




# A Review of Supervised Classification based on Contrast Patterns: Applications, Trends, and Challenges

Octavio Loyola-González · Miguel Angel Medina-Pérez · Kim-Kwang Raymond Choo 

Received: 16 August 2019 / Accepted: 19 June 2020  
© Springer Nature B.V. 2020

**Abstract** Supervised classification based on Contrast Patterns (CP) is a trending topic in the pattern recognition literature, partly because it contains an important family of both understandable and accurate classifiers. In this paper, we survey 105 articles and provide an in-depth review of CP-based supervised classification and its applications. Based on our review, we present a taxonomy of the existing application domains of CP-based supervised classification, and a scientometric study. We also discuss potential future research opportunities.

**Keywords** Supervised classification · Contrast patterns · Review · Taxonomy

## 1 Introduction

Artificial intelligence (e.g., machine learning and deep learning) approaches are increasingly popular in the literature [3, 140, 164, 166, 207]. Examples of machine learning and deep learning approaches include random forests, gradient boosting, convolutional neural networks, and pattern recognition. Supervised classification is one of the most popular pattern recognition approaches [32, 119], which has been widely studied and applied to many domains, such as bioinformatics [13, 84, 203], human activity recognition [94, 120, 146, 190], rare event forecasting [34, 78, 162], information retrieval [18, 30, 171, 191], face recognition [9, 134, 135], fingerprint identification [79, 143], Internet of Things [8, 202], and more recently COVID-19 (also referred to as novel Coronavirus, 2019-nCoV and SARS-CoV-2) [138, 182].

Contrast pattern-based classification continues to attract interest from the research and practitioner communities since its introduction in 1963, as evidenced by the number of supervised classifiers proposed over the past decade. However, for many practical problems, obtaining a high-classification result is insufficient, because experts should also understand the classification model [46, 119, 127]. In many application domains, the lack of comprehensibility in the classification model(s) can result in resistance or reluctance to use certain classifiers, and in other cases, it becomes mandatory to use an understandable model. For example, when credit is denied to a customer, *the U.S.'s*

---

O. Loyola-González (✉)  
Tecnologico de Monterrey, Reserva Territorial Atlixcáyotl,  
Vía Atlixcáyotl No. 2301, Puebla 72453, Mexico  
e-mail: octavioloyola@tec.mx

M. A. Medina-Pérez  
Tecnologico de Monterrey, Carretera al Lago de Guadalupe  
Km. 3.5, Atizapán, Estado de Mexico 52926, México  
e-mail: miguel@tec.mx

K.-K. R. Choo  
Department of Information Systems and Cyber Security,  
University of Texas at San Antonio, San Antonio, TX  
78249, USA  
e-mail: raymond.choo@fulbrightmail.org

*Equal Credit Opportunity Act* requires financial institutions to provide reasons for rejecting the application, and vague reasons for denial are illegal [119, 133]. Similarly, if the classifier is used to provide evidence in a court of law, it would be generally required to explain how the evidence was obtained.

Contrast pattern-based classification has been shown to achieve both accurate classification results and understandable models in many practical contexts, such as gene expression profiles [50], characterization for subtypes of leukemia [108], gene transfer and microarray concordance analysis [132], classification of spatial and image data [98], structural alerts for computational toxicology [16], criminal behavior [120], and prediction of heart diseases [93].

A supervised classifier based on contrast patterns uses a collection of contrast patterns to create a model that classifies a query object in a predefined class [47, 200]. A *pattern* is an expression defined in a certain language that describes a collection of objects [46, 47, 123–125, 127]. Usually, a pattern is represented by a conjunction of relational statements (a.k.a *items*), each with the form:  $[f_i \# v_j]$ , where  $v_j$  is a value in the domain of feature  $f_i$ , and  $\#$  is a relational operator from the set  $\{\in, \notin, =, \neq, \leq, >\}$  [47, 127, 200]. For example,  $[seizures\_per\_day \in [5, 15]] \wedge [Number\_of\_fatigues > 5] \wedge [Headaches \leq 10] \wedge [cognitive\_impairment = "Mild"]$  is a pattern describing a collection of people suffering from brain cancer metastases [142]. Let  $p$  be a pattern, and  $C = \{C_1, C_2, C_3, \dots, C_n\}$  a set of classes such that  $C_1 \cup C_2 \cup C_3 \cup \dots \cup C_n = U$ ,  $\bar{C}_i = C_i^c$ ; then,  $support(p, C_i, \bar{C}_i)$  is the fraction resulting from dividing the number of objects belonging to  $C_i$  described by  $p$  by the total number of objects belonging to  $C_i$  [46, 47, 49, 127, 200].

