure.<sup>55</sup> In this procedure, only SJEPs that can cover examples that are not already covered by better-ranked patterns are kept.
- *iEPMiner*<sup>44</sup> (iEPM). This uses the P-Tree data structure to store the dataset. The objective of this algorithm is to mine Chi-EPs. The selectors are sorted by GR when building the tree. The mining procedure follows a depth-first strategy. On each node, conditions 1, 2, and 4 of Chi-EPs are checked. If the pattern fulfills these conditions, it is saved. The pruning strategy is based on condition 4. If the value of the  $\chi^2$  test between the node and the candidate child node is less than 3.84, then the addition of the new selector does not significantly change the behavior of the EP, because it is independent with a 95% of confidence. In addition, the behavior of supersets that contains this selector will be the same, so the sub-tree is pruned. After finishing the tree traversal, condition 3 is checked over all patterns mined in order to obtain only Chi-EPs.
- *StrongJEP-C*<sup>25</sup> (SJEP-C). This algorithm mines SJEPs with a support greater than a given threshold. To obtain these patterns, it uses a new data structure called a Contrast Pattern Tree (CP-Tree) which is similar to P-Tree, but without node-links. The algorithm builds the tree sorting the selectors by support ratio in descending order. This measure is defined as  $\text{SuppRatio}(P) = \max\left(\frac{\text{TPR}(P)\text{FPR}(P)}{\text{FPR}(P)\text{TPR}(P)}\right)$ . The use of this measure allows patterns to be obtained for both classes in a single execution. The method mines the tree in a depth-first traversal, checking if the node contains zero-counts in one dataset and nonzero counts in the other, and prunes when true. In addition, it prunes when the node has a support value for both datasets lower than the threshold.
- *Top-k minimal JEPs*<sup>46</sup> (Top-k). This is an improvement of the StrongJEP method which only keeps those  $k$  patterns with the highest supports. The idea is that SJEPs with high support are the most discriminative patterns. The mining scheme is similar to SJEP-C. However, the pruning strategies are different. Top-k uses two pruning strategies: first, on each node it checks if the pattern associated to the node is considered  $D$ -discernibility minimal, i.e., let  $X$  be the pattern associated to the node,  $X$  is  $D$ -discernibility minimal if and only if  $\forall_{Y \subset X} \text{supp}D'(X) < \text{supp}D'(Y)$ , where  $\text{supp}D'(X)$  is the support of pattern for the negative class. If the node is not  $D$ -discernibility minimal, it prunes. In addition, a minimum support threshold pruning is used. This threshold grows dynamically by storing the patterns in a heap by nonascending support. Then if the number of elements in the heap is equal to  $k$ , the support threshold is raised to the support of the first element of the queue plus one. This allows an important pruning in the CP-Tree, obtaining the best  $k$  patterns much faster than SJEP-C.
- *DGCP-Tree*<sup>48</sup> (DGCP). This uses a new data structure called a Dynamically Growing Contrast Pattern Tree (DGCP-Tree). The counts of a node are stored by the BSC-Tree structure.<sup>56</sup> This structure allows efficient storing of the bit string which represents the examples covered by a selector. The main characteristic of this method is that the mining of EPs is performed during the construction of the tree. This mining is performed in two phases: First, the singleton selectors and BSC-Tree for both classes are identified and constructed for each one. In addition, they are sorted by GR and singleton SJEPs are mined. After that, the tree grows by copying those right sibling nodes as the children of the node are being processed, if they cover new instances and their support is greater than the threshold.

The mining heuristic, in almost all cases, is based on a minimum support threshold pruning in order to obtain subsets of MinEPs. In fact, most of them try to mine SJEP with high support. The main idea behind this prune is that patterns with very low support are completely useless and can be considered as noise in data. This heuristic smooths the strong sensibility of EPs to noise. This is an advantage because they reduce the search space. Thus, they are less complex and faster than the border-based approach. In

fact, this approach has been widely used in the literature. Nevertheless, the heuristics used by these algorithms for the extraction of SJEPS does not guarantee that all patterns extracted are SJEPS.

## Decision Tree-Based Algorithms

Algorithms in this section induce a set of decision trees from training data. These decision trees are modified by introducing more candidate splits, where operators such as  $=$ ,  $\neq$ ,  $\leq$ ,  $\geq$ ,  $<$ ,  $>$ ,  $\in$ ,  $\notin$  are used. The use of these operators allows the handling of numeric attributes directly without a previous discretization phase, which must be performed with border-based or tree-based approaches. Moreover, the number of EPs and their interpretability is improved by the use of these operators.

The general working schema for generating a decision tree-based algorithm is outlined in Figure 4.

The trees generated can be limited in depth in order to speed up the mining. The extraction of EPs from these trees can be performed by means of traversing all the paths of the tree, or performing additional operations on the divided search space that produces each tree. Algorithms that follow this approach are:

- *LCMine*.<sup>45</sup> This is the first algorithm that uses this approach. It generates a fixed number of trees, specified by the user. To generate a tree it is necessary to provide a vector  $(k_1, k_2, \dots, k_l)$ , where  $l$  is the maximum depth of the tree and  $k_i$  is a number which indicates that the  $k$ th best split in the  $i$ th level is picked up. The information gain measure used is entropy.<sup>57</sup> After that, a post-processing filter is executed in order to reduce redundancy.
- *CEPMine*.<sup>47</sup> This is an improvement of the *LCMine*. This algorithm induces EPs from a set of C4.5 decision trees.<sup>57</sup> The information gain

is weighted with respect to each training instance. In addition, the extraction of EPs is performed when the candidate splits are generated. In this case, whenever a split contains at least  $\mu$  objects in one class and at most one in the other one, a pattern is generated even if the split is not considered the best one. After that, the best  $k$  splits are expanded, updating object weights after each induction. The tree stops when no patterns are extracted.

- *EP random forest*<sup>49</sup> (EP-RF). This is a modification of the Random Forest<sup>58</sup> algorithm. The algorithm generates the decision trees in the same way as Random Forest. In this way, each simple decision tree creates a partitioning of the space, where instances are classified by the Random Forest algorithm. EP-RF takes the same leaf node with the same index for all generated trees. After that, examples that are within the space delimited for each tree are taken and divided into positive and negative. Once collected, EPs are extracted by executing the MBD-LLBorder algorithm described in this taxonomy.
- *FEP*.<sup>50</sup> This is a modification of *LCMine* in which the tree induction method obtains fuzzy EPs instead of crisp EPs. In this way, the calculus of the information gain for each split is adapted to deal with fuzzy logic,<sup>59,60</sup> which takes into account fuzzy memberships functions and fuzzy hedges. Also, splits are generated taking into account fuzzy hedges, where each hedge is considered a potential split. A fuzzy EP (FEP) is a modification of traditional EPs which use fuzzy logic in order to represent numerical variables. The use of fuzzy logic is performed through linguistic labels, which gives an interpretation of numerical variables which is closer to human reasoning.<sup>61</sup> An example of FEP is represented below:

```

BEGIN
while non stopping criteria do
    Generate the decision tree using a diversity generation procedure
    Extract patterns from tree
    Join patterns to the result
    Simplify the patterns
end while
END

```

**FIGURE 4** | General schema of the building of a decision tree for the extraction of emerging patterns (EPs).

$$X : \{(\text{Odor} = \text{none}), (\text{Size} \in \text{Broad}), (\text{Ring.Number} \in \text{Slightly(Low)})\}, \quad (13)$$

where,  $\text{Size} \in \text{Broad}$  and  $\text{Ring.Number} \in \text{Low}$  are numeric variables whose values are represented by the linguistic labels *Broad* and *Low* with the fuzzy hedge *Slightly*, respectively.  $\text{Odor} = \text{none}$  is a categorical variable.

