Cognitive Computation (2019) 11:400–417
https://doi.org/10.1007/s12559-018-9612-7

A Big Data Approach for the Extraction of Fuzzy Emerging Patterns

Ángel Miguel García-Vico[1] · Pedro González[1] · Cristóbal José Carmona[1,2] · María José del Jesus[1]

Received: 11 April 2018 / Accepted: 12 November 2018 / Published online: 4 January 2019

Abstract

Nowadays, the growth of available data, known as big data, and machine learning techniques are changing our lives. The extraction of insights related to the underlying phenomena in data is key in order to improve decision-making processes. These underlying phenomena are described in emerging pattern mining by means of the description of the discriminative characteristics between the outputs of interest, which is a very important characteristic in machine learning. However, emerging pattern mining algorithms for big data environments have not been widely developed yet. This paper presents the first multi-objective evolutionary algorithm for emerging pattern mining in big data environments called BD-EFEP. BD-EFEP implements novelties for emerging pattern mining such as the MapReduce approach to improve the efficiency of the evaluation of the individuals, or the use of a token-competition-based procedure in order to boost the extraction of simple, general and reliable emerging pattern models. The experimental study performed using datasets with high number of examples shows the advantages of the algorithm proposed for the emerging pattern mining task in big data problems. Results show that the approach used by BD-EFEP opens new research lines for the extraction of high descriptive emerging patterns in big data environments.

Keywords Emerging pattern mining · Evolutionary algorithms · Fuzzy systems · Big data

Introduction

The quick progress in the development of information technologies has led to an exponential growth of the stored information due to the Internet, mobile devices, social networks, sensor networks, and so on. This kind of data is widely known as big data [97]. The traditional definition of big data was defined by Gartner [8] and it is defined as high volumes of data, arriving at high velocity and/or from a high variety of sources in the systems. These big data contain valuable knowledge for enterprises in order to improve their decision-making processes [59]. In addition, thanks to the development of cloud computing technologies [37] and machine learning frameworks and methods, big data analytics is nowadays a growing trend on enterprises [95] and academia [1].

There is a plethora of bio-inspired optimisation algorithms which are able to extract knowledge by means of the optimisation of some function of interest. Examples of this kind of methods are evolutionary algorithms (EA) [38], differential evolution [89], nature-inspired algorithms [88], amongst others. These algorithms were very successful throughout the literature [73] in many domain fields, where cognitive fields such as image recognition [17] or disease identification [2] are highlighted. This success is mainly produced due to their ability for the extraction of high-quality knowledge (the optimal solution is not guaranteed by these methods) on hard domains, in a reasonable amount of time, without the inclusion of expert knowledge within the learning process. All of these characteristics are well-suited for extraction of knowledge in big data.

The traditional machine learning algorithms were not designed to meet the requirements of these massive datasets.

✉ Ángel Miguel García-Vico
agvico@ujaen.es

Pedro González
pglez@ujaen.es

Cristóbal José Carmona
ccarmona@ujaen.es

María José del Jesus
mjjesus@ujaen.es

[1] Department of Computer Science, Interuniversity Andalusian Institute on Data Science and Computational Intelligence, University of Jaén, 23071, Jaén, Spain

[2] Leicester School of Pharmacy, De Montfort University, LE1 9BH Leicester, UK

For example, most of these algorithms assume that the whole dataset can fit in memory. Therefore, the processing of big data imposes new challenges [68]. In this way, one of the most popular processing paradigms is MapReduce [21] and their open-source frameworks Hadoop [87] and Spark [102]. Actually, there are evolutionary algorithms developed under the MapReduce paradigm for different data mining tasks such as discretisation [82], feature selection [78], association rules mining [76, 77], subgroup discovery [79] and emerging pattern mining [45], amongst others [85].

One of the most interesting capacities of machine learning methods is to distinguish between the different classes of interest in a given problem. In many domains, these capabilities must be easy to understand by the experts in order to provide a justification of their decisions [69]. Emerging pattern mining (EPM) [28, 44] is a machine learning task whose main objective is the description, by means of easy-to-understand patterns, of the discriminative characteristics between one class against the remaining ones. Using these patterns, the task is able to extract insights about the underlying phenomena in data that can be easily analysed by the experts. Therefore, the task can help in decision-making processes because experts can know about the underlying nature of their data. So a better justification of their decision can be provided. Actually, EPM have been successfully applied in several tasks such as disease detection [98], bioinformatics [70, 74] or hotel management [61], amongst others [46].

Throughout the literature, researchers have been looking for more efficient ways for extracting the most discriminative patterns. In this way, subsets of interesting emerging patterns (EPs) and methods for their efficient extraction have been proposed. Also, several approaches have been proposed [44] such as border-based [28, 62], tree-based [6, 32, 33, 64, 92], decision-tree-based [40, 42, 93] and evolutionary fuzzy systems (EFSs) [43, 46]. Amongst other approaches, the use of EFSs is a recent and successful approach for the extraction of interpretable EPM models. Despite the benefits of the extraction of EPs, to the best of our knowledge, only a few efforts have been employed in the development of scalable methods for the extraction of emerging patterns in big data [45].

This paper presents a big data approach for the extraction of fuzzy emerging patterns (BD-EFEP). It is a multi-objective evolutionary algorithm that employs a competitive-cooperative schema where individuals compete but they also cooperate in order to accurately describe the greatest possible area of the space of examples. The algorithm also uses the MapReduce approach [22] in the evaluation of the individuals in the population. This approach allows us to efficiently process big amounts of data by means of a parallel processing. It follows a global MapReduce approach, which means that the results extracted will be the same regardless the number of partitions chosen. In addition, the algorithm uses a post-processing filter based on confidence, and a competitive scheme similar to token competition (TC) [60]. These characteristics produce high descriptive patterns in big data environments for EPM. The source code of BD-EFEP is publicly available at GitHub (https://github.com/SIMIDAT/bd-efep) under the GNU General Public License.

The paper is organised as follows. Section "Background" presents the background of the main concepts used in this paper. Section "The BD-EFEP Algorithm" presents the BD-EFEP algorithm and its main characteristics. Section "Experimental Study" shows details of how the experimental study was carried out, the results of the experimental study and their analysis. Finally, "Conclusions" presents the conclusions of this work.

Background

A brief description of the main concepts related to the proposal introduced in this paper is presented in this section. First, the EPM task and its main characteristics are described in "Emerging Pattern Mining". Next, "Evolutionary Fuzzy Systems" summarises the topic of EFSs. Section "Big data" briefly introduces the MapReduce paradigm. Finally, "Evolutionary Fuzzy Systems in Big data" summarises the main developments of EFS and EAs for big data.

Emerging Pattern Mining

The emerging pattern mining (EPM) [28] is a data mining task whose main aim is to find patterns whose supports change significantly from one dataset to another or from one class with respect to the remaining ones of a single problem.

Formally, let $V = \{v_1, v_2, ..., v_n\}$ be the set of variables of the problem. Usually, one of these variables is a variable of interest, which will be noted as v_c. Let $X_i = \{x_{i1}, x_{i2}, ...x_{in}\}$ be the different categorical values or the numeric domain of the variable v_i. If there is a variable of interest v_c, we refer to the values of X_c as classes. Let $Rel = \{=, \neq, \in, \notin, >, <, \geq, \leq\}$ be a set of relational connectors. In addition, let $E = \{(v_1, x_{1j}), (v_2, x_{2j}), ..., (v_n, x_{nj})\}$ be an example. A set D of examples is defined as a dataset.

A selector [71] is defined as a triple $\left(v_i, r, x_{ij}\right)$ where $v_i \in V, r \in Rel$ and $x_{ij} \in X_i$. Let $I = \{i_1, i_2, ..., i_n\}$ be the set of all possible selectors. A logical complex (l-complex) L [71] is a type of pattern formed by conjunctions of selectors. Therefore, an example E contains the l-complex or the l-complex covers E if and only if all the relations of

the elements of L are satisfied by E. An emerging pattern (EP) P is a l-complex whose growth rate (GR) [28] is higher than a given threshold $\rho > 1$. The GR is defined as in Eq. 1.

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

where $Supp_{D_i}(P)$ is the support of the pattern P in the dataset i. These supports are calculated as $Sup_{D_i}(P) = \frac{count_{D_i}(P)}{|D_i|}$ where $count_{D_i}(P)$ is the number of examples covered by P on dataset i and $|D_i|$ is the number of examples in dataset i. It is important to remark that EPs are usually labelled in order to determine what is D_1 and D_2. In this way, if EPM is employed to find discriminative differences between classes, D_1 is the dataset formed by the examples that belongs to the class labelled in the EP. Analogously, D_2 is the dataset formed by the examples of the remaining classes.

As an illustrative example, let suppose we are looking for the discriminative characteristics between edible and poisonous mushrooms, so we have one dataset with two classes for the variable of interest. These two EPs were extracted from the Mushroom dataset available at the UCI repository [27]:

$$\begin{aligned} P_1 &= \{(Odor = none) \wedge (G.Size = broad) \wedge (Ring.Num = one)\} \rightarrow Edible \\ P_2 &= \{(Bruises = no) \wedge (G.Spacing = close) \wedge (V.col = white)\} \rightarrow Poisonous \end{aligned} \quad (2)$$

The results extracted for each pattern are presented in Table 1. As can be observed, P_1 only describe elements for the class "edible", so its GR = ∞. On the other hand, the GR of P_2 is equal to 21.4 which means that mushrooms with those characteristics are 21.4 times more likely to be poisonous than edible. In this way, both patterns are easy to understand and they describe very well the discriminative characteristics between classes.

The EPM is a descriptive task framed within the supervised descriptive rule discovery (SDRD) framework [58]. EPM can be used for different objectives, such as the description of discriminative characteristics between classes of a single dataset, the description of emerging behaviour in timestamped datasets, or the detection of differences amongst variables. This study is focused on the first objective, in order to ease the understanding of the differentiating characteristics between classes of a data set.