We describe the function *support* using three parameters to be consistent with the notation that we introduced in Section 2.2.2 for describing the pattern's quality measures. Now, a *contrast pattern* (CP) for a class  $C_i$  is a pattern  $p$  where the support of  $p$  for  $C_i$  is significantly higher than any support of  $p$  for every class other than  $C_i$  [46, 47, 49, 127, 200].

Interest in classification based on contrast patterns (CPs) remains strong, as evidenced by the publication trend shown in Fig. 1<sup>1</sup> [24–26, 46, 51, 70, 72, 74,

75, 127]. From this figure, one can observe that there was an average of 700 published articles and 600 citations per year between 2009 and 2019. Therefore, in this paper, we seek to provide an in-depth overview of CP-based supervised classification that includes both theoretical approaches and real-world applications. Specifically, we review different approaches for mining CPs, methods for filtering CPs, and classification strategies based on CPs. We also focus on the utilization of CP-based supervised classification to solve real-world problems. In addition, we present a scientometric study from several papers indexed on Scopus, which are focused on pattern-based classification.

We also remark that while there are a number of similar literature review and survey articles [1, 47, 62, 73, 101], and books [48, 158], there are a number of differences as explained below:

- We review different types of contrast patterns such as frequent items, association rules, emerging patterns, and decision rules, and identify their advantages and drawbacks from both theoretical and practical perspectives.
- We provide an in-depth scientometric study taking into account the pattern types in pattern-based classification.
- We also focus on contrast patterns based on decision trees. This is a recent trending topic, where there is a lack of literature review or survey article at the time of this study.

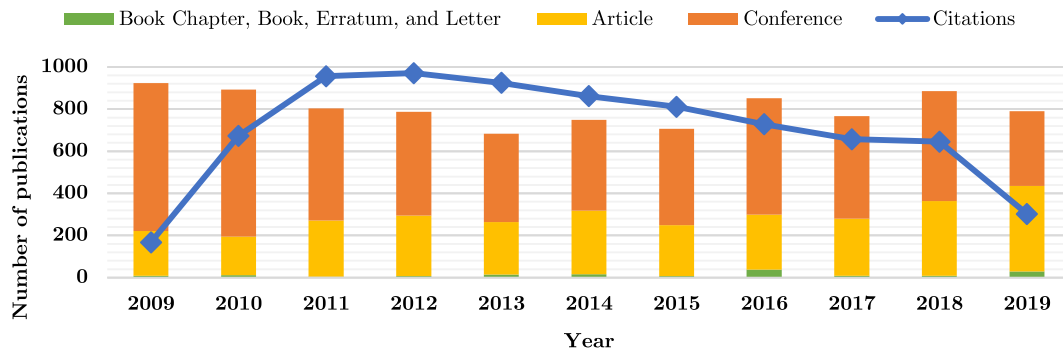
A comparative summary of existing contrast pattern reviews and surveys is presented in Table 1.

We will now explain the layout for this paper. Section 2 reviews each stage of the CP-based supervised classification, namely: mining, filtering, and classification. Also, in this section, we present our taxonomy and the scientometric study. In Section 3, we present the review of CP-based classification for real-world applications. In Section 4, we present the main trends and challenges associated with CP-based supervised classification. Finally, Section 5 presents our conclusion.

## 2 Supervised Classification Based on Contrast Patterns

In this section, we will provide an in-depth review of algorithms included in each stage to build a supervised classifier based on CPs [96, 127]. The stages

<sup>1</sup>Data for the figure were extracted from Scopus, Elsevier's abstract and citation database that indexes peer-reviewed articles published in scholarly books, journals, and conferences.