Methods following this approach can deal with numerical attributes directly, so they evade a previous discretization phase. This avoids the loss of knowledge of the problem. Moreover, the addition of operators such as inequalities, etc. or the use of fuzzy logic on numeric variables obtains patterns which are more interpretable.

### Evolutionary Fuzzy System-Based Algorithms

This type of algorithms makes use of evolutionary algorithms<sup>62</sup> together with fuzzy logic under the name of evolutionary fuzzy systems (EFSs).<sup>63</sup> The objective is to find very descriptive FEPs. A general schema is presented in Figure 5.

First, the algorithm initializes a population with candidates EPs through codification of the ‘chromosome = pattern’ approach.<sup>64</sup> In this approach, an individual represents a potential EP. This population is evolved in order to find high quality patterns. The evolutionary process is guided by a fitness function which measures the quality of an individual. This function could be one of the quality measures presented previously or an aggregation of measures in order to find high descriptive patterns. Finally, those individuals with the highest fitness value are the EPs returned to the expert.

This group of algorithms for EPM consists nowadays of only one algorithm: EvAEP.<sup>51</sup> This method is based on a mono-objective evolutionary algorithm.<sup>65</sup> EvAEP follows an iterative approach where the evolutionary process is repeated until a stopping criterion is reached. The evolutionary process returns the best pattern in the population upon reaching the stopping criterion. The genetic operators used are a two-point crossover operator and a biased mutation operator.

The main advantages of this kind of method with respect to the other groups are that the search strategy performs an efficient global search through the space. Execution time can be adapted in order to find a good balance between execution time and quality of results. The representation approaches and genetic operators can be easily adapted in order to

ease this search. In addition, it is possible to improve multiple objectives by means of multi-objective evolutionary algorithms (MOEAs).<sup>66,67</sup>

## EXPERIMENTAL STUDY

This section outlines the results of an experimental study carried out in order to determine the quality of the patterns extracted for each method focused on descriptive induction. The behavior of a pattern with respect to this approach is determined by quality measures such as WRAcc, Conf, GR, TPR, and FPR. These measures combine several factors such as interpretability, precision and differentiation between sets, amongst others. Additionally, the number of patterns and the average number of variables are analyzed. In order to achieve this objective the study is structured as follows: first, the experimental framework is presented, where datasets, algorithms, and statistical procedures used are shown. Then, the analyses of the results of the study are presented below.<sup>a</sup>

### Experimental Setup

In this section, the experimental setup is shown. First, the characteristics of the datasets used are presented. After that, the EPs algorithms and their parameter configuration are shown. Finally, a brief explanation of the statistical tests used in the study is presented.

### Datasets

The algorithms in this study are compared with the use of 36 datasets coming from the UCI Repository.<sup>16b</sup> This is a compilation of well-known problems which is widely used throughout the literature. Datasets are split using a distributed optimally balanced stratified cross-validation.<sup>68</sup> These partitions of data have been taken from the KEEL Repository.<sup>69c</sup> Details of these datasets are summarized in Table 4, where the number of attributes together with the number of instances and classes for each one are shown.

Some algorithms work only with nominal variables. Thus, a discretization of these datasets is performed in order to work with these methods. The discretization process used is the Fayyad discretize.<sup>70</sup>

### Parameters and Base Algorithms

The parameters of the algorithms analyzed are presented in Table 5. Parameters were set according to the values suggested by the author of each method in their corresponding papers. For iEPM and BCEP, the authors do not recommend any parameters. In this case, parameters are chosen in order to be as similar

```

BEGIN
  P population is initialised
  while stopping criteria not reached do
    P population is evaluated
    The descendant population Q is generated through genetic operators
    Additional operations are performed in Q
    P for the next generation is replaced through P and Q
  end while
RETURN P
END

```

**FIGURE 5** | General schema of an evolutionary fuzzy system (EFS) algorithm for the extraction of emerging patterns (EPs).

**TABLE 4** | Datasets Used in the Study, Including the Number of Attributes, the Number of Examples and the Number of Classes for Each One

Name	# Attributes	# Examples	# Classes	Name	# Attributes	# Examples	# Classes
Appendicitis	7	106	2	Led7digit	7	500	10
Australian	14	690	2	Magic	10	19,020	2
Balance	4	625	3	Monk-2	6	432	2
Bands	19	365	2	Mushroom	22	5644	2
Breast	9	277	2	Newthyroid	5	215	3
Bupa	6	345	2	Page-blocks	10	5472	5
Car	6	1728	4	Phoneme	5	5404	2
Cleveland	13	297	5	Pima	8	768	2
Contraceptive	9	1473	3	Saheart	9	462	2
Crx	15	653	2	Segment	19	2310	7
Flare	11	1066	6	Tae	5	151	3
Glass	9	214	7	Tic-tac-toe	9	948	2
Haberman	3	306	2	Titanic	3	2201	2
Hayes-roth	4	160	3	Vehicle	18	846	4
Heart	13	270	2	Vowel	13	990	11
Hepatitis	19	80	2	Wine	13	178	3
Housevotes	16	232	2	Wisconsin	9	683	2
Iris	4	150	3	Yeast	8	1484	10

as possible to the other algorithms to have a fair comparison.

Due to the nondeterministic character of the EvAEP algorithm, it has been executed three times with different seeds. In this way, the results of EvAEP are the average results of the three executions on the five partitions.

There are no public implementations of any of these algorithms. Therefore, the algorithms have been implemented following the description in the paper in which it is presented. They have been

developed in Java, and are grouped in a framework called EPM-Framework, publicly available in a website.<sup>d</sup>

### Evaluation Setup

The objective of this study is determination of the descriptive capacity of the patterns extracted. In particular, it is necessary to know if the descriptive knowledge obtained is as general as possible in order to be simple and interpretable. Also, the knowledge

**TABLE 5** | Algorithms Used and Their Parameters Configuration for the Experimental Study

Algorithm	Parameters
DeEPs	$\alpha = 0.12$
TBJEP-C <sup>1</sup>	Ordering = hybrid; $\alpha = 0.3$ ; Pattern Max Length = -1
BCEP <sup>1</sup>	Minimum support = 0.01; Minimum growth rate = 40
iEPM <sup>1</sup>	Minimum support = 0.01; Minimum growth rate = 10; Minimum $\chi^2 = 3.84$
SJEP-C <sup>1</sup>	Minimum support = 0.01
Top-k <sup>1</sup>	$k = 20$
DGCP <sup>1</sup>	Minimum support = 0.01
LCMine	max of items = 5; growth rate = 10; subset relation = superset; maximum depth = 10; maximum variability per level = 5, 4, 3, 2
CEPMine	max of items = 5; growth rate = 10; subset relation = superset; maximum depth = 10; maximum iteration = 100
EP-RF	max of items = 5; growth rate = 10; subset relation = superset; maximum depth = 10; tree count = 100
FEPM	max of items = 5; growth rate = 10; subset relation = superset; maximum depth = 10; tree count = 100; number of labels = 4; hedges: very, somewhat, extremely, little, slightly, positively, generally.
EvAEP	number of labels = 3; number of evaluation = 10,000; population length = 100; crossover probability = 0.6; mutation probability = 0.01