Different EPM algorithms have been developed up to date. These algorithms can be classified according to the approach employed for mining the EPs. Four approaches are highlighted in EPM: border-based [28, 62], tree-based [6, 32, 33, 64, 92], decision-tree-based [40, 42, 93] and EFSs [43, 46]. Details of these approaches together with an analysis of its descriptive characteristics are shown in [44]. As can be observed in these works, although EPM is framed as a descriptive task within SDRD [58], the majority of the approaches in EPM are focused on classification. This means that, despite the good capabilities for the description of the emerging behaviour or the discriminative characteristics of the dataset, most EPM algorithms try to improve the accuracy of the results, regardless of the descriptive capabilities of the extracted patterns. This fact can be observed across the literature in several reviews for the task [41, 80] where methods for mining EPs are described focused on supervised classification. Moreover, algorithms presented in [6, 32, 33, 40, 42, 64, 92, 93] are focused on classification. Finally, works presented in [65–67] are focused on classification with EPM techniques on imbalanced data. In fact, these models usually contain a high number of very specific patterns in order to obtain the best classification accuracy. Additionally, in order to perform a classification, patterns usually present dependencies amongst them in order to determine the class label of unseen examples. This makes harder to understand the underlying phenomena in data. As an example, the CAEP method [29], which is the most popular classification method for EPM, performs a prediction of the new example by means of an aggregation by class of the support of all the patterns that covers the new example. After that, it assigns the class label of the most supported class. As can be observed, all the patterns that cover the example take part within the prediction process. All these facts produce a very important effort to the experts as they need to perform an extensive analysis in order to extract some useful insights. However, EPM tries to describe emerging behaviour or discriminative characteristics, so patterns should be analysed as independent pieces of knowledge in order to achieve this aim. Therefore, knowledge in EPM should be simple, in terms of number of patterns and variables, able to describe a high number of positive

Table 1 Results obtained for P_1 and P_2 in the Mushroom dataset

EP	$Sup_{Poisonous}$	Sup_{Edible}	GR
P_1	0.000	0.639	∞
P_2	0.814	0.038	21.4

examples and with low error rate in order to provide an easy, robust way for the justification of their decisions. Therefore, it is not necessary to find patterns with the lowest error rate; patterns with a low one but simpler are desirable in order to find an easy description of data.

Quality Measures for Emerging Pattern Mining

The quality measures used in EPM are defined to quantify the interest of a pattern. However, there is no consensus about how to determine interest in the SDRD context. In data mining, interest can be defined as a concept that emphasises conciseness, coverage, reliability, peculiarity, diversity, novelty, surprisingness, utility and actionability [47]. According to this, and following the recommendations proposed in [44], measures for EPM should be focused, in most of the cases, on reliability, novelty, coverage and conciseness. All these concepts are represented by measures that can be calculated throughout a contingency table that takes into account two concepts: first, if the EP covers an example; and second, if the example belongs to the class determined by the EP.

Table 2 Contingency table of a pattern

	Class	No class
Covered	tp	fp
Not covered	fn	tn

Table 2 presents such a contingency table, where the values are tp, the number of correctly covered examples, i.e., examples covered that belong to the class determined by the EP; fp, the number of examples incorrectly covered; fn, the number of incorrectly non-covered examples and tn, the number of correctly non-covered examples.

Using this contingency table, the most used quality measures for EPM are:

– Weighted Relative Accuracy (WRAcc). It estimate the trade-off between the accuracy gain of the pattern with respect to its coverage. It is a hybrid measure that joins novelty, reliability and coverage [13]. It is computed as:

$$WRAcc(P) = \frac{tp+fp}{tp+fn+fp+tn}\left(\frac{tp}{tp+fp} - \frac{tp+fn}{tp+fn+fp+tn}\right) \tag{3}$$

The domain of this value depends on the percentage of positive class examples. Thus, it is necessary to perform a normalisation of the measure in order to perform comparisons. This normalisation is presented below [13]:

$$WRAcc\_Normalised(P) = \frac{WRAcc(P) - \left(1-\frac{Pos}{T}\right)\left(0-\frac{Pos}{T}\right)}{\frac{Pos}{T}\left(1-\frac{Pos}{T}\right) - \left(1-\frac{Pos}{T}\right)\left(0-\frac{Pos}{T}\right)} \tag{4}$$

where $Pos = tp + fn$ and $T = tp + fp + tn + fn$.

– Growth rate (GR). It defines the EPs. It computes the ratio of support between classes. It is a reliability measure, computed as [28]:

$$GR(P) = \frac{tp\,(fp+tn)}{fp\,(tp+fn)} \tag{5}$$

– Confidence (CONF). It calculates the accuracy of the pattern with respect to the examples it covers. So, it is a precision measure computed as [34]:

$$Conf(P) = \frac{tp}{tp+fp} \tag{6}$$

– True Positive Rate (TPR). It is related to coverage. It quantifies the number of correctly covered examples with respect to the total number of positive examples [56]. It is calculated as:

$$TPR(P) = \frac{tp}{tp+fn} \tag{7}$$

– False Positive Rate (FPR). It quantifies the number of incorrectly covered examples with respect to the total amount of negative examples [39]. Unlike the previous measures, the objective is to minimise the FPR value of the obtained patterns. It is computed as:

$$FPR(P) = \frac{fp}{fp+tn} \tag{8}$$

– Strength. It quantifies the relation between the GR of the pattern and its support in both classes. Therefore, a pattern with high strength indicates that the pattern is more likely to represent the real underlying phenomena [80]. It is measured as:

$$Strength(P) = \frac{\left(\frac{tp}{tp+fn}\right)^2}{\frac{tp}{tp+fn} + \frac{fp}{fp+tn}} \tag{9}$$

– Chi-squared (χ^2). The value of this statistical test measures as null hypothesis the non-existence of significant

Table 3 Expected contingency table for measuring the χ^2 measure

	Class	No class
Covered	$\frac{(tp+fn)(tp+fp)}{T}$	$\frac{(fp+tn)(tp+fn)}{T}$
Not covered	$\frac{(tp+fn)(fn+tn)}{T}$	$\frac{(fp+tn)(fn+tn)}{T}$

differences between the proportions of positive and negative, covered and not-covered examples. In this way, a significant value, i.e., greater than 0.95, means that there is significant differences between these elements and, therefore, the pattern is interesting. The value is computed as follows [7]:

$$\chi^2(P) = \sum_{i=1}^{2}\sum_{j=1}^{2}\frac{\left(O_{ij} - E_{ij}\right)^2}{E_{ij}} \tag{10}$$

where O is the contingency matrix presented in Table 2, and E is the expected contingency table. This table is calculated as in Table 3. Using $CDF\left(\chi^2(P), n_c - 1\right)$ we calculate the significance value, where CDF is the cumulative distribution function of the χ^2 distribution with $n_c - 1$ degrees of freedom, where n_c means the number of classes.

Evolutionary Fuzzy Systems

An EFS [49] is a hybridisation of fuzzy systems [99] augmented with a learning process based in an evolutionary algorithm (EA) [53]. Fuzzy systems are one of the most important areas for the use of fuzzy sets theory [100], characterised by their ability to handle imprecision and uncertainty, and to describe the behaviour of complex systems. The most common fuzzy systems consist of a collection of logical fuzzy rules, or patterns, and are known as fuzzy rule-based systems (FRBSs). It is important to remark that rules and patterns are similar concepts, so rule-based systems can be used for the extraction of patterns. This kind of systems has been widely applied in finance, control, engineering and medicine [3, 5, 15, 46, 51, 75, 86], amongst others, as they provide a comprehensible representation of the knowledge extracted and a good approach for handling continuous variables. The use of fuzzy linguistic labels [99] for the representation of numeric variables is easier to understand than a discretised representation [52] and it prevents the loss of information produced in a discretisation process.

On the other hand, evolutionary computation, including EAs [48, 50], genetic programming [57] and evolutionary programming [38], amongst others, does not only contribute their ability to deal with large search spaces and to find near-optimal solutions without a precise description of the problem, but is also able to incorporate knowledge into the search process. For example, the incorporation of knowledge in FRBSs can be performed through the parameters of the fuzzy membership functions, in the form of linguistic variables, in the way individuals are codified or in the definitions of the genetic operators.

EFSs have been widely used for other SDRD tasks such as subgroup discovery [11, 15, 55]. The models extracted were very helpful for experts in many real-world fields such as medicine [9, 12, 15], web usage mining [14] or photovoltaic technology [10], amongst others. In fact, the first SDRD model for big data environments was developed for subgroup discovery: the MEFASD-BD algorithm [79], which is based on a local MapReduce approach where the NMEEF-SD algorithm [11] is executed in the map phase and the reduce phase produce the final result set by means of token competition [60].

Within the learning procedure of an EFS, two strategies could be used for encoding the individuals of the population [19]:

- "Chromosome = set of rules". Also known as Pittsburgh approach, in which each individual represents a set of rules [26].
- "Chromosome = rule". An individual represents a single rule, and the complete rule set is obtained by the combination of several individuals. These can be combined using three generic approaches: Michigan [16], iterative rule learning (IRL) [20] and the "cooperative-competitive" approach [96].

Currently, there are two proposals developed using EFSs to address the EPM task: the EvAEP algorithm [46], and the MOEA-EFEP algorithm [43]. The former is an EFS method based on a mono-objective EA which follows the "chromosome = rule" approach within an IRL process for the extraction of the best pattern at the end of the evolutionary process. The stop criterion of the iterative process is that the pattern extracted is not an EP, or it does not cover new examples by any of the patterns extracted so far. A MapReduce adaptation of this mono-objective algorithm, called EvAEFP-Spark [45], has been developed in which the evaluation of the individuals is performed in parallel and the evolutionary process is carried out in the master node. MOEA-EFEP is a multi-objective EA that follows the "chromosome = rule" approach based on a cooperative-competitive schema. This population cooperates in order the get global optima due to the use of an elite population with the best WRAcc and competes by means of the use of the token competition procedure.

Big data

We are living in the information era because of the increasing and the improvement of technologies. We are

surrounded by devices that constantly generate data. These data contain valuable information that can be employed for the improvement of everyday lives. These huge amounts of data are commonly called as big data. The term big data was defined by Gartner [8] as big volumes of data, arriving at high velocity from a variety of sources. These characteristics imposes us several difficulties for the extraction of knowledge. Firstly, traditional data mining algorithms are not able to handle big data as they usually assume the whole dataset fits in main memory. Secondly, many algorithms cannot extract knowledge fast enough to process data as it arrives. Finally, the variety of sources produces a huge heterogeneity of formats that must be normalised.