**Fig. 1** Statistics on contrast pattern-based classification published papers between January and December 2019, indexed on SCOPUS. From this figure, one can observe that there is significant interest in classification based on contrast patterns

are *Mining* (Section 2.1), *Filtering* (Section 2.2), and *Classification* (Section 2.3). We also provide a taxonomy based on the analysis of our review, where all algorithms are grouped into different categories (Section 2.4) and a scientometric study about supervised classification based on CPs (Section 2.5).

## 2.1 Mining Contrast Patterns

A key requirement for supervised classification based on CPs is to have a good collection of patterns. Therefore, several CP mining algorithms have been proposed with the aim of extracting a collection of high-quality patterns [46, 47, 127]. However, mining CPs remains challenging and it has been proven to be NP-hard [187] due to the exponential number of candidate patterns [61, 81, 86, 172, 197].

The algorithms for mining CPs perform an exploratory analysis using a search-space, which is defined by a set of inductive constraints provided by the user. CP mining algorithms can be broadly categorized into *exhaustive-search-based algorithms* (Section 2.1.1) and *decision tree-based algorithms* (Section 2.1.2). An exhaustive-search-based algorithm performs an exhaustive search of a combination of values for a set of features appearing to be significant in a class regarding the remaining classes, and a decision tree-based algorithm extracts CPs from a collection of decision trees.

### 2.1.1 Exhaustive-search-based algorithms for mining contrast patterns

It is common for exhaustive-search-based algorithms to transform all numerical features into nominal

features by using an *initial discretization* (a.k.a *a priori discretization*). In this way, the domain of the numerical features is partitioned into a finite number of disjoint intervals (bins). Let us consider an example where the feature *Weight* contains values (expressed in kg) from 1 to 150 in the training dataset. One possibility is to discretize this feature into three bins, namely:  $[-\infty, 50)$ ,  $[50, 100)$ , and  $[100, +\infty)$ . The square brackets “[” and “]” denote closed ends of intervals, and the round brackets “(” and “)” denote open ends. We also remark that the first bin starts with  $-\infty$  and the last bin ends with  $+\infty$ . This is to avoid errors during the classification stage, where query objects can have values outside the initial range of those values contained in the training dataset. For example, if the last bin is defined as  $[100, 150]$ , then a query object having 200 as a value in the feature *Weight* cannot be classified using that feature.

Since exhaustive-search-based algorithms transform all numerical features into nominal features, all resulting CPs have only items with the symbol = as the relational operator; hence, using items in the form  $[feature = value]$ . Usually, these mining algorithms use the method proposed by [60] as initial discretization because it has shown better performance than other methods [60].

While initial discretization allows us to use efficient algorithms for mining CPs, it may not be practical due to information loss and the limited interpretability of transformed data [70, 97].

The first classifier based on CPs is KORA-3, proposed by [17]. KORA-3 defines a CP for a two-class problem, as a combination of three values from three features appearing at least a minimum threshold of objects in one class, and not appearing in the other

**Table 1** Existing contrast pattern-based literature reviews and books

Year	Type	Ref.	Key objective(s)	Key merit(s)	Potential improvement(s)
2006	Review	[77]	To review the quality measures used in data mining	In-depth study of quality measures from a theoretical point of view	Focus on the use of quality measures in classification (filtering and ranking). Providing a section of advantages and disadvantages of each quality measure from both, theoretical and practical, points of view.
2007	Book	[76]	To review the quality measures used in data mining	More quality measures were examined here, than in [77]	A detailed taxonomy about quality measures. Similar to the above row, this book could be improve by providing a section of advantages and disadvantages of each quality measure from both, theoretical and practical, point of views. Also, this book could be improved by including quality measures for pattern-based unsupervised classification.
2011	Book	[158]	To study pattern-based algorithms used to solve key problems in data mining by using largest datasets	Studied the problems of finding frequent itemsets in both supervised and unsupervised problems	A comprehensive study, which includes practical problems. Also, it could provide in-depth experimentation by using real-world databases.
2012	Book	[48]	To present a state-of-the-art review of contrast patterns and their applications	Focused on contrast patterns for grouping associated topics such as rules, itemsets, and patterns.	A review that includes other types of patterns different than the studied emerging patterns. A taxonomy of the emerging patterns. A review about contrast pattern mining based on decision trees.
2016	Review	[101]	To review algorithms for mining association rules	Focused on rule-based classification when operations such as insert, delete, and update are executed	A study about the relation between the quality measures for rules and the studied operations (insert, delete, and update). An experimental setup including related algorithms, such as emerging patterns, jumping patterns, and fuzzy patterns.
2016	Review	[123]	To study the effect of class imbalance on quality measures for contrast patterns	An in-depth experimental study based on 32 quality measures for contrast patterns and 95 class imbalance databases. It provides a study of the quality measures by different class imbalance levels	A taxonomy of quality measures for contrast patterns. It could be including more quality measures to the experimental study.