<sup>1</sup>The method can only deal with nominal variables.

obtained should be as precise as possible. The final objective of EPM is the description of the underlying phenomena in data. For this aim, the measures studied are those presented in *Quality Measures* section, i.e., WRAcc, GR, Conf, TPR, FPR, number of patterns, and number of variables. It is important to remark that, following the results of the study presented in Ref<sup>34</sup>, WRAcc, GR and Conf are not correlated. However, TPR and FPR are highly correlated with GR. TPR is introduced to measure the generality of the patterns obtained, which is key in descriptive induction. On the other hand, FPR is introduced in order to measure the precision of patterns extracted, which is also a key point.

Each algorithm is evaluated over each dataset following a distributed optimally balanced stratified cross-validation<sup>68</sup> with five folds which tries to divide the dataset into folds, keeping the data distribution as similar as possible. The advantages of using this approach are:

1. Using cross-validation allows the calculation of the quality measures over test data. Descriptive knowledge in EPM should be general because it describes the underlying phenomena of the problem. Therefore, the evaluation of EPs on test data allows us to determine whether knowledge extracted is adapted to the underlying phenomena.
2. Quality measures related with precision components are calculated properly.

3. The robustness of the different kinds of EPs extracted is measured because each fold slightly changes the distribution of data.

Methods are compared with a single value for quality measure and dataset. For example, ‘BCEP algorithm on WRAcc measure gets an average value of 0.6 on appendicitis dataset.’ This allows a simple comparison between methods for each quality measure studied. This value is calculated averaging, for all mined

**TABLE 6** | Best Patterns Filters for Each Quality Measure

Algorithm	WRAcc	Conf	GR	TPR	FPR	Selected
BCEP	Conf	Conf	Conf	Conf	Conf	Conf
CEPMine	Min	Conf	Min	<b>Min</b>	<b>Conf</b>	Min
DeEPS	<b>Conf</b>	<b>Conf</b>	<b>Conf</b>	<b>Conf</b>	<b>Conf</b>	Conf
DGCP	Min	Conf	Conf	Min	Conf	Conf
EP-RF	<b>Min</b>	<b>Min</b>	<b>Min</b>	<b>Min</b>	–	Min
EvAEP	–	–	–	–	–	–
FEPM	<b>Conf</b>	<b>Conf</b>	–	Conf	<b>Conf</b>	Conf
iEPM	–	–	–	–	–	–
LCMine	<b>Min</b>	<b>Min</b>	<b>Min</b>	<b>Min</b>	Min	Min
SJEP-C	<b>Min</b>	Min	<b>Min</b>	<b>Min</b>	<b>Conf</b>	Min
Top-K	<b>Conf</b>	<b>Conf</b>	Conf	Conf	<b>Conf</b>	Conf
TBJEP-C	Conf	<b>Conf</b>	Conf	Min	<b>Conf</b>	Conf

The selected filter for each algorithm appears in the ‘Selected’ column. Conf, confidence filter; Min, MinEPs filter; and –, no filter. Results obtained with significant differences with respect to others filters employed are in bold.

patterns, the values of the quality measure using the testing datasets.

## Statistical Tests for Performance Comparison

The analysis of this study is supported by the Friedman test.<sup>71</sup> This test is a nonparametric statistical procedure for the comparison of more than two algorithms or observations. The null hypothesis is that the median of the observations are equal. The significance level considered in this study is  $\alpha = 0.05$ . In the case of significant results, it is assumed that there are

significant differences between the algorithms. These differences can be assessed by a *post-hoc* method. The *post-hoc* method used in this experiment is the Shaffer method.<sup>72</sup> After that, if two algorithms have significant differences the best of both is determined by the algorithm with a lower Friedman rank value. This procedure allows us to define safely the pairwise comparison of this kind of studies.

In the tables presented in this study, a highlighted result means that there are significant differences, i.e.,  $P\text{-value} \leq \alpha$ . For the comparison between algorithms or observations, like the one presented in Table 7, the Friedman and Shaffer

**TABLE 7** | Summary of Friedman and Shaffer Test for the Different Measures Studied

WRAcc				Conf				GR			
Algorithm	Ranking	+	±	Algorithm	Ranking	+	±	Algorithm	Ranking	+	±
FEPM	2.7500	9	11	iEPM	3.1388	7	11	iEPM	2.9861	7	11
EvAEP	3.2500	8	11	FEPM	3.1388	7	11	EvAEP	3.6666	6	11
iEPM	3.7638	6	11	Top-K	4.7777	4	11	Top-K	4.8472	4	11
TBJEP-C	5.5833	3	10	BCEP	5.6527	2	11	TBJEP-C	5.0138	4	11
CEPMine	6.1666	1	9	DGCP	5.6666	2	11	BCEP	5.44861	3	11
Top-K	6.2083	1	9	TBJEP-C	6.2638	2	9	DGCP	6.0416	2	10
BCEP	6.9166	1	8	CEPMine	6.7777	1	9	CEPMine	6.4444	1	9
DGCP	7.7916	0	8	EvAEP	6.7777	1	9	FEPM	7.3888	1	9
EP-RF	8.2777	0	8	SJEPC	8.0000	0	8	SJEPC	8.2083	0	7
LCMine	8.3333	0	7	EP-RF	8.1666	0	8	EP-RF	8.2777	0	6
SJEPC	8.6805	0	7	LCMine	9.1111	0	5	LCMine	8.7778	0	6
DeEPS	10.2777	0	4	DeEPS	10.5277	0	3	DeEPS	10.5277	0	3

TPR				FPR						
Algorithm	Ranking	+	±	Algorithm	Ranking	+	±	Algorithm	# Patterns	# Variables
EvAEP	2.0555	9	11	DeEPS	2.8055	6	11	BCEP	90.5151	4.0368
FEPM	2.6944	8	11	DGCP	2.9166	6	11	CEPMine	505.8500	3.6209
iEPM	3.5833	6	11	SJEPC	3.8472	5	11	DeEPS	2257.0444	6.6976
TBJEP-C	4.9861	4	10	Top-K	4.8055	4	11	DGCP	91.6758	<b>1.9042</b>
CEPMine	6.0000	4	9	BCEP	5.1250	4	11	EP-RF	2503.4888	5.1740
Top-K	6.1666	4	9	EP-RF	5.4722	4	11	EvAEP	<b>10.9444</b>	3.7036
BCEP	7.1527	1	8	LCMine	5.8888	4	9	FEPM	139.8722	2.8964
DGCP	7.5000	1	8	iEPM	7.0972	2	8	iEPM	40.8375	1.9499
EP-RF	8.8611	0	5	CEPMine	9.0555	0	4	LCMine	864.5555	5.0941
LCMine	8.9166	0	5	TBJEPC	9.2916	0	4	SJEPC	17043.2600	5.2436
SJEPC	9.2500	0	5	FEPM	10.1111	0	3	Top-K	40.5818	2.7690
DeEPS	10.8333	0	3	EvAEP	11.5833	0	3	TBJEP-C	367.9636	2.3565