MapReduce [22, 23] is one of the most popular programming paradigms to deal with big data, mainly with the volume of data and the velocity of arrival. It is a functional, distributed programming framework following the divide-and-conquer paradigm for processing massive amounts of data. One of its main advantages is that algorithms can be easily executed under distributed computing centres as it contains all necessary communications and parallelisation mechanisms. Actually, one of the main reasons of the success of MapReduce is due to all the mechanisms that make possible the parallel computation of tasks, such as distribution of jobs and data, replication mechanisms, hardware and software failure treatments, and so on, are completely transparent to the programmer. So they can focus on programming their algorithms.

Basically, MapReduce contains two primary functions: the map and the reduce. These functions must be designed by the user. In a nutshell, the map function processes the data in parallel, extracting some intermediate results. After that, the reduce function aggregates these results in order to produce the final output. An extended definition of each of these phases is described below [81]:

- Map phase. The master node splits the input data into several partitions. Each of this partition is identified by a key-value pair ($< k, v >$) and it is sent to each node in the cluster maximising the data locality in order to minimise data transfer. After that, the processing of each pair is carried out concurrently on the nodes of the cluster. During this processing, the input pair $< k, v >$ is processed by the map function developed by the user. The result of this function is another $< k', v' >$ pair with intermediate data. After that, these pairs are shuffled and ordered by key and will be the input to the reduce function.
- Reduce phase. This function is the responsible of aggregating the outputs of the map function. Once all maps have finished, they are sent to the reducers where the key-value pairs are sorted and merged by key. After that, the reduce function, developed by the user, is executed for every k' key, where all the values are aggregated. Therefore, the reduce function returns a new ($< k', v'' >$) pair with the final output for each key.

One of the main drawbacks of the MapReduce paradigm is its performance regarding to iterative jobs [63] because of the overload produced from reloading the whole job from disk. In this way, alternative solutions have been developed in order to avoid this issue. Nowadays, one of the most popular frameworks that implements the MapReduce paradigm is Apache Spark [101]. Spark uses a structure called resilient distributed datasets (RDD) to keep data objects in main memory, together with transformations and actions that are performed over the RDD in parallel. The main advantage of RDDs is the ability to persist the intermediate results produced across several map processes in main memory, so iterative algorithms can be carried out avoiding the re-execution of the whole MapReduce process. In this way, as BD-EFEP is an EA, which is mainly an iterative process, the Spark framework is employed for the development of the method.

Evolutionary Fuzzy Systems in Big data

The advantages of fuzzy systems, in particular FRBSs, in big data are relevant for the community because of its robustness against scalability issues [35, 36]. Across the literature, two main approaches for the development of big data algorithms have been proposed:

- A local approach. It is based on the execution within the map phase of a baseline FRBS, and then extract insights according to the data of each map. After that, the reduce phase removes redundant or non-relevant knowledge.
- A global approach. The whole algorithm, or one of its most computationally expensive tasks can be executed in parallel because of their nature. This kind of algorithms are characterised because of their ability for the extraction of the same results independently of the number of maps used in the process.

Throughout the literature we can find applications of FRBSs in big data for several data mining tasks. In [78], it is presented an EA for feature selection in big data based on the CHC algorithm [31] using a local approach. This method executes CHC on each partition within the map phase, where a binary vector of selected features is returned. Then, the reduce phase averages the number of times the feature is selected on each map. For the classification task in big data, the FRBS presented in [83] was the first approach for the task. It is based on a local approach. It executes the Chi et al. [18] algorithm on each partition, extracting fuzzy rules for the data of the partition. The reduce phase collects all rules and modify the weights accordingly. Recently, in [30] a global approach of the Chi et al. method is presented where the quality of the classification is improved with

respect to the local approach and the average runtime is improved as well. In [84], a genetic algorithm for the extraction of Takagi-Sugeno-Kang (TSK) rules [90] based on a local approach is presented for the regression task. This method performs a previous fuzzyfication of the variables, and then it executes the F-RULER algorithm [84] on each partition, where TSK rules on each partition are extracted. Then the reduce phase aggregates all the rules extracted in order to obtain the final rule base. In [77] a genetic programming algorithm following a global approach is proposed for the extraction of rare association rules. In this method, the evolutionary process is sequentially executed until the evaluation of the individuals. In this part, the map phase calculates the support of each individual on each partition. Then, the reduce phase aggregates all the supports of the individual. For the subgroup discovery task, [79] presents an EA based on a local approach for the extraction of subgroups. The method executes in the map phase the NMEEF-SD algorithm [11], where subgroups are extracted from each partition. Then, on the reduce phase the final subgroups are aggregated by means of a token competition procedure [60]. Finally, for EPM, the EvAEFP-Spark algorithm is presented in [45]. This method is an EA following a global approach where only the evaluation of the individuals is executed in parallel on each partition. After that, all values calculated on each map are aggregated in the reduce phase.

As can be observed, the majority of the proposed FRBS for big data follows a local approach. Although local approaches are usually much faster than global ones because of they tend to have less MapReduce processes than global approaches, local approaches are more likely to suffer from data-division problems such as the increasing of small disjuncts, skewed class distribution, lack of training data or the extraction of less accurate models [36]. In exchange for avoiding these problems, global approaches usually spend a much higher amount of time due to the larger amount of MapReduce procedures launched throughout their mining processes. Nevertheless, cloud computing technologies allow us an easy deployment of very large clusters where the trade-off between quality and execution time would be better for global approaches.

The BD-EFEP Algorithm

BD-EFEP is the first MOEA focused on EPM for big data problems. It contains specific operators in order to deal with the extraction of high descriptive EPs. The main objective of BD-EFEP is the extraction of patterns with high discriminative capacity in order to describe the underlying phenomena of a problem. This knowledge should be as comprehensible as possible to be useful for the experts. These restrictions means that the patterns extracted should be general, simple and with a proper level of reliability. Therefore, the extraction of EPs in BD-EFEP can be defined as a multi-objective problem. In this way, a multi-objective evolutionary algorithm (MOEA) is a well-suited approach for this kind of problems. So BD-EFEP uses a MOEA approach in order to extract all the patterns in the Pareto front [24] in order to get good balance in the quality measures analysed.

The BD-EFEP algorithm proposes specific operators oriented towards the extraction of highly descriptive EPs. These operators try to find general, simple and reliable fuzzy or crisp patterns, depending on the type of variables of the problem. It also provides mechanisms to extract knowledge from big data environments within an affordable time. This is a key advantage of the proposal because the extraction of EPs is a very hard problem when the volume of data is huge, in particular with respect to the number of variables [94]. Despite the relevance of the knowledge extracted by EPM algorithms, the research of mining methods focused on big data environments has not been widely explored. Actually, to the best of our knowledge, it only exists another algorithm that can extract EPs from big data environments, the EvAEFP-Spark algorithm [45].

The representation of EPs can be easily codified within rule-based systems. It is important to remark that in this context we can use the term pattern or rule interchangeably. The representation employed within these systems is very close to the way the experts extracts and analyse this kind os knowledge. Therefore, rule-based systems are very suitable for the extraction of EPs. Moreover, as patterns extracted in EPM are considered as independent pieces of knowledge, BD-EFEP employs a "chromosome = rule" representation where both the antecedent and consequent parts of the pattern are included in each individual of the population. This representation is focused on the optimisation of each potential pattern as independent solutions, so the approach is well-suited for the task. Moreover, the representation of the consequent part within the chromosome allows the extraction of patterns for all the classes in a single execution of the algorithm. This is an advantage with respect to other EPM algorithms throughout the literature, that must be executed once for each class.

BD-EFEP uses a TC-based competitive scheme. It produces a competition between chromosomes in order to keep the best ones. In this procedure are considered the trade-off between generality and reliability of each individual, together with the description of knowledge not described by stronger individuals. In fact, the objective of the TC-based procedure is to keep the minimum set of patterns that can describe all the data. The novelty of this procedure is that patterns are kept according to their coverage and not by their GR, which it normally leads to the extraction of very specific patterns.

The generalisation of the individuals is promoted in the evolutionary process by the use of mutation and oriented initialisation operators that promote the generation of high coverage patterns.

Diversity is promoted by the crowding-distance-based niching technique of the NSGA-II algorithm [25] and a reset procedure when the population reaches a local optimum.

Finally, the precision of the patterns is improved by the use of a confidence-based post-processing filter. The main idea of this component used for EPM is to find a trade-off between the generality and precision of the patterns obtained.

The main features of the BD-EFEP algorithm with respect to the classical EPM methods are described below:

- It is a MOEA-based algorithm that allows the optimization of different interest measures at the same time.
- The generality of patterns is improved by means of mutation and oriented initialisation operators. They produce patterns with high generality also promoting them along the execution of the evolutionary process. This produces a more general pattern model with less overfitting.
- A reset mechanism is applied in order to prevent falling into local optima.
- A competitive approach by means of a TC-based procedure is applied in order to keep high quality patterns with high coverage level. This procedure allows the extraction of a reduced set of patterns with high coverage and high descriptive capacity.
- The use of a MapReduce approach in the evaluation of the individuals in order to be more efficient in big data environments. The evaluation procedure is the most expensive element of BD-EFEP.

In the following subsections, the details of the BD-EFEP algorithm are presented. First, the pattern representation proposed is described in "Pattern Representation". Next, the details of the operational scheme of the algorithm are shown in "Operational Scheme". A brief description of the genetic operators is presented in "Genetic Operators", the reset operator is shown in "Reset Procedure" and the TC-based procedure is outlined in "Token-Competition-Based Procedure". Finally, the MapReduce approach used is shown in "Evaluation with MapReduce".

Pattern Representation

The patterns obtained by BD-EFEP use fuzzy logic to represent continuous variables. These are represented by means of linguistic labels (LLs), allowing the use of datasets without a previous discretisation of the continuous variables. In addition, the knowledge obtained when using LLs is more interpretable than that of other representations [52]. The continuous variables are therefore considered as linguistic variables using a set of LLs. Each fuzzy set corresponding to each LL can be specified by the user or by means of an uniform partition with triangular membership functions. The latter option is useful when expert knowledge is not available.

The BD-EFEP algorithm represents the patterns in canonical form, where the antecedent part is formed by conjunctions of variable-value pairs and the consequent part is a class, that is, a value of the target variable.