**Table 1** (continued)

Year	Type	Ref.	Key objective(s)	Key merit(s)	Potential improvement(s)
2017	Review	[73]	To review the emerging pattern mining in supervised descriptive rule discovery	Taxonomy, empirical study, trends, and prospects.	A more comprehensive study including other types of patterns different than studied emerging patterns. More quality measure could be analyzed. It could provide the challenges of processed data (e.g., noise, high dimension) for mining emerging patterns.
2017	Review	[62]	To review itemset mining	Discussed research opportunities and the relationship to other popular pattern mining problems.	A more comprehensive study including other types of contrast patterns, like emerging patterns.
2019	Book	[47]	To explore the power of group differences by using patterns to solve data analysis problems	Reviewed pattern-based classification taking into account its applicability in practical problems.	A more comprehensive study including other types of patterns different than studied emerging patterns. A taxonomy of the studied contrast pattern algorithms.
-	Review	Proposed review	To review the contrast pattern-based classification	Reviewed different type of patterns (rules, itemsets, emerging patterns, fuzzy patterns) from both theoretical and practical perspectives.	A review about contrast pattern mining based on decision trees.

class. A CP covers an object if the object has the same value(s) that the CP has in their respective items. The main drawback of KORA-3 is that it only works with two-class problems and only generates CPs containing three Boolean items. This limits one in finding new and better CPs. A pattern  $p_1$  is *better* than another pattern  $p_2$  if both patterns are extracted from the same training dataset and  $p_1$  allows discriminating more objects than  $p_2$ . One way to measure the discriminative power among CPs is to use the support of [49] (see Section 1) as a quality measure for patterns.

In 1993, Apriori was introduced [2] and gradually became one of the most prominent pattern-based algorithms. The Apriori algorithm was proposed to extract frequent item sets from a market basket dataset. An item set (a.k.a *itemset*) is labeled as *frequent itemset* when its support exceeds a minimum threshold [2]. The algorithm finds patterns without taking into account the classes. However, we can use it to find CPs by restricting the search space to objects belonging to the same class.

Apriori uses a level-wise search to extract patterns, where  $l$ -patterns are used to find  $(l + 1)$ -patterns. Apriori extracts, at the first stage, all possible patterns containing only one item. After, it extracts patterns containing more than one item. At each iteration, a pruning procedure is conducted to remove patterns that do not meet the minimum support threshold. One drawback of Apriori is that it needs to scan the dataset at every iteration. Hence, many variants have been proposed to improve the efficiency using hashing, sampling, and distributed techniques. However, these variants do not obtain better classification results than recent pattern-based proposals [91]. Another drawback of Apriori is that it is difficult to find the best minimum support threshold for each specific dataset.

In 1998, KORA-3 was extended to KORA- $\Omega$  [184], where this extension allows finding patterns containing more than three items and handling multiple-class problems with mixed and incomplete data. The extension also uses an exhaustive search procedure for extracting patterns, which make it more difficult to be applied in several practical domains.