# Patterns and # Variables show the average value of the measure. The values in bold mean the best values obtained.

tests,<sup>71,72</sup> are joined and values are aggregated in order to facilitate the analysis. The table is outlined as follows: for each algorithm, the Friedman rank is obtained and methods are sorted by this value in ascending order. After that, it is necessary to determine whether the differences obtained in the ranking are significant. Here two values are shown, the number of algorithms is significantly better than (+) and the number of methods is equal to or better than ( $\pm$ ). '+' values are calculated as the number of algorithms whose  $P$ -value in the Shaffer test are lower than  $\alpha$  and whose Friedman rankings are higher than the method analyzed. The ' $\pm$ ' values are calculated as the number of algorithms whose rankings are higher than the method analyzed or whose  $P$ -values are higher than  $\alpha$ . Therefore, one algorithm in the table is significantly better than the last '+' algorithms in the table, and it is statistically equal to the nearest (' $\pm$ ' - '+') algorithms, beginning at the position ('+' +1). More information about these statistical procedures can be found in Ref<sup>73</sup>

The results have been obtained by means of the *scmp*<sup>74</sup> package from R statistical software.<sup>75</sup>

## ANALYSIS OF RESULTS

This section presents the results of the study carried out as follows:

1. The huge search space and the amount of patterns extracted makes EPM a problem in itself from the descriptive point of view. Thus, in order to reduce the number of patterns keeping only those with high quality, some filter strategies are proposed and analyzed. These strategies are:
  - The original set of patterns.
  - The set of MinEPs. Throughout the literature, these kinds of EPs have been widely used due to their ease of mining and their discriminative power as SJEPs.
  - The set of MaxEPs. This set of patterns contains the most specific patterns. It is thus interesting to show their behavior against MinEPs, which are the opposite.
  - The set of EPs whose confidence is greater than 60%. Patterns with high confidence are important in order to obtain precise knowledge. Their study is interesting in order to show if precise patterns are also descriptive.
2. In the next phase, the best filter strategy considered previously for each method is employed in

order to determine which of the proposed approaches fits better with descriptive induction objectives. To do that, a comparison against all algorithms is carried out for each quality measure analyzed. This allows us to determine the best approach for each measure and give some directions in order to perform future research on those approaches that fit best with the objectives.

## Filtering of Patterns

The first part of the study is the selection of the best filter strategy for each method. In order to do so, the corresponding quality measures are obtained for each filter and algorithm. After that, Friedman and Shaffer tests are performed in order to choose the best filter on each quality measure analyzed. Once the best filters on each measure and algorithm are determined, the best filter for an algorithm is chosen. A filter is selected if the number of quality measures where it is the best is the highest. In the case of ties, the filter with the lowest average number of patterns and variables is chosen.

In conclusion, this study allows the determination of the best filtering strategy and drives future research on the development of methods that can mine only these kinds of patterns, so more efficient methods with better results can be developed.

Table 6 presents a summary of the analysis performed, where the best filter for each quality measure is shown. In addition, the final filter chosen appears in the 'Selected' column. 'Conf' means confidence filter and 'Min' means MinEPs filter. The sign '-' means the original set of patterns.

As can be observed, in the majority of algorithms it is necessary to filter the complete set of patterns. This is due to a search strategy that finds nonrelevant EPs in some methods. Nevertheless, the iEPM and EvAEP are methods wherein these filters are not necessary due to the search strategy performed. Future strategies can follow a hybrid approach where the mining of minimal patterns can be performed. Nevertheless, these should be focused on those minimal patterns with the highest confidence in order to obtain patterns with higher descriptive power.

## Comparison Between Algorithms

In the first part of the study the best filter strategy has been obtained for each method. After that, in this second part, a comparison of all methods with their

corresponding best filter applied is performed. The main objective of this part is to show which approach is most suitable for descriptive induction, which allows us to obtain a guideline for future research in this field. Additionally, it provides a guide of use for nonresearchers of EPs algorithms, selecting the method which best fits its necessities.

In this website<sup>e</sup> the complete Friedman test for each measure is available. The tests show significant differences for each quality measure, so it is necessary to perform the *post-hoc* procedure in order to see which pair of methods have significant differences.

Table 7 shows the results obtained for each method regarding each quality measure. This table summarizes the results of the Friedman test and the Shaffer *post-hoc* method.

The analyses of the results have been carried out for each quality measure:

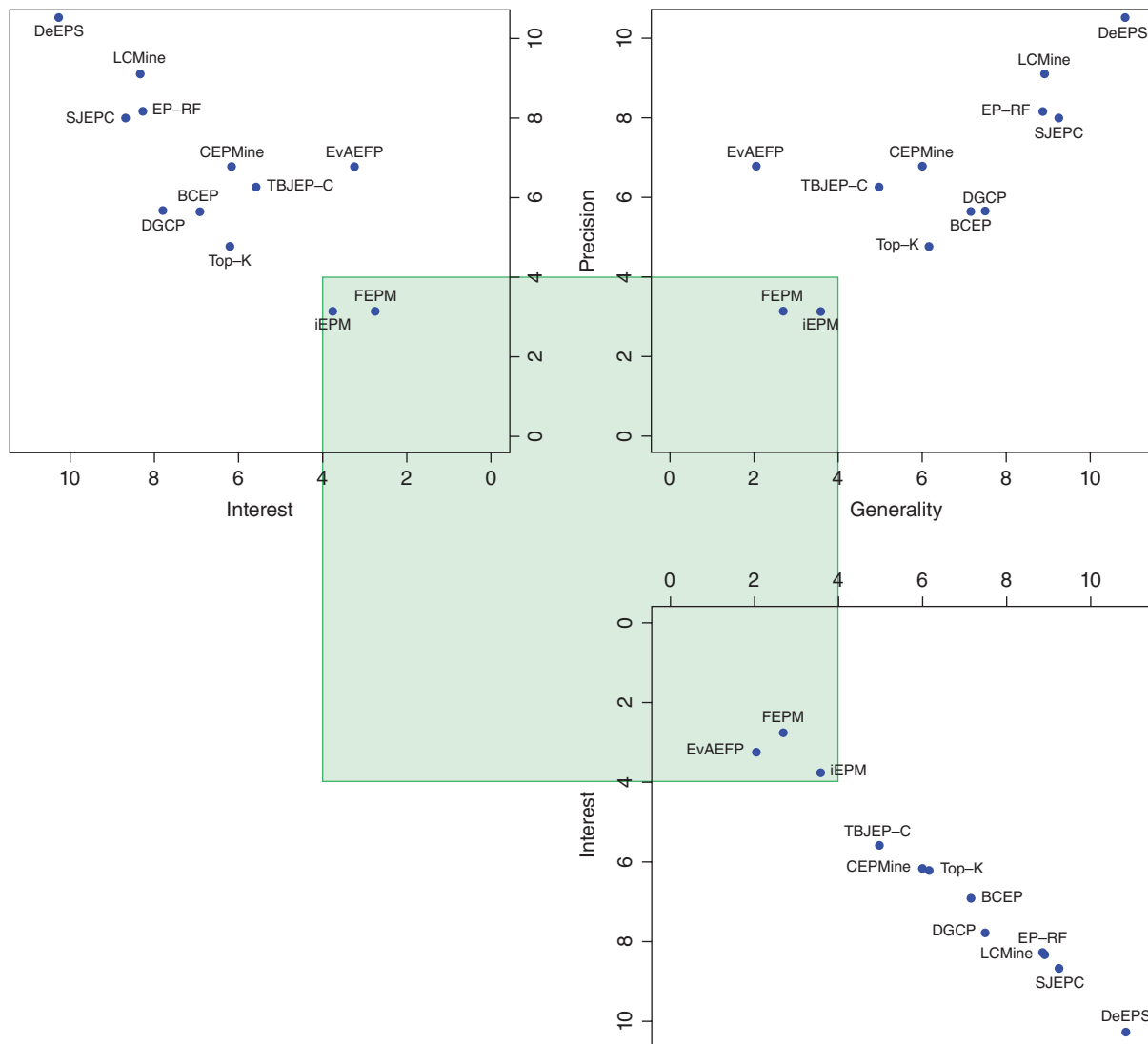
- *WRAcc*. The algorithm with the best results in this measure is the FEPM algorithm, followed by EvAEP and iEPM. FEPM is statistically better than nine methods. EvAEP and iEPM outperform eight and six methods, respectively. Moreover, they are better than or equal to the rest of the methods ( $\pm = 11$ ). Hence, the use of algorithms based on fuzzy logic is interesting in order to obtain patterns with high *WRAcc*. In addition, Chi-EPs shows a good behavior regarding this quality measure due to the use of the  $\chi^2$  test, which only takes those patterns that refuse the test, so they have relevant knowledge.
- *Confidence*. This test shows that iEPM and FEPM obtains the best confidence results with similar ranks. They are better than seven methods. The use of Chi-EPs allows obtaining of high confidence patterns due to restriction number 3 that defines this kind of EP, which obtains patterns that cover the data very well. Additionally, the fuzzy logic used in FEPM allows a better coverage of the data, so a higher confidence value is obtained. In addition, there are three tree-based methods that are equal to or better than the rest. This result shows that the representation by a tree structure is good at the acquisition of high confidence patterns. A tree tries to find a good trade-off between the support and the specificity of the pattern, which results in high confidence patterns.
- *GR*. The percentage of EPs obtained in training that are also EPs on test is measured. The best results are obtained by the iEPM algorithm, which is better than seven methods, followed by EvAEP which outperforms six. The iEPM method tends to obtain general EPs due to the support and GR restrictions that Chi-EPs have. Additionally, the patterns obtained are significant because of the  $\chi^2$  test pruning method, so these kinds of patterns generalize better than other alternatives. The evolutionary strategy of the EvAEP shows a good behavior when extracting real EPs because it prefers general EPs to specific ones, these being more suitable to be EPs in a test.
- *TPR*. The evolutionary approach of the EvAEP algorithm obtains the best results. It is better than nine methods. It is followed by FEPM and iEPM, which outperform eight and six methods, respectively. The evolutionary strategy of EvAEP stimulates the production of EPs with high coverage in order to obtain a reduced set of patterns. This is interesting because it describes the data with fewest patterns. Fuzzy logic also plays a key role. The use of LLs is well-suited to obtaining more general patterns. In addition, the Chi-EPs is a good alternative, because of restriction number 3 which tends to obtain patterns with the highest support, so the number of examples covered is increased.
- *FPR*. The DeEPS and DGCP methods obtain the best results, both being better against six methods. These methods obtain patterns that are not general, as can be observed in the TPR results, but are very specific instead. This is a normal behavior of border-based methods, such as DeEPS, which are only focused on classification. The approach of these methods is promising when patterns with a fine-grained level of description are necessary.
- *# Patterns*. In the case of the number of patterns, the EvAEP algorithm obtains the most reduced set of patterns, followed by the iEPM and Top-K methods. These results are not surprising in methods such as Top-K, where the number of patterns obtained is determined by the user. iEPMiner also obtains fewer patterns than other approaches because Chi-EPs is a subset of EPs, so there must be fewer EPs to mine. However in general, from the descriptive point of view, a reduced set of patterns allows an easier interpretation of the knowledge obtained and the evolutionary approach is a very well-suited one. Nevertheless, the tree-based approach is also a good methodology when the number of patterns is key, but the



number of patterns is normally significantly higher with respect to the results obtained with the use of the evolutionary approach of EvAEP.

- **# Variables.** The method with the fewest number of variables on average is the DGCP method followed by the iEPM algorithm. This method's results show that Chi-EPs and the mining strategy of iEPM are very well-suited when obtaining patterns with a low number of variables. The pruning strategy proposed on the DGCP algorithm performs very well when obtaining patterns with a low number of variables. Additionally, other approaches like the use of fuzzy logic in FEPM allow us to obtain patterns with a lower number of variables.

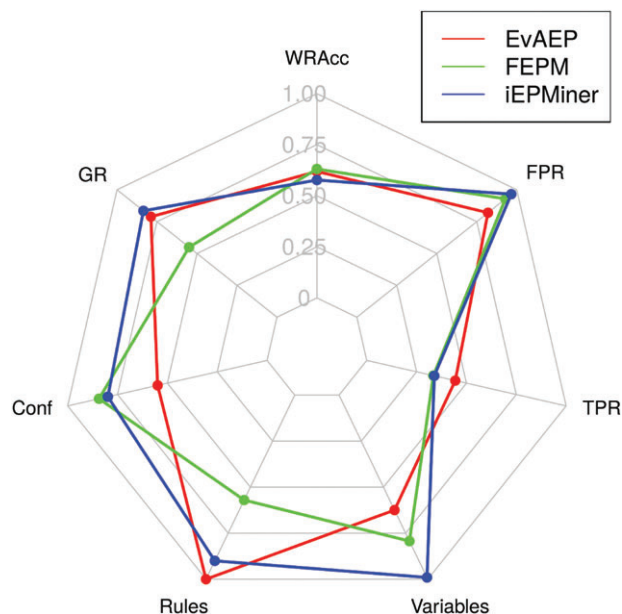
The results of the study show that Chi-EPs used in the iEPM method obtains very good descriptive patterns, with high unusualness, confidence, that generalize very well and with good TPR values. However, the number of patterns obtained is higher than with other alternatives. The evolutionary approach of EvAEP is well-suited for description and it obtains the most reduced set of patterns, but with a high FPR value. In general, throughout the study the use of FEPs for knowledge representation is good for descriptive induction. Hence, a probably promising research line could be the hybridization of these techniques, e.g., fuzzy Chi-EPs with evolutionary algorithms as a search strategy in order to obtain descriptive patterns with high quality.