In Fig. 1 an example of a chromosome in BD-EFEP is shown. A chromosome is represented as an integer vector whose length is equal to the number of variables of the problem, including the target variable. For the antecedent part of the pattern, these values represent the ith value of a discrete variable, or the ith LL in case of a numeric variable. It is important to remark that a value of zero means that the variable does not participate in the antecedent part of the pattern. For the consequent part of the pattern, the integer value used codifies the class, i.e. the value of the discrete target variable.

This representation makes it possible to obtain patterns for all the classes of the variable of interest in a single execution due to the introduction of the class value in the representation of the chromosome. This is even more relevant in big data environments, where the execution of an algorithm has a high cost.

The evaluation of each chromosome follows a MapReduce approach,explained in detail in Section 2. In a nutshell, the quality measures associated to a chromosome are calculated by means of its contingency table. This table is filled according to whether the underlying pattern that represents the chromosome covers a given example or not. Considering:

- $\{X_m/m = 1, ..., n_v\}$ a set of variables that can be categorical or numeric, where n_v is the number of variables,
- $\{Class_j/j = 1, ..., n_c\}$ the set of values of the target variable, where n_c is the number of classes, and
- $\{E^k = \left(e_1^k, e_2^k, ..., e_{n_v}^k, Class_j^k\right)/k = 1, ..., n_{ex}\}$ a set of examples, where $Class_j^k$ is the value of the target

$$
\begin{array}{c}
Genotype \\
\begin{array}{|c|c|c|c|c|}
\hline
X_1 & X_2 & X_3 & X_4 & Class \\
\hline
1 & \emptyset & 3 & \emptyset & 1 \\
\hline
\end{array} \\
\Downarrow \\
Phenotype \\
(X_1 \in Low) \wedge (X_3 = Sports) \rightarrow (Class = Positive)
\end{array}
$$

Fig. 1 Representation of a fuzzy canonical pattern with continuous and categorical variables in BD-EFEP

variable for the example E^k and n_{ex} is the number of examples of the problem,

an example E^k is correctly covered by a pattern R_i if and only if:

$$APC(E^k, R_i) > 0 \wedge Class_j^k = Class_j \tag{11}$$

where the Antecedent Part Compatibility (APC) value indicates the degree of compatibility of the examples with respect to the antecedent part of the pattern. Therefore, E^k is covered by the pattern P_i if E^k has a membership degree greater than zero in the fuzzy subspace delimited by the antecedent part of the pattern. This value is calculated as in Eq. 12,

$$APC(E^k, P_i) = T\left(bd_1\left(e_1^k, LL_1^j\right), ..., bd_{n_v}\left(e_{n_v}^k, LL_{n_v}^j\right)\right) \tag{12}$$

where:

- $LL_{n_v}^j$ is the LL number j of the variable n_v. This is the LL assigned to the pattern represented by the individual.
- T is the selected *t-norm* to represent the fuzzy AND operator, i.e., the fuzzy intersection. In this case, the minimum t-norm is selected.
- $bd_{n_v}\left(e_{n_v}^k, LL_{n_v}^j\right)$ is a function that assigns the belonging degree of the example to $LL_{n_v}^j$ by means of the application of the maximum *t-conorm*. If this *t-conorm*

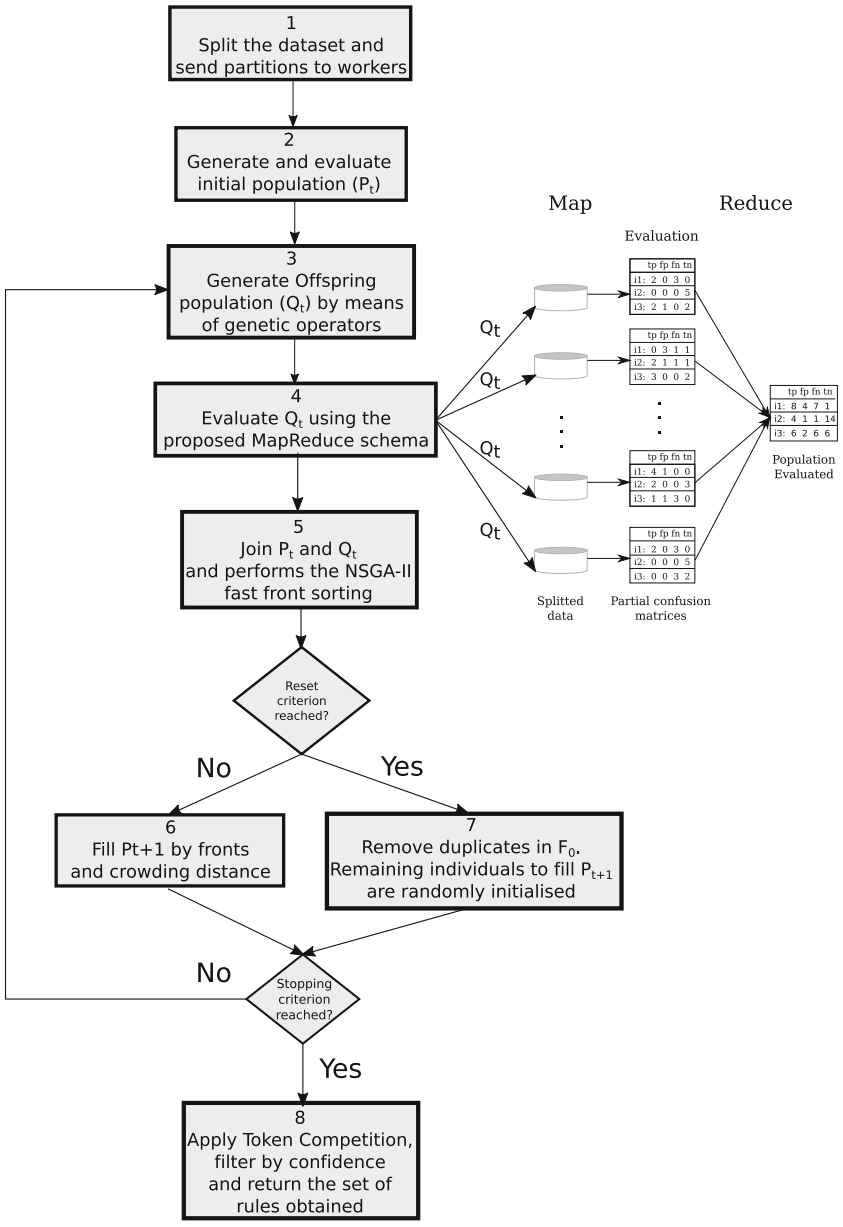

Fig. 2 Operational scheme of the BD-EFEP algorithm

matches the belonging degree of $LL_{n_v}^j$, then this value is returned, otherwise, a value of zero is returned. More details are presented in Eq. 13.

$$bd(e_i^k, LL_i^j) = \begin{cases} \mu_{LL_i^j}\left(e_i^k\right), & If\ max\{\mu_{LL_i^1}\left(e_i^k\right), ..., \mu_{LL_i^{l_{n_v}}}\left(e_i^k\right)\} = \mu_{LL_i^j}\left(e_i^k\right) \\ 0, & another\ case \end{cases} \tag{13}$$

where $\mu_{LL_i^j}\left(e_i^k\right)$ is the degree of membership of the example e^k in the LL number j for the variable i. For categorical variables, this value is one if $e_i^k = X_i$ or zero otherwise.

Operational Scheme

The operational scheme of BD-EFEP is shown in Fig. 2. In addition, pseudo-codes of the map and the reduce functions are shown in Algorithm 1 and Algorithm 2 respectively. The algorithm works as follows:

1. Following the MapReduce approach, data are split into partitions which are sent to the worker nodes in the cluster.
2. In the driver or main node, the EA starts with the generation of the initial population (P_t) by means of the oriented initialisation operator.
3. The offspring population (Q_t) is generated by means of the application of the genetic operators, where the size of Q_t is the same as P_t.
4. When Q_t is generated, the population is evaluated following the MapReduce approach proposed in "Evaluation with MapReduce".
5. P_t and Q_t are joined into a new population R_t. Then, the fast sorting procedure based in the crowding distance of NSGA-II is applied over R_t. This operator sorts the individuals according to their dominance level in different fronts (F_i), i.e., the individuals in the Pareto front (F_0) are not dominated by any other individual, F_1 contains individuals dominated by only one individual, and so on.
6. If the reset criterion is not reached, the population for the next generation (P_{t+1}) is filled by introducing the different fronts in order. If the number of individuals in F_i is bigger than the number of the remaining individuals to be introduced in P_{t+1}, the chromosomes of F_i are sorted by crowding distance and then introduced in order until P_{t+1} is filled.
7. If the reset criterion is reached, the reset procedure is applied.
8. Finally, the stop criterion is checked. The algorithm stops when the specified number of evaluations are reached. Once the evolutionary process has finished, a TC-based procedure is applied in P_t in order to remove redundant patterns. After that, the confidence-based filter is applied to the resulting population and the final pattern set obtained is returned. It is important to remark that this filter keeps at least one pattern per class, even if it does not reach the confidence threshold. This allows to get patterns describing all the classes of the target variable.