Later, in 1998, Max-Miner was proposed [15] as an algorithm for mining long patterns. Max-Miner avoids the limitation due to the exponential complexity of the Apriori algorithm for extracting patterns. Specifically, Max-Miner extracts only maximal patterns containing maximal frequent itemsets. An itemset is labeled as

*maximal frequent* (a.k.a *frequent closed*) if it has no superset that is frequent because any frequent itemset is a subset of a maximal frequent itemset. Max-Miner removes the bottom-up traversal of the search space proposed by [2], and it uses a heuristic strategy for quickly extracting long-frequent item sets. After that, Max-Miner uses a procedure for pruning the extracted patterns to find new frequent itemsets. The authors also showed that Max-Miner achieves improved performance, in comparison to Apriori. Nevertheless, Max-Miner has a computational complexity higher than other recent proposals based on patterns.

Later, in 1999, a new family of supervised classifiers based on the concept of Emerging Pattern (EP) was introduced by [49]. For knowing when a CP is an EP, the authors use the quality measure *Growth Rate*, which is computed using the mean of the highest ratio between the support of a certain pattern in one class and the support of the same pattern in the other classes. The way for extracting patterns is similar to the those of KORA- $\Omega$  [184] and Max-Miner [15].

In 2000, the frequent pattern (FP)-growth approach for mining patterns was introduced by [82]. This was presented as an alternative to the Apriori-based approach, and uses divide-and-conquer to produce items by creating a compact tree-structure, FP-Tree. In the experiments, the authors showed that the FP-growth method is efficient and scalable for mining both long and short patterns, and it is about an order of magnitude faster than the Apriori algorithm. However, recent methods for mining patterns have proven to be more accurate than FP-growth [91].

Another algorithm for mining emerging patterns (JEPProducer) was presented in [107], which can obtain the border representation of all jumping emerging patterns. A pattern is labeled as a *Jumping Emerging Pattern* (JEP) when it covers objects from only one class [121, 124]. JEPProducer obtains the horizontal border of each dataset. A horizontal border is represented as  $\langle \emptyset, R_1 \rangle$ , which contains all nonzero support patterns. Then, in order to obtain JEPs, it executes the *BorderDiff*( $t, T$ ) function for obtaining the border representing the difference of these sets represented by the horizontal borders. The function *BorderDiff*( $t, T$ ) [49] returns a set of minimal jumping emerging patterns that occur in  $t$  but never in  $T$ . For example, for  $t = \{1, 2, 3, 4\}$  and  $T = \{\{2, 3, 5, 6\}, \{2, 4, 7, 8\}, \{3, 4, 6, 8\}\}$ , we have  $BorderDiff(t, T) = \{\{1\}, \{2, 3, 4\}\}$ .

During the same year (2000), an algorithm for extracting maximal patterns (CLOSET) was proposed [149]. Similar to the concept of maximal itemset, a *maximal pattern* is a pattern whose super-patterns are not patterns anymore. CLOSET uses the FP-Tree structure proposed in [82], but CLOSET introduces a divide-and-conquer method at the data level. The idea is to divide the dataset into five non-overlap subsets to mine itemsets. The authors proposed a strategy based on set theory to merge the subsets and extract those only maximal patterns. The experimental setup proposed by [149], however, lacks an in-depth comparison using tens of datasets, state-of-the-art algorithms, and statistical methods for corroborating the experimental results.

ECLA [198], an algorithm for mining patterns, uses a *vertical tid-list*. The latter is a structure proposed by [181] for linking those objects covered by a collection of patterns. The authors of [198] showed that all patterns could be enumerated via simple tid-list intersections, and proposed a lattice-theoretic approach to decompose the original search space into smaller sub-spaces. For doing that, they introduced two techniques for achieving the decomposition, namely: prefix-based and maximal-clique-based partition. Additionally, they proposed three new search strategies for enumerating the patterns within each sub-space, namely: bottom-up, top-down, and hybrid search. CPs are mined using each sub-space. The bottom-up search is based on a recursive decomposition of each class into smaller classes induced by the equivalence relation. The top-down search starts with the top pattern. If it is a contrast pattern, then it is extracted. Otherwise, the algorithm checks each subset at the next level for mining patterns. This process is repeated until all levels are processed. The hybrid search is based on the intuition that the higher the support of a pattern, the more likely it is to be part of a collection of maximal patterns. The authors showed that the three new search strategies are suitable for mining patterns more efficiently than the Apriori algorithm [2]. The main drawback is that ECLA is time-consuming, especially when the dataset contains many objects [199].