**FIGURE 6** | Comparison of the average Friedman rank of the algorithms against the three relevant aspects of the supervised descriptive rule discovery (SDRD) framework: interest, generality, and precision.

From the point of view of the SDRD framework, a visual summary of the most important characteristics of SDRD<sup>11</sup>: interest, precision and generality, measured as WRAcc, Conf, and TPR respectively, is presented in Figure 6. It shows the Friedman rank of the algorithms. The green square shows the zone with a rank lower than 4. This zone shows the most relevant and interesting approaches for future research where FEPM, iEPM, and EvAEP are within this square. Hence the approaches of these algorithms are very interesting. Additionally, a more detailed analysis of these methods is presented in Figure 7, where the behavior on all quality measures is shown. In this figure, GR represents the percentage of patterns extracted that are EPs on test data, Patterns and Variables has been normalized to appear in [0,1] and the FPR value is actually 1-FPR, in order to improve the interpretability of the chart.

On the other hand, the interpretability of results is key in order to obtain valuable knowledge. EvAEP has a much lower number of patterns than iEPM and FEPM due to the fact that EvAEP was developed to fulfill SDRD objectives, and iEPM and FEPM were developed for classification. This figure also provides a guideline for future research on the approach with better results in order to improve its behavior. As an example, the approach of EvAEP is good at TPR, so evolutionary algorithms are an interesting research line to follow in order to improve this TPR value.



**FIGURE 7** | Comparison of the behavior of the three best methods against all analyzed quality measures.

Finally, the majority of the methods in the literature are focused on classification and this fact is reflected in this study. Border-based methods such as DeEPS and the first tree-based algorithms such as BCEP, SJEP-C, or Top-K obtain lower values of FPR and a high number of patterns with high number of variables, which allows the obtaining of very precise patterns that are useful in classification, but are useless for descriptive induction.

## TRENDS AND PROSPECTS

In this section some relevant aspects for future research lines with the aim of obtaining more descriptive EPs methods are shown. Following the results of the complete study, possible future research areas of study are:

- Research on more evolutionary approaches for EPM. EvAEP shows good descriptive results. Nevertheless, this method is only the first one following this approach. The methodology used can be improved in many different ways, e.g., the use of MOEAs, representation based on disjunctive normal form or the use of cooperative-competitive approaches in the evolutionary process.
- Research on ways of efficiently mining Chi-EPs. The study reflects the high descriptive potential of Chi-EPs. Thus, the creation of new models that can efficiently mine these patterns can improve the descriptive potential of EPM in several ways.
- The introduction of fuzzy logic in EPs methods allows us to obtain high levels of unusualness and a better generality of the patterns obtained. Additionally, the knowledge obtained by these kinds of patterns is more understandable for humans.<sup>76</sup> This fact, joined with a search strategy that finds a reduced set of patterns, allows the obtaining of valuable knowledge for experts. Thus, the use of fuzzy logic in EPM is nowadays a promising research line.
- The evolutionary approach has been demonstrated to be good in obtaining descriptive knowledge. However, there are other kinds of meta-heuristics like simulated annealing,<sup>77</sup> swarm intelligence,<sup>78</sup> and memetic algorithms<sup>79</sup> amongst others that can also obtain relevant results for EPM. There is an unexplored field of research which seems to be promising for obtaining descriptive EPs.

- Nowadays we generate high volumes of data, with great variety, at high speed in noisy environments, and it is necessary to maintain their veracity. This defines the 4V's model of Big Data, which is a hot topic in the fields of enterprise and academia.<sup>80,81</sup> In this review, it has been shown that EPM is complex when the amount of data is huge, namely the number of variables. However, some tree-based strategies like the EP-RF algorithm can be easily adapted to be executed in distributed environments. The descriptive point of view of EPM becomes even more important when data grow, due to it being harder to obtain a description of the data. Thus, an important research direction of EPM is on the development of methods that can deal with large amounts of data focused on description.
- This review has been focused mainly on the descriptive point of view of the task. However, it is important to remark upon the duality of EPM, where these patterns can be used as a classifier. Thus, future studies can also be focused on obtaining a good trade-off in accuracy/description in order to exploit the full capacity of EPM.
- The number of variables of the patterns extracted for each algorithm has a direct dependent with the parameters employed in their executions. In this way, there is a need to analyze the best setup for each algorithm in order to improve the trade-off between description and interpretability. A low number of variables with good values in the different quality measures are desired.
- One of the main objectives of EPM is to find emerging trends in time-stamped data. In this way, a data stream can be considered as a special type of time-stamped data where data arrives continuously in the system in sequential order. The acquisition of knowledge from these streams can improve on-demand services like

health monitoring and trading, amongst others. However, the development of EPM for data streams is a challenge. It is necessary to adapt quality measures to this sequential data. Moreover, it must be maintained incrementally in order to be efficient. The approaches presented should be adapted to data stream properties, which include infinite size, quick response, and robustness against the concept drift.<sup>82</sup> A possible first approach to deal with data streams in EPM is the use of sliding windows approaches, which allows the use of classical methods over small datasets. Nevertheless, the development of new efficient approaches is necessary with the aim of obtaining better results more efficiently.

## CONCLUSIONS

This paper presents a complete study of EPM under the SDRD framework. In fact, it presents a taxonomy of the existing algorithms present in the literature and the existing kinds of EPs. Additionally, this paper presents a complete study where the Chi-EPs, and the pruning strategy based on a  $\chi^2$  test are highlighted. It is important to remark that algorithms that use fuzzy logic show good values regarding the measures studied, with an easier interpretation of results. Evolutionary fuzzy systems obtain good results on the measures studied. The study also highlights the importance of mining those minimal patterns with the highest confidence in order to obtain patterns with higher descriptive power.

## NOTES

<sup>a</sup> Details of this experimentation appear in the website <http://simidat.ujaen.es/papers/OverviewEPM>

<sup>b</sup> <http://archive.ics.uci.edu/ml/>

<sup>c</sup> <http://www.keel.es/datasets.php>

<sup>d</sup> <http://github.com/SIMIDAT/epm-framework>—

<sup>e</sup> <http://simidat.ujaen.es/papers/OverviewEPM>

## ACKNOWLEDGMENT

This work was partially supported by the Spanish Ministry of Economy and Competitiveness under the project TIN2015-68454-R (FEDER Funds).