Algorithm 1 Map function of BD-EFEP

Require: Q_t a population of individuals.
$tp_{ji}, tn_{ji}, fp_{ji}, fn_{ji} \leftarrow 0$
for all examples E in the map and individuals R in Q_t **do**
if $APC(E_k, R_i) > 0$ **then**
if $Class(E_k) == Class(R_i)$ **then**
$tp_{ji} \leftarrow tp_{ji} + 1$
else
$fp_{ji} \leftarrow fp_{ji} + 1$
end if
else
if $Class(E_k) == Class(R_i)$ **then**
$fn_{ji} \leftarrow fn_{ji} + 1$
else
$tn_{ji} \leftarrow tn_{ji} + 1$
end if
end if
end for
return $tp_{ji}, tn_{ji}, fp_{ji}, fn_{ji}$

Algorithm 2 Reduce function of BD-EFEP

Require: M_p. A set of contingency tables returned from the Map function.
$tp_i, tn_i, fp_i, fn_i \leftarrow 0$
for $j = 1$ **to** p **do**
for $i = 1$ **to** $NumIndividuals$ **do**
$tp_i \leftarrow tp_i + tp_{M_{ji}}$
$fp_i \leftarrow fp_i + fp_{M_{ji}}$
$tn_i \leftarrow tn_i + tn_{M_{ji}}$
$fn_i \leftarrow fn_i + fn_{M_{ji}}$
end for
end for
return tp_i, tn_i, fp_i, fn_i

Genetic Operators

The genetic operators used in the BD-EFEP algorithm are described below:

- Oriented initialisation operator. According to the number of individuals specified by the user, 75% of such individuals are generated with at most 25% of variables randomly initialised. The remaining 25% of the individuals are randomly generated. This is done in order to provide a set of rules with high diversity and generality at the begining of the evoluationary process, which is key.
- Binary tournament selection [72]. Two individuals of the population are randomly selected. The one with the best crowding distance is added in the offspring population. The use of crowding distance is for the improvement of the diversity in the population. In addition, this operator is selected because it produces the least selective pressure in order to avoid a premature convergence.
- Multi-point crossover [50]. Two parents with the same class and two random points of these individuals are selected. The chunk amongst these two points is interchanged. The result is added into the offspring population. This operator have been widely employed across the literature because of its good exploitation capabilities.
- Oriented mutation operator. It removes a variable that already participates in the pattern or otherwise it sets a random value for that variable. Each possibility can be applied with the same probability. It is key as it produces high diversity and it keeps the individuals in the population with a low number of variables, which is one of the main aims of the algorithm.

Reset Procedure

BD-EFEP uses a reset procedure in order to avoid falling into local maxima. This procedure is applied when the reset criterion is reached. This criterion is triggered when the population does not evolve for at least a 10% of the total evaluations. The population does not evolve if it does not cover examples not covered up to date.

The reinitialisation procedure first introduces those non-duplicated individuals of F_0 in the new population of the next generation (P_{t+1}). This allows us to keep the best individuals found up to the moment. After that, the remaining individuals in P_{t+1} are randomly generated by means of the oriented initialisation procedure in order to find another promising area of the search space.

Token-Competition-Based Procedure

BD-EFEP uses a competitive schema where individuals must compete amongst them in order to survive. In this way, BD-EFEP employs a TC-based procedure [60] for the extraction of a set of patterns that improves the diversity, in terms of coverage of the examples of the dataset, and the reduction of redundant patterns. The TC-based procedure proposed in BD-EFEP works pretty similar to the original one. Let suppose that for each example E^k, we have a token T_k that can be *true* or *false*. Initially, all T_k are set to *true*. In addition, we have a counter $C_j = 0$ for each individual I_j in the population. The TC-based procedure of BD-EFEP initially sorts the population by WRAcc. This promotes the extraction of patterns with a high generality-reliability trade-off, as well as patterns with high GR as these measures are related [13]. Then, for each individual I_j, the tokens of the correctly covered examples, i.e., the covered examples that belongs to the class determined by the pattern, are set to *false* if and only if $T_j = true$. Then, C_j is updated with the number of tokens set to *false* by the pattern. This procedure is repeated for all individuals in the population. Finally, those I_j with $C_j = 0$ are removed from the population.

Evaluation with MapReduce

The most expensive task in BD-EFEP is the evaluation of the individuals. This is because the quality measures used as objectives in BD-EFEP are based in the values provided by a contingency table. This table counts the number of examples correctly/incorrectly covered/non-covered by a given pattern in the whole dataset. An example of such table is shown in Table 2. Therefore, it is necessary to traverse the dataset once for each non-evaluated individual for each generation of the evolutionary process in order to calculate the objective measures. In big data environments, to traverse the whole dataset implies a huge computational cost.

Therefore, the evaluation procedure has been developed following a MapReduce approach in order to improve the efficiency. The process is graphically presented as part of Fig. 2 within step four, and pseudo-codes of these procedures are presented in Algorithm 1 and 2 respectively. The description of the map and reduce phases is presented below:

- Map. On each generation of the evolutionary process, the map phase sends i non-evaluated individuals to each partition. Let the number of partitions be k. Then, each worker node generates a set of partial contingency tables M_{ki} according to the data it owns.
- Reduce. On the reduce phase the complete contingency table M_i for an individual i is created. It collects all the sets of contingency tables from the worker nodes and calculates $M_i = \sum_{j=1}^{k} M_{ji}$. After that, the quality measures used as objectives are calculated for each individual and returned to the master node in order to continue with the evolutionary process.

As stated in [54], the NSGA-II approach does not perform well when the number of objectives is high.

Therefore, it is necessary to choose a reduced set of objectives that should improve the whole set of quality measures used to determine the descriptive quality in EPM. The objectives used in BD-EFEP are the FPR, described in Eq. 8 and the Jaccard index (Jacc) [91], calculated as in Eq. 14.

$$Jacc(R) = \frac{tp}{tp + fp + fn} \tag{14}$$

Jacc measures the similarity between two sets. In BD-EFEP, these sets are the set of examples that belongs to the class determined by the pattern and the set of examples covered by the pattern respectively. The use of this objective allows to get patterns with a good trade-off between the generality and reliability of the pattern. In addition, it is the measure that presents the best behaviour for imbalanced datasets, independently of the imbalance ratio [65]. The use FPR as an objective boost the extraction of patterns with a great balance between coverage and precision. BD-EFEP contains several mechanisms to improve coverage. Therefore, the use of FPR, which is a reliability measure, allows the algorithm to keep those reliable patterns within the evolutionary process.

Experimental Study

In this section, the experimental study performed is presented. The main aim is the determination of the quality of the BD-EFEP algorithm with respect to other approaches for the extraction of EPs within big data environments. To do so, the experimental framework is first presented, where the details of the study carried out are described. Next, a comparison of the quality of the results of the different algorithms is carried out. Finally, a scalability and time comparison for the methods used is presented.

Experimental Framework

The experimental framework used to evaluate the quality of BD-EFEP is outlined in this section. First, the datasets used in the experimental study and their main characteristics are presented. Next, the algorithms used and their configuration are outlined. Then, the quality measures used to analyse the quality of the results are shown. Finally, the run environment for the executions of the experiments is shown.

– Datasets. The study was carried out using a set of 6 well-known large-scale real datasets from the UCI repository [4]. The properties of these datasets are presented in Table 4, where the number of examples (# Instances), the number of variables (# Variables), separated in real, integer and nominal (R/I/N), the size of the datasets in gigabytes (GB), and the number of classes (# Classes) are shown.

Table 4 Properties of the datasets used in the experiments

Name	# Instances	# Variables (R/I/N)	Size (GB)	# Classes
Census	299284	41 (1/12/28)	0.151	3
kddcup	494020	41 (26/0/15)	0.049	23
rlcp	5749132	11 (11/0/0)	0.452	2
susy	5000000	18 (18/0/0)	1.503	2
higgs	11000000	28 (28/0/0)	4.772	2
hepmass	10500000	29 (29/0/0)	4.886	2

– Algorithm and parameters. The algorithms employed in this experimental study, together with their parameters configuration, are presented in Table 5. Classical EPM algorithms were not able to extract knowledge from these big datasets. Only those algorithms developed to deal with big data environments, EvAEFP-Spark [45] and the algorithm introduced in this paper, BD-EFEP, could be applied to these datasets. This is the reason why this comparison only includes these two algorithms. The parameters used for EvAEFP-Spark has been taken from [45]. The parameters used for BD-EFEP where chosen in order to perform a comparison as fair as possible with EvAEFP-Spark. This is the reason for choosing 10000 evaluations for the stop condition and three LLs for each fuzzy variable. The remaining parameters were selected following the recommendations of previous SDRD works based on the NSGA-II approach such as NMEEF-SD [11] and MOEA-EFEP [43]. Actually, a low population size allows us a fast execution, while the use of a high mutation probability allows us a better exploration of other areas of the search space that could be interesting.
– Quality measures. The quality measures used to determine the quality of the knowledge extracted by the algorithms are those presented in "Quality Measures for Emerging Pattern Mining" due to they are the most common quality measures used in EPM to determine the quality of the descriptions obtained. In addition, the average number of patterns (n_p) and the average number of variables of each pattern (n_v) are analysed in order to measure the complexity of the model. EPM tries to determine the underlying phenomena that causes the discriminative characteristics in data. Therefore, it is necessary to evaluate the results extracted with respect to unseen instances of the problem in order to assert that the knowledge extracted truly belongs to the underlying phenomena that generates the data. Traditional data splitting procedures such as hold-out or cross-validation are used in order to assert the veracity of the knowledge extracted independently of the data partition employed.

Table 5 Algorithms and their parameters used in this experimental study

Algorithm	Parameters
EvAEFP-Spark [45]	Number of labels = 3
	Number of evaluations = 10000
	population length = 100
	Crossover probability = 0.6
	Mutation probability = 0.01
BD-EFEP	Number of labels = 3
	Number of evaluations = 10000
	Population length = 51
	Crossover probability = 0.6
	Mutation probability = 0.1

In this way, the results presented are averages obtained using a fivefold stratified cross-validation procedure for all measures except GR. For GR, the value presented is the percentage of patterns whose GR is greater than one in test data. This is because it is not possible to compute an average, as the domain of this measure is $[0, \infty]$.

– Run environment. The experimental study has been carried out using the computation cluster of the Advanced Studies in Information and Communication Technologies of the University of Jaén[1]. This cluster is composed of 16 nodes with 2 × Intel Xeon E5-2670v2, 10 cores at 2.50 Ghz and 64 GB of RAM. The cluster is based in RedHat Enterprise Linux (release 7.3).

Analysis of the Results of Quality Measures

This section shows and analyses the results of the algorithms considering the different quality measures. It is important to remark that both EvAEFP-Spark and BD-EFEP use a global MapReduce approach. This means that the results are the same regardless the number of partitions employed. Therefore, the number of partitions used is not presented in this analysis. Nevertheless, normally the number of partitions used is presented in order to view the scalability of the method. This study is presented in "Scalability Analysis and Time Comparison".

Table 6 presents the average results of each algorithm on each dataset. The last two rows contain the median for all the datasets in order to ease the analysis. The use of the median was due to the high dispersion presented in the results of each dataset, so the median was used instead of the average in order to avoid the bias produced by the dispersion of the data. An analysis of the results for each quality measure is shown below:

[1] http://ceatic.ujaen.es/en

Table 6 Average results obtained by the big data algorithms for emerging pattern mining

Dataset	Algorithm	n_r	n_v	WRACC	CONF	GR	Strength	χ^2	TPR	FPR
census	BD-EFEP	21.400 ± 1.673	4.092 ± 0.450	0.621 ± 0.014	0.937 ± 0.022	1.000 ± 0.000	0.405 ± 0.033	1.000 ± 0.000	0.621 ± 0.051	0.379 ± 0.041
	EvAEFP-Spark	9.800 ± 1.095	6.785 ± 2.610	0.552 ± 0.023	0.807 ± 0.097	0.892 ± 0.114	0.305 ± 0.143	0.966 ± 0.059	0.535 ± 0.262	0.431 ± 0.222
hepmass	BD-EFEP	13.400 ± 4.827	3.079 ± 0.919	0.675 ± 0.024	0.815 ± 0.036	1.000 ± 0.000	0.409 ± 0.068	1.000 ± 0.000	0.542 ± 0.106	0.191 ± 0.060
	EvAEFP-Spark	2.000 ± 0.000	2.100 ± 0.224	0.709 ± 0.001	0.642 ± 0.000	1.000 ± 0.000	0.606 ± 0.000	1.000 ± 0.000	0.944 ± 0.000	0.526 ± 0.001
higgs	BD-EFEP	5.000 ± 2.449	5.896 ± 0.950	0.517 ± 0.003	0.627 ± 0.063	1.000 ± 0.000	0.084 ± 0.020	1.000 ± 0.001	0.149 ± 0.041	0.115 ± 0.041
	EvAEFP-Spark	2.000 ± 0.000	6.400 ± 1.746	0.505 ± 0.001	0.653 ± 0.085	1.000 ± 0.000	0.251 ± 0.106	1.000 ± 0.000	0.496 ± 0.212	0.486 ± 0.214
kddcup	BD-EFEP	19.000 ± 2.121	1.993 ± 0.332	0.566 ± 0.032	0.410 ± 0.079	0.663 ± 0.074	0.214 ± 0.035	0.586 ± 0.107	0.249 ± 0.032	0.088 ± 0.027
	EvAEFP-Spark	50.600 ± 8.678	15.205 ± 0.434	0.694 ± 0.026	0.532 ± 0.040	0.603 ± 0.056	0.414 ± 0.059	0.603 ± 0.056	0.416 ± 0.061	0.002 ± 0.002
rlcp	BD-EFEP	9.400 ± 1.342	2.762 ± 0.258	0.958 ± 0.007	0.998 ± 0.002	1.000 ± 0.000	0.917 ± 0.014	1.000 ± 0.000	0.930 ± 0.015	0.014 ± 0.003
	EvAEFP-Spark	2.000 ± 0.000	2.300 ± 0.274	0.969 ± 0.025	0.606 ± 0.091	1.000 ± 0.000	0.939 ± 0.050	1.000 ± 0.000	0.952 ± 0.059	0.013 ± 0.010
susy	BD-EFEP	15.400 ± 2.608	5.506 ± 0.431	0.458 ± 0.185	0.564 ± 0.288	0.787 ± 0.441	0.111 ± 0.063	0.850 ± 0.320	0.182 ± 0.100	0.155 ± 0.041
	EvAEFP-Spark	2.800 ± 0.447	6.833 ± 1.312	0.387 ± 0.038	0.172 ± 0.020	0.267 ± 0.149	0.121 ± 0.055	0.367 ± 0.075	0.213 ± 0.078	0.212 ± 0.027
Median	BD-EFEP	14.400 ± 2.503	**3.586 ± 0.557**	0.593 ± 0.044	**0.721 ± 0.081**	**1.000 ± 0.086**	0.310 ± 0.039	**1.000 ± 0.071**	0.395 ± 0.058	**0.135 ± 0.036**
	EvAEFP-Spark	**2.400 ± 1.703**	6.592 ± 1.100	**0.623 ± 0.019**	0.624 ± 0.056	0.946 ± 0.053	**0.359 ± 0.069**	0.983 ± 0.032	**0.515 ± 0.112**	0.321 ± 0.079

The best median result for each quality measure is highlighted in bold

- n_p. The number of patterns obtained by each big data algorithm is low. This reduces significantly the complexity of the model and ease the analysis. It is remarkable that the number of patterns obtained by BD-EFEP is usually much higher than those obtained in EvAEFP-Spark. In addition, the variability in the number of patterns extracted is higher as well, so EvAEFP-Spark is more robust in terms of the extraction a lower number of patterns.
- n_v. The results obtained by BD-EFEP contain a significantly smaller number of variables than in the case of EvAEFP-Spark. Moreover, the variation of the results is lower in BD-EFEP than in EvAEFP-Spark, so patterns are more likely to remain with low number of variables. This is due to the use of the oriented initialisation and the TC-based procedures, that promote the extraction of more general patterns with less variables taking part.
- WRACC. The patterns extracted by the EvAEFP-Spark algorithm present a higher interest value than the ones extracted by BD-EFEP. This fact can be produced because of the use of optimisation objectives focused on reliability in BD-EFEP which penalises the extraction of more general patterns that have a great influence for achieving a higher WRAcc value.
- TPR and FPR. The results of these two measures are analysed at the same time in order to facilitate its understanding. In general, both algorithms obtain very high TPR values which means that examples of positive class are well-covered. The difference between TPR and FPR means that the knowledge extracted contains a good trade-off between generality and reliability. The comparison of these algorithms show that BD-EFEP obtains better results than EvAEFP-Spark for FPR, but not for TPR. However, the trade-off generality/reliability is better in BD-EFEP than in EvAEFP-Spark, as the difference between TPR and FPR is higher. This is a consequence of the good synergy between the mechanisms focused in precision used in BD-EFEP and the ones focused in generality and diversity. Additionally, this mechanisms allow a better robustness in the results with respect to EvAEFP-Spark as its standard deviation is significantly less.
- CONF. BD-EFEP obtains, in general, higher values in confidence than EvAEFP-Spark. The variation in the confidence of BD-EFEP patterns is higher than in EvAEFP-Spark, but it is not significantly high with respect to EvAEFP-Spark. This can be explained by the use in BD-EFEP of objectives composed of quality measures focused on the extraction of precise patterns, and also by the use of the post-processing filter based on confidence. Therefore, it can be observed that patterns obtained by BD-EFEP are much more reliable than that extracted by EvAEFP-Spark.
- GR. In this measure, the average results obtained by BD-EFEP are significantly higher than those obtained in EvAEFP-Spark at the cost of a higher variability. This result can be a produced by the use of the oriented initialisation and mutation operators in BD-EFEP, together with the optimisation objectives that promote the extraction of patterns with less FPR, so they are more precise and therefore, it is more likely to extract patterns with a GR greater than one.
- Strength. The results extracted for this measure are pretty similar to those extracted by TPR. Actually, the results extracted by EvAEFP-Spark presents a higher generality than those extracted by BD-EFEP. This fact can be produced because of the mechanisms used focused on reliability in BD-EFEP that can extract patterns more adjusted to training data so they are more difficult to generalise.
- χ^2. Patterns extracted by BD-EFEP presents a slightly higher χ^2 value than those of EvAEFP-Spark. However, the result is more variable than in EvAEFP-Spark. These means that patterns extracted by BD-EFEP are slightly more interesting than those extracted by EvAEFP-Spark. This can be produced by the use of the mechanisms used for reliability, which produces the extraction of patterns with more significant differences between its coverage amongst the positive and negative examples.

In EPM, reliable patterns are key as we are finding the discriminative relationships between classes. According to the results extracted, the pattern model of BD-EFEP presents a better reliability than the one extracted by EvAEFP-Spark, because of the FPR, CONF and GR values. This is due to the use of the optimisation objectives mainly focused on reliability and its final confidence filter. On the other hand, the generality of the patterns is higher in EvAEFP-Spark. As a side effect, we can observe that WRACC is higher in EvAEFP-Spark than in BD-EFEP because of this generality. However, the trade-off between these aspects, which is a key point, is better in BD-EFEP. One of the main drawbacks in BD-EFEP is that its pattern model is more complex in terms of patterns, but it is simpler with respect to the number of variables. However, the more complex models, in BD-EFEP better guarantees the description of the whole dataset is more guaranteed than in EvAEFP-Spark because of its TC-based procedure. Therefore, the use of more patterns is justified in order to extract a better description of the data. As a conclusion, BD-EFEP is a good alternative for the extraction of more reliable EPs with a good trade-off between its generality.

Scalability Analysis and Time Comparison

In this section, a scalability study for the BD-EFEP algorithm is shown. In this study, BD-EFEP is executed with different number of partitions in order to determine if a greater parallelisation produces a decrease of the execution times. The number of partitions used on each dataset are: 16, 32, 64, 128 and 256 partitions. BD-EFEP is also compared against EvAEFP-Spark in order to determine whether BD-EFEP is faster than EvAEFP-Spark or not.

Table 7 presents the execution times of BD-EFEP and EvAEFP-Spark. The best execution time for each dataset is highlighted. A "not defined" result means an execution time grater than 86400 s (24 h). The results show that the execution time of BD-EFEP decreases when the number of partitions used is increased. However, it is important to remark that the decrease in the execution time is non-linear. This is explained by the fact that the MapReduce process is not applied over the whole evolutionary process. Despite of this, the MapReduce approach used in BD-EFEP is able to efficiently deal with bigger datasets if the number of partitions used is big enough.

In addition to this, BD-EFEP significantly outperforms the execution time of EvAEFP-Spark algorithm, regardless of the number of partitions used. In fact, the big difference is due, on the one hand, to the representation of the class in the chromosomes, allowing to perform a single execution of BD-EFEP to obtain results for all the classes. On the other hand, the EvAEFP-Spark uses an iterative rule learning approach which is very slow. Therefore, BD-EFEP is an alternative allowing the extraction of descriptive knowledge in less time than other alternatives for the extraction of EPs within big data environments.