## REFERENCES

1. Dong GZ, Li JY. Efficient mining of emerging patterns: discovering trends and differences. In: *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, California, USA. ACM Press; 1999, 43–52.
2. 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.
3. Sherhod R, Gillet VJ, Hanser T, Judson PN, Vessey JD. Toxicological knowledge discovery by

- mining emerging patterns from toxicity data. *J Chem Inf Model* 2013, 5:9.
- Lepailleur A, Poezevara G, Bureau R. Automated detection of structural alerts (chemical fragments) in (eco) toxicology. *Comput Struct Biotechnol J* 2013, 5:1–8.
  - Angriyasa PW, Rustam Z, Sadewo W. Non-invasive intracranial pressure classification using strong jumping emerging patterns. In: *Proceedings of the 2011 International Conference on Advanced Computer Science and Information System (ICACSIS)*, Jakarta, Indonesia. IEEE; 2011, 377–380.
  - Yu Y, Yan K, Zhu X, Wang G. Detecting of PIU behaviors based on discovered generators and emerging patterns from computer-mediated interaction events. In: *Proceedings of the 15th International Conference on Web-Age Information Management*, Macau, China. Lecture Notes in Computer Science, 8485. Elsevier; 2014, 277–293.
  - Gambin T, Walczak K. A new classification method using array comparative genome hybridization data, based on the concept of limited jumping emerging patterns. *BMC Bioinformatics* 2009, 10:1.
  - Piao M, Lee HG, Sohn GY, Pok G, Ryu KH. Emerging patterns based methodology for prediction of patients with myocardial ischemia. In: *Proceedings of the 6th International Conference on Fuzzy Systems and Knowledge Discovery*, Tianjin, China. IEEE; 2009, 174–178.
  - Tzani G, Kavakiotis I, Vlahavas IP. Polya-iep: a data mining method for the effective prediction of polyadenylation sites. *Expert Syst Appl* 2011, 38: 12398–12408.
  - Li G, Law R, Vu HQ, Rong J, Zhao XR. Identifying emerging hotel preferences using emerging pattern mining technique. *Tour Manag* 2015, 46:311–321.
  - Carmona CJ, González P, del Jesus MJ, Herrera F. Overview on evolutionary subgroup discovery: analysis of the suitability and potential of the search performed by evolutionary algorithms. *WIREs: Data Min Knowl Discov* 2014, 4:87–103.
  - Atzmueller M. Subgroup discovery. *WIREs: Data Min Knowl Discov* 2015, 5:35–49.
  - Bay SD, Pazzani MJ. Detecting group differences: mining contrast sets. *WIREs: Data Min Knowl Discov* 2001, 5:213–246.
  - Dong GZ, Li JY. Mining border descriptions of emerging patterns from dataset pairs. *Knowledge Inform Syst* 2005, 8:178–202.
  - Michalski RS, Stepp R. Revealing conceptual structure in data by inductive inference. *Machine Dermatol Int* 1982, 10:173–196.
  - Asuncion A, Newman DJ. UCI Machine Learning Repository, 2007. Available at: <http://www.ics.uci.edu/~mllearn/MLRepository.html>. (Accessed August 02, 2017)
  - 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.
  - Wang Z, Fan H, Ramamohanarao K. Exploiting maximal emerging patterns for classification. In: *Proceedings of the 17th Australian Joint Conference on Artificial Intelligence*, Cairns, Australia. Lecture Notes in Computer Science, 3339. Springer; 2005, 1062–1068.
  - Cherkassky V, Mulier F. *Learning from Data. Concepts, Theory and Methods*. 2nd ed. New York: IEEE Press; 2007.
  - Box G, Jenkins G, Reinsel G. *Time Series Analysis: Forecasting and Control*. 4th ed. Oxford, UK: Wiley; 2008.
  - Zembowicz R, Zytkow JM. From contingency tables to various forms of knowledge in databases. In: *Advances in Knowledge Discovery and Data Mining*. Menlo Park, CA: AAAI/MIT Press; 1996, 329–349.
  - Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large databases. In: *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, Washington, D.C., USA. ACM Press; 1993, 207–216.
  - Dong GZ, Li JY, Zhang X. Discovering jumping emerging patterns and experiments on real datasets. In: *Proceedings of the 9th International Database Conference on Heterogeneous and Internet Databases*, Hong Kong. ACM Press; 1999, 155–168.
  - Fan H, Ramamohanarao K. An efficient single-scan algorithm for mining essential jumping emerging patterns for classification. In: *Proceedings of the 6th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Taipei, Taiwan. Springer; 2002, 456–462.
  - Fan H, Ramamohanarao K. Fast discovery and the generalization of strong jumping emerging patterns for building compact and accurate classifiers. *IEEE Trans Knowl Data Eng* 2006, 18:721–737.
  - Bailey J, Manoukian T, Ramamohanarao K. A fast algorithm for computing hypergraph transversals and its application in mining emerging patterns. In: *Proceedings of the 3th International Conference on Data Mining*, Melbourne, FL, USA. IEEE; 2003, 485–488.
  - Terlecki P, Walczak K. Jumping emerging patterns with negation in transaction databases classification and discovery. *Inform Sci* 2007, 177:5675–5690.
  - Fan H, Ramamohanarao K. Noise tolerant classification by chi emerging patterns. In: *Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Sydney, Australia. Lecture Notes in Computer Science, 3056. Springer; 2004, 201–206.
  - Ramamohanarao K, Fan H. Patterns based classifiers. *World Wide Web* 2007, 10:71–83.