Table 7 Average execution time, in seconds, of the EvAEFP-Spark and BD-EFEP algorithms with different number of partitions

Dataset	Partitions	BD-EFEP	EvAEFP-Spark
Census	16	245	2863
	32	183	2522
	64	149	2169
	128	**135**	**2100**
	256	136	2298
kddcup	16	384	13980
	32	275	12472
	64	230	**11859**
	128	**196**	11960
	256	206	12347
rlcp	16	3208	3660
	32	2730	2969
	64	2564	2746
	128	**2535**	**2621**
	256	2765	2948
Susy	16	4097	11645
	32	3336	10729
	64	2882	10481
	128	2772	**9929**
	256	**2766**	10865
Higgs	16	8131	Not defined
	32	8394	Not defined
	64	7171	Not defined
	128	6799	**85900**
	256	**6796**	Not defined
Hepmass	16	7058	Not defined
	32	7376	Not defined
	64	6198	Not defined
	128	5779	**77618**
	256	**5701**	Not defined

The fastest execution time for each algorithm and dataset is highlighted in bold

Conclusions

This paper presents the BD-EFEP algorithm, an EFS whose objective is the extraction of EPs within big data environments. BD-EFEP is a MOEA oriented towards the extraction of high quality EPs, in order to get a simple and interpretable set of patterns that describes the discriminative characteristics of the data. For this purpose, specific operators have been developed. BD-EFEP is based in a "chromosome = rule" representation where the consequent part of the pattern is represented in order to extract patterns for all the classes in a single execution. Within the "chromosome = rule" approach, it follows a competitive-cooperative schema. The patterns compete amongst them but they cooperate in order to describe accurate information about the greatest possible area of the space. The generality of the patterns is achieved by means of the mutation and oriented initialisation operators. The diversity is promoted by the use of the NSGA-II crowding distance operator. It also uses a TC-based competition and a confidence-based filter in order to promote the precision of the patterns. Finally, the algorithm performs the evaluation of the individuals by means of a MapReduce-based global approach in order to improve the efficiency in big data environments without degrading the quality of the results obtained. The MapReduce approach computes a partial contingency table, obtained from the data available on each partition. Then, all of these tables are joined in order to get the complete contingency table where all the quality measures can be calculated easily.

The suitability of BD-EFEP has been proven by means of a comparison with other EPM approach focused in big

data. As a conclusion, BD-EFEP obtains a set of patterns with a significant improvement in reliability, which is key in the extraction in EPM, together with a better trade-off between the generality and the reliability of the results extracted. This improvement produces the extraction of a higher number of patterns than the other alternative. Nevertheless, the number of patterns is low enough to be easily analysed while the description of the whole dataset is improved. These patterns usually define different areas of data, so patterns extracted define the underlying phenomena of the different zones of the search space, which is the main objective of EPM. Finally, a comparison of the execution time against EvAEFP-Spark shows a very good scalability of the algorithm, where patterns are extracted significantly faster in BD-EFEP than in the other approach. Therefore, the proposed algorithm is a relevant alternative for the extraction of high-quality EPs in big data environments.

Funding Information This study was funded by the Spanish Ministry of Economy and Competitiveness under the project TIN2015-68454-R and FPI 2016 Scholarship reference BES-2016-077738 (FEDER Founds).

Compliance with Ethical Standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Abbasi A, Sarker S, Chiang RH. Big data research in information systems: toward an inclusive research agenda. J Assoc Inf Syst. 2016;17(2):1–32.
2. Aljarah I, Alam AZ, Faris H, Hassonah MA, Mirjalili S, Saadeh H. Simultaneous feature selection and support vector machine optimization using the grasshopper optimization algorithm. Cogn Comput. 2018;10(3):478–495.
3. Antonelli M, Bernardo D, Hagras H, Marcelloni F. Multiobjective evolutionary optimization of type-2 fuzzy rule-based systems for financial data classification. IEEE Trans Fuzzy Syst. 2017;25(2):249–264.
4. Asuncion A, Newman DJ. UCI machine learning repository. 2007. http://www.ics.uci.edu/mlearn/MLRepository.html.
5. Babaei M, Sheidaii M. Desirability-based design of space structures using genetic algorithm and fuzzy logic. International Journal of Civil Engineering. 2017;15(2):231–245.
6. Bailey J, Manoukian T, Ramamohanarao K. Fast algorithms for mining emerging patterns. In: Principles of data mining and knowledge discovery. Berlin: Springer; 2002. p. 187–208.
7. Bethea R, Duran B, Boullion T. Statistical methods for engineers and scientists. 1995.
8. Beyer MA, Laney D. The importance of 'big data': a definition. 2012.
9. 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(8):3439–3448.
10. 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.
11. 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(5):958–970.
12. 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(12):2435–2448.
13. Carmona CJ, del Jesus MJ, Herrera F. A unifying analysis for the supervised descriptive rule discovery via the weighted relative accuracy. Knowl-Based Syst. 2018;139:89–100.
14. 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 Systems with Applications. 2012;39:11,243–11,249.
15. 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. Inf Sci. 2015;298:180–197.
16. Casillas J, Carse B, Bull L. Fuzzy-XCS: a michigan genetic fuzzy system. IEEE Trans Fuzzy Syst. 2007;15(4):536–550.
17. Chakraborty S, Dey N, Samanta S, Ashour AS, Barna C, Balas M. Optimization of non-rigid demons registration using cuckoo search algorithm. Cogn Comput. 2017;9(6):817–826.
18. Chi Z, Yan H, Pham T. Fuzzy algorithms: with applications to image processing and pattern recognition, vol 10 World Scientific. 1996.
19. Cordón O, Herrera F, Hoffmann F, Magdalena L. Genetic fuzzy systems: evolutionary tuning and learning of fuzzy knowledge bases world scientific. 2001.
20. Cordón O., del Jesus MJ, Herrera F, Lozano M. MOGUL: A methodology To obtain genetic fuzzy rule-based systems under the iterative rule learning approach. Int J Intell Syst. 1999;14:1123–1153.
21. Dean J, Ghemawat S. Mapreduce: Simplified data processing on large clusters. In: Operating systems design and implementation (OSDI); 2004. p. 137–150.
22. Dean J, Ghemawat S. Mapreduce: Simplified data processing on large clusters. Commun ACM. 2008;51(1):107–113.
23. Dean J, Ghemawat S. Mapreduce: A flexible data processing tool. Commun ACM. 2010;53(1):72–77.
24. Deb K. Multi-objective optimization using evolutionary algorithms. Hoboken: Willey; 2001.
25. Deb K, Pratap A, Agrawal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput. 2002;6(2):182–197.
26. DeJong K, Spears W, Gordon DF. Using genetic algorithms for concept learning. Mach Learn. 1997;13(2):161–188.
27. Dheeru D, Karra Taniskidou E. UCI machine learning repository. 2017. http://archive.ics.uci.edu/ml.
28. Dong GZ, Li JY. Efficient mining of emerging patterns: discovering trends and differences. In: Proc of the 5th ACM SIGKDD international conference on knowledge discovery and data mining. New York : ACM Press; 1999. p. 43–52.
29. Dong GZ, Zhang X, Wong L, Li JY. CAEP: Classification By aggregating emerging patterns. In: Proc of the discovery science, LNCS. Berlin: Springer; 1999. p. 30–42.

30. Elkano M, Galar M, Sanz J, Bustince H. Chi-bd: a fuzzy rule-based classification system for big data classification problems. Fuzzy Sets Syst. 2018;348:75–101.
31. Eshelman LJ. Foundations of genetic algorithms, chap. The CHC adaptive search algorithm: how to have safe search when engaging in nontraditional genetic recombination, pp 265–283. 1991.
32. Fan H, Ramamohanarao K. Efficiently mining interesting emerging patterns. In: Proc of the 4th international conference on web-age information management; 2003. p. 189–201.
33. 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(6):721–737.
34. Fayyad UM, Piatetsky-Shapiro G, Smyth P. From data mining to knowledge discovery: an overview. In: Advances in knowledge discovery and data mining. Palo Alto: AAAI/MIT Press; 1996. p. 1–34.
35. Fernández A, Altalhi A, Alshomrani S, Herrera F. Why linguistic fuzzy rule based classification systems perform well in big data applications. Int J Comput Intell Syst. 2017;10(1):1211–1225.
36. Fernández A, Carmona CJ, del Jesus MJ, Herrera F. A view on fuzzy systems for big data: progress and opportunities. International Journal of Computational Intelligence Systems. 2016;9(1):69–80.
37. Fernández A, Río S, López V, Bawakid A, del Jesus M, Benítez J, Herrera F. Big data with cloud computing: an insight on the computing environment, mapreduce and programming frameworks. WIREs Data Mining and Knowledge Discovery. 2014;5(4):380–409.
38. Fogel DB. Evolutionary computation - toward a new philosophy of machine intelligence. IEEE Press. 1995.
39. Gamberger D, Lavrac N. Expert-guided subgroup discovery: methodology and application. J Artif Intell Res. 2002;17:501–527.
40. García-Borroto M, Martínez-Trinidad J, Carrasco-Ochoa J. Fuzzy emerging patterns for classifying hard domains. Knowl Inf Syst. 2011;28(2):473–489.
41. García-Borroto M, Martínez-Trinidad JF, Carrasco-Ochoa JA. A survery of emerging patters for supervised classification. Artif Intell Rev. 2014;42(4):705–721.
42. 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 classifications. Pattern Recogn. 2010;43(9):3025–3034.
43. García-Vico AM, Carmona CJ, González P, del Jesus MJ. Moea-efep: Multi-objective evolutionary algorithm for extracting fuzzy emerging patterns. IEEE Transactions on Fuzzy Systems (In Press).
44. García-Vico A, Carmona C, Martín D., García-Borroto M, del Jesus M. An overview of emerging pattern mining in supervised descriptive rule discovery: taxonomy, empirical study, trends, and prospects. WIREs Data Mining Knowl Discov. 2018;8:e1231. https://doi.org/10.1002/widm.1231.
45. García-Vico AM, González P, del Jesus MJ, Carmona CJ. A first approach to handle emergining patterns mining on big data problems: the evaefp-spark algorithm. In: IEEE International conference on fuzzy systems; 2017. p. 1–6.
46. García-Vico AM, Montes J, Aguilera J, Carmona CJ, del Jesus MJ. Analysing concentrating photovoltaics technology through the use of emerging pattern mining. In: Proc of the 11th international conference on soft computing models in industrial and environmental applications. Berlin: Springer; 2016. p. 1–8.
47. Geng L, Hamilton HJ. Interestingness measures for data mining: a survey. ACM Comput Surv (CSUR). 2006;38(3):9.
48. Goldberg DE. Genetic algorithms in search, optimization and machine learning. Addison-wesley Longman Publishing Co. Inc. 1989.
49. Herrera F. Genetic fuzzy systems: taxomony, current research trends and prospects. Evol Intel. 2008;1:27–46.
50. Holland JH. Adaptation in natural and artificial systems. Cambridge: University of Michigan Press; 1975.
51. Huang HC, Chiang CH. Backstepping holonomic tracking control of wheeled robots using an evolutionary fuzzy system with qualified ant colony optimization. Int J Fuzzy Syst. 2016;18(1):28–40.
52. Hüllermeier E. Fuzzy methods in machine learning and data mining: status and prospects. Fuzzy Sets Syst. 2005;156(3):387–406.
53. Hüllermeier E. Fuzzy sets in machine learning and data mining. Appl Soft Comput. 2011;11(2):1493–1505.
54. Ishibuchi H, Tsukamoto N, Hitotsuyanagi Y, Nojima Y. Effectiveness of scalability improvement attempts on the performance of nsga-ii for many-objective problems. In: Proceedings of the 10th annual conference on genetic and evolutionary computation (GECCO '08); 2008. p. 649–656.
55. 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(4):578–592.
56. Kloesgen W. Explora: a multipattern and multistrategy discovery assistant. In: Advances in knowledge discovery and data mining, pp 249–271. American association for artificial intelligence; 1996.
57. Koza JR. Genetic programming: on the programming of computers by means of natural selection. Cambridge: MIT Press; 1992.
58. Kralj-Novak P, Lavrac N, Webb GI. Supervised descriptive rule discovery: a unifying survey of constrast set, emerging pattern and subgroup mining. J Mach Learn Res. 2009;10:377–403.
59. Larson D, Chang V. A review and future direction of agile, business intelligence, analytics and data science. Int J Inf Manag. 2016;36(5):700–710.
60. Leung KS, Leung Y, So L, Yam KF. Rule learning in expert systems using genetic algorithm: 1, concepts. Proc of the 2nd international conference on fuzzy logic and neural networks. In: Jizuka K, editors; 1992. p. 201–204.
61. 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.
62. Li JY, Dong GZ, Ramamohanarao K, Wong L. DeEPs: a new instance-based lazy discovery and classification system. Mach Learn. 2004;54(2):99–124.
63. Lin J. Mapreduce is good enough? if all you have is a hammer, throw away everything that's not a nail!. Big Data. 2013;1(1):28–37.
64. 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(3):598–615.
65. Loyola-González O, Martínez-Trinidad JF, Carrasco-Ochoa JA, García-Borroto M. Effect of class imbalance on quality measures for contrast patterns: an experimental study. Inf Sci. 2016;374:179–192.
66. Loyola-González O, Martínez-Trinidad JF, Carrasco-Ochoa JA, García-Borroto M. Study of the impact of resampling methods for contrast pattern based classifiers in imbalanced databases. Neurocomputing. 2016;175:935–947.
67. Loyola-González O, Medina-Pérez MA, Martínez-Trinidad JF, Carrasco-Ochoa JA, Monroy R, García-Borroto M. Pbc4cip: a new contrast pattern-based classifier for class imbalance problems. Knowl-Based Syst. 2017;115:100–109.

68. L'heureux A, Grolinger K, Elyamany HF, Capretz MA. Machine learning with big data: challenges and approaches. IEEE Access. 2017;5(5):777–797.
69. Martens D, Baesens B, Van Gestel T, Vanthienen J. Comprehensible credit scoring models using rule extraction from support vector machines. Eur J Oper Res. 2007;183(3):1466–1476.
70. Métivier JP, Lepailleur A, Buzmakov A, Poezevara G, Crémilleux B, Kuznetsov SO, Goff JL, Napoli A, Bureau R, Cuissart B. Discovering structural alerts for mutagenicity using stable emerging molecular patterns. J Chem Inf Model. 2015;55(5):925–940.
71. Michalski RS, Stepp R. Revealing conceptual structure in data by inductive inference. Machine Intelligence. 1982;10:173–196.
72. Miller BL, Goldberg DE. Genetic algorithms, tournament selection, and the effects of noise. Complex System. 1995;9:193–212.
73. Molina D, LaTorre A, Herrera F. An insight into bio-inspired and evolutionary algorithms for global optimization: review, analysis, and lessons learnt over a decade of competitions. Cognitive Computation, pp 1–28. 2018.
74. Nie Y, Wang H, Lu X, Qin Y. Parallel emerging patterns in microarray. In: Proc of the 6th intelligent human-machine systems and cybernetics; 2014. p. 82–85.
75. Onieva E, Hernandez-Jayo U, Osaba E, Perallos A, Zhang X. A multi-objective evolutionary algorithm for the tuning of fuzzy rule bases for uncoordinated intersections in autonomous driving. Inf Sci. 2015;321:14–30.
76. Padillo F, Luna JM, Herrera F, Ventura S. Mining association rules on big data through mapreduce genetic programming. Integrated Computer-Aided Engineering (In Press), 1–19. 2018.
77. Padillo F, Luna JM, Ventura S. An evolutionary algorithm for mining rare association rules: a big data approach. In: 2017 IEEE Congress on evolutionary computation (CEC); 2017. p. 2007–2014.
78. Peralta D, Río S, Ramíez-Gallego S, Triguero I, Beníez JM, Herrera F. Evolutionary feature selection for big Data classification: a mapreduce approach. Math Probl Eng. 2015;2015:1–11.
79. Pulgar-Rubio F, Rivera-Rivas AJ, Pérez-Godoy MD, González P, Carmona CJ, Del Jesus MJ. MEFASD-BD: multi-objective evolutionary fuzzy algorithm for subgroup discovery in big data environments - a mapreduce solution. Knowl-Based Syst. 2017;117:70–78.
80. Ramamohanarao K, Fan H. Patterns based classifiers. World Wide Web. 2007;10(1):71–83.
81. Ramírez-Gallego S, Fernández A., García S, Chen M, Herrera F. Big data: tutorial and guidelines on information and process fusion for analytics algorithms with mapreduce. Information Fusion. 2018;42:51–61.
82. Ramírez-Gallego S, García S, Benítez J, Herrera F. A distributed evolutionary multivariate discretizer for big data processing on apache spark. Swarm Evol Comput. 2018;38:240–250.
83. del Río S, López V, Benítez JM, Herrera F. A mapreduce approach to address big data classification problems based on the fusion of linguistic fuzzy rules. International Journal of Computational Intelligence Systems. 2015;8(3):422–437.
84. Rodríguez-Fdez I, Mucientes M, Bugarín A. FRULER: Fuzzy rule learning through evolution for regression. Inf Sci. 2016;354:1–18.
85. Ruiz E, Casillas J. Adaptive fuzzy partitions for evolving association rules in big data stream. Int J Approx Reason. 2018;93:463–486.
86. Sanz JA, Bernardo D, Herrera F, Bustince H, Hagras H. A compact evolutionary interval-valued fuzzy rule-based classification system for the modeling and prediction of real-world financial applications with imbalanced data. IEEE Trans Fuzzy Syst. 2015;23(4):973–990.
87. Shvachko K, Kuang H, Radia S, Chansler R. The hadoop distributed file system. In: Proceedings of the 2010 IEEE 26th symposium on mass storage systems and technologies (MSST2010); 2010. p. 1–10.
88. sSiddique N, Adeli H. Nature inspired computing: an overview and some future directions. Cogn Comput. 2015;7(6):706–714.
89. Storn R, Price K. Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces. Tech. Rep TR-95-012. 1995.
90. Takagi T, Sugeno M. Fuzzy identification of systems and its applications to modeling and control. IEEE Trans Syst Man Cybern. 1985;15(1):116–132.
91. Tan PN, Kumar V, Srivastava J. Selecting the right objective measure for association analysis. Inf Syst. 2004;29(4):293–313. Knowledge Discovery and Data Mining (KDD 2002).
92. Terlecki P, Walczak K. Efficient discovery of Top-K minimal jumping emerging patterns. In: Proc of the 6th international conference rough sets and current trends in computing. Berlin: Springer; 2008. p. 438–447.
93. Wang L, Wang Y, Zhao D. Building emerging pattern (ep) random forest for recognition. In: Proc of the 17th IEEE international conference on image processing; 2010. p. 1457–1460.
94. Wang Z, Fan H, Ramamohanarao K. Exploiting maximal emerging patterns for classification. In: Proc of the 17th australian joint conference on artificial intelligence, LNCS. Berlin: Springer; 2005. p. 1062–1068.
95. Wixom B, Ariyachandra T, Douglas DE, Goul M, Gupta B, Iyer LS, Kulkarni UR, Mooney JG, Phillips-Wren GE, Turetken O. The current state of business intelligence in academia: the arrival of big data. Commun Assoc Inf Syst. 2014;34(1):1–13.
96. Wong ML, Leung KS. Data mining using grammar based genetic programming and applications. Dordrecht: Kluwer Academics Publishers; 2000.
97. Yaqoob I, Hashem IAT, Gani A, Mokhtar S, Ahmed E, Anuar NB, Vasilakos AV. Big data: from beginning to future. Int J Inf Manag. 2016;36(6):1231–1247.
98. 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: Proc of the 15th international conference on web-age information management, LNCS. Amsterdam: Elsevier; 2014. p. 277–293.
99. Zadeh LA. The concept of a linguistic variable and its applications to approximate reasoning. Parts I, II, III. Inf Sci. 1975;8-9:199–249,301–357, 43–80.
100. Zadeh LA. Soft computing and fuzzy logic. IEEE Softw. 1994;11(6):48–56.
101. Zaharia M, Chowdhury M, Das T, Dave A, Ma J, McCauley M, Franklin M, Shenker S, Stoica I. Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing. In: Proceedings of the 9th USENIX symposium on networked systems design and implementation; 2012.
102. Zaharia M, Chowdhury M, Franklin MJ, Shenker S, Stoica I. Spark: Cluster computing with working sets. In: Proceedings of the 2nd USENIX conference on hot topics in cloud computing; 2010. p. 10–10.