30. Madden S. From databases to big data. *IEEE Internet Comput* 2012, 16:4–6.
31. Galar M, Fernández A, Barrenechea E, Bustince H, Herrera F. An overview of ensemble methods for binary classifiers in multi-class problems: experimental study on one-vs-one and one-vs-all schemes. *Pattern Recog* 2011, 44:1761–1776.
32. Tan P-N, Kumar V, Srivastava J. Selecting the right interestingness measure for association patterns. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, Alberta, Canada. ACM; 2002, 32–41.
33. Herrera F, Carmona CJ, González P, del Jesus MJ. An overview on subgroup discovery: foundations and applications. *Knowledge Inform Syst* 2011, 29:495–525.
34. García-Borroto M, Loyola-González O, Martínez-Trinidad JF, Carrasco-Ochoa JA. Evaluation of quality measures for contrast patterns by using unseen objects. *Expert Syst Appl* 2017, 83:104–113 <https://doi.org/10.1016/j.eswa.2017.04.038>.
35. Fayyad UM, Piatesky-Shapiro G, Smyth P. From data mining to knowledge discovery: an overview. In: *Advances in Knowledge Discovery and Data Mining*. Menlo Park, CA: AAAI/MIT Press; 1996, 1–34.
36. Lavrac N, Flach PA, Zupan B. Rule evaluation measures: a unifying view. In: *Proceedings of the 9th International Workshop on Inductive Logic Programming*. Lecture Notes in Computer Science, 1634. Berlin, Heidelberg: Springer; 1999, 174–185.
37. Carmona CJ, Ruiz-Rodado V, del Jesus MJ, Weber A, Grootveld M, González P, Elizondo D. A fuzzy genetic programming-based algorithm for subgroup discovery and the application to one problem of pathogenesis of acute sore throat conditions in humans. *Inform Sci* 2015, 298:180–197.
38. Gamberger D, Lavrac N. Expert-guided subgroup discovery: methodology and application. *J Artif Intell Res* 2002, 17:501–527.
39. 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.
40. Bailey J, Manoukian T, Ramamohanarao K. Fast algorithms for mining emerging patterns. In: *Principles of Data Mining and Knowledge Discovery*, vol. 2431. Berlin, Heidelberg: Springer; 2002, 187–208.
41. Li JY, Ramamohanarao K, Dong GZ. The space of jumping emerging patterns and its incremental maintenance algorithms. In: *Proceedings of the 17th International Conference on Machine Learning*. Stanford, CA: ACM; 2000, 551–558.
42. Fan H, Ramamohanarao K. A Bayesian approach to use emerging patterns for classification. In: *Proceedings of the 14th Australasian Database Conference*, vol. 17, Adelaide, Australia. Australian Computer Society, Inc; 2003, 39–48.
43. Li JY, Dong GZ, Ramamohanarao K, Wong L. DeEPs: a new instance-based lazy discovery and classification system. *Machine Learning* 2004, 54:99–124.
44. Fan H, Ramamohanarao K. Efficiently mining interesting emerging patterns. In: *Proceedings of the 4th International Conference on Web-Age Information Management*, Chengdu, China. Springer; 2003, 189–201.
45. García-Borroto M, Martínez-Trinidad JF, Carrasco-Ochoa JA, Medina-Pérez MA, Ruiz-Shulcloper J. LCMine: an efficient algorithm for mining discriminative regularities and its application in supervised classification. *Pattern Recog* 2010, 43:3025–3034.
46. Terlecki P, Walczak K. Efficient discovery of top-k minimal jumping emerging patterns. In: *Proceedings of the 6th International Conference Rough Sets and Current Trends in Computing*, Akron, OH, USA. Springer; 2008, 438–447.
47. García-Borroto M, Martínez-Trinidad JF, Carrasco-Ochoa JA. *A New Emerging Pattern Mining Algorithm and its Application in Supervised Classification*. Berlin, Heidelberg: Springer; 2010, 150–157.
48. Liu Q, Shi P, Hu Z, Zhang Y. A novel approach of mining strong jumping emerging patterns based on BSC-tree. *Int J Syst Sci* 2014, 45:598–615.
49. Wang L, Wang Y, Zhao D. Building emerging pattern (ep) random forest for recognition. In: *Proceedings of the 17th IEEE International Conference on Image Processing*, Hong Kong, China. IEEE; 2010, 1457–1460.
50. García-Borroto M, Martínez-Trinidad JF, Carrasco-Ochoa JA. Fuzzy emerging patterns for classifying hard domains. *Knowledge Inform Syst* 2011, 28:473–489.
51. García-Vico AM, Montes J, Aguilera J, Carmona CJ, del Jesus MJ. Analysing concentrating photovoltaics technology through the use of emerging pattern mining. In: *Proceedings of the 11th International Conference on Soft Computing Models in Industrial and Environmental Applications*, San Sebastian, Spain. Springer; 2016, 1–8.
52. Bayardo JR, Roberto J. Efficiently mining long patterns from databases. *SIGMOD Rec* 1998, 27:85–93.
53. Li JY, Dong GZ, Ramamohanarao K. Making use of the most expressive jumping emerging patterns for classification. *Knowledge Inform Syst* 2001, 3:131–145.
54. Han J, Pei J, Yin Y. Mining frequent patterns without candidate generation. In: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, Dallas, Texas, USA. ACM Press; 2000, 1–12.

55. Leung KS, Leung Y, So L, Yam KF. Rule learning in expert systems using genetic algorithm: 1, concepts. In: Jizuka K, ed. *Proceedings of the 2nd International Conference on Fuzzy Logic and Neural Networks*, Lizuka, Japan. Fuzzy Logic Systems Institute; 1992, 201–204.
56. Jiang X, Gruenwald L. Microarray gene expression data association rules mining based on bsc-tree and fistree. *Data Knowledge Eng* 2005, 53:3–29.
57. Quinlan J. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Series in Machine Learning. San Francisco, CA: Morgan Kaufmann; 1993.
58. Breiman L. Random forests. *Machine Learning* 2001, 45:5–32.
59. Zadeh LA. Fuzzy sets. *Inf Control* 1965, 8:338–353.
60. Zadeh LA. The concept of a linguistic variable and its applications to approximate reasoning. *Inform Sci* 1975: 199–249.
61. Hüllermeier E. Fuzzy methods in machine learning and data mining: status and prospects. *Fuzzy Set Syst* 2005, 156:387–406.
62. Goldberg DE. *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA: Addison-Wesley Longman Publishing Co, Inc; 1989.
63. Herrera F. Genetic fuzzy systems: taxonomy, current research trends and prospects. *Evol Intell* 2008, 1:27–46.
64. Wong ML, Leung KS. *Data Mining Using Grammar Based Genetic Programming and Applications*. Norwell, MA: Kluwer Academics Publishers; 2000.
65. Eiben AE, Smith JE. *Introduction to Evolutionary Computation*. Berlin, Heidelberg: Springer; 2003.
66. Deb K. *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY: John Wiley & Sons; 2001.
67. Coello CA, Lamont GL, van Veldhuizen DA. *Evolutionary Algorithms for Solving Multi-objective Problems*. *Genetic and Evolutionary Computation*. 2nd ed. Berlin, Heidelberg: Springer; 2007.
68. Moreno-Torres JG, Sáez JA, Herrera F. Study on the impact of partition-induced dataset shift on-fold cross-validation. *IEEE Trans Neural Netw Learn Syst* 2012, 23:1304–1312.
69. 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 Logic Soft Comput* 2011, 17:255–287.
70. Fayyad UM, Irani KB. Multi-interval discretization of continuous-valued attributes for classification learning. In: *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, Chambery, France. Morgan Kaufmann Publishers; 1993, 1022–1029.
71. Friedman M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J Am Stat Assoc* 1937, 32:675–701.
72. Shaffer JP. Modified sequentially rejective multiple test procedures. *J Am Stat Assoc* 1986, 81:826–831.
73. García S, Herrera F. An extension on “statistical comparisons of classifiers over multiple data sets” for all pairwise comparisons. *J Mach Learn Res* 2008, 9:2677–2694.
74. Calvo B, Santafe G. scmamp: statistical comparison of multiple algorithms in multiple problems. *The R Journal* 2016, 8:248–256.
75. R Core Team. *R: A Language and Environment for Statistical Computing*, Vienna, Austria. R Foundation for Statistical Computing; 2016 Available at: <https://www.R-project.org/>.
76. Hüllermeier E. Fuzzy sets in machine learning and data mining. *Appl Soft Comput* 2011, 11:1493–1505.
77. Suman B, Kumar P. A survey of simulated annealing as a tool for single and multiobjective optimization. *J Operat Res Soc* 2006, 57:1143–1160.
78. Bonabeau E, Dorigo M, Theraulaz G. *Swarm Intelligence: From Natural to Artificial Systems*, vol. 1. New York, NY: Oxford University Press; 1999.
79. Krasnogor N, Smith J. A tutorial for competent memetic algorithms: model, taxonomy, and design issues. *IEEE Trans Evolutionary Comput* 2005, 9:474–488.
80. Fernandez A, Río S, López V, Bawakid A, del Jesus MJ, Benítez JM, Herrera F. Big data with cloud computing: an insight on the computing environment, MapReduce and programming frameworks. *WIREs: Data Min Knowl Discov* 2014, 4:380–409.
81. Fernández A, Carmona CJ, del Jesus MJ, Herrera F. A view on fuzzy systems for big data: progress and opportunities. *Int J Comput Int Syst* 2016, 9:69–80.
82. Gama J. *Knowledge Discovery from Data Streams*. Boca Raton, FL: CRC Press; 2010.