Later, in 2002, a fast algorithm for mining emerging patterns (TBJEP) was proposed by [11]. TBJEP performs a global search in the training stage, and makes a previous discretization of all numerical features by using the method proposed in [60]. The authors used

a tree-based representation of the complete training dataset with the aim of representing all information of the features in a multi-value tree structure. TBJEP makes a depth-first traverse of the tree for mining patterns.

An algorithm for mining Essential Jumping Emerging Patterns (EJEPs) was introduced by [56]. Usually, EJEPs are EPs contain high Growth Rate values [57]. This algorithm extracts a set of EJEPs, which is a subset of JEPs but it does not include noise and redundant information. The single scan algorithm for mining EJEPs obtains fewer patterns than other state-of-the-art mining algorithms. However, it only works with two-class problems and uses a previous discretization of all numerical features.

In 2003, BCEP was proposed to mine emerging patterns [57]. BCEP uses a tree-based representation, similar to the one proposed by [11], for mining JEPs. Using the tree-based representation, BCEP sorts all items by using Growth Rate as a quality measure for patterns. BCEP takes advantage of its tree-based representation by performing a depth-first traversal procedure for extracting EPs. The tree is pruned to obtain only EJEPs utilizing a minimum support threshold or when a JEP is found. Similar to other algorithms discussed above, BCEP uses an initial discretization (its key drawback).

In the same year (2003), another algorithm for mining EPs (iEPMiner) was proposed [58]. iEPMiner extracts patterns using conditions of the  $X^2$  test [168], which are labeled as *Chi-EPs*. iEPMiner uses a tree-based representation, similar to the one used in [11, 57]. Based on this representation, a depth-first procedure is executed for checking certain conditions at each node of the tree. If a path of the tree fulfills these conditions, then it is saved as a candidate pattern. If the value of the  $X^2$  test between a node and its candidate child node is less than 3.84, then the addition of the new item does not significantly change the behavior of the candidate EP, because it is independent with a 95% confidence level. After finishing the tree traversal, a condition is checked over all the extracted patterns with the aim of obtaining only Chi-EPs. The main drawback of iEPMiner is that it uses an initial discretization of all numerical features and relies on the  $X^2$  test, where some assumptions need to be checked in the training dataset before applying it.

In 2004, the EPRC miner was proposed [6] for mining EPs in class imbalance problems. This type

of problems arises when there exist significantly fewer objects belonging to a class (commonly labeled as minority class) regarding the remaining classes. EPRC extracts EPs from a training dataset using an exhaustive-search-based algorithm. For each pattern of the minority class, new EPs for the minority class are built by replacing the item value by the feature value of the corresponding feature having the highest Growth Rate value; and keeping all other items as they are in the original pattern. The main drawback of EPRC is that it could build fake patterns since these new EPs do not necessarily cover objects of the minority class in the training dataset.

In the same year (2004), DeEPS for mining EPs using a lazy learning approach was proposed [106]. DeEPS uses the frequency of an object's subsets of features and the frequency-change rate of the subsets among the classes of the training dataset for removing an object from the model. The authors proposed using JEPProducer [107], with the aim of obtaining the border-based representation of a set of JEPs for each class covering the query object. The resulting set of EPs is obtained by repeating this procedure for each query object and then removing duplicate patterns.

A year later in 2005, an algorithm for mining EPs from data streams (EPDSminer) was presented [7]. A *data stream* is a continuous, unbounded, and not necessarily ordered, real-time sequence of data items [23]. Usually, exhaustive-search-based algorithms need to use the entire training dataset for mining EPs, but in data stream problems, it is infeasible to store all the data on disk. Consequently, EPDSminer uses an exhaustive-search-based algorithm for mining EPs but using the following strategy for dealing with data streams. First, data is obtained in blocks of size  $N$ , which are labeled as a two-class problem ( $c$  and  $\bar{c}$ ) and they are related to a period of time  $t$ , ( $B_c^t$  and  $B_{\bar{c}}^t$ ). Then, for each block of  $t$ , EPs are extracted and stored for both classes ( $EP_c^t$  and  $EP_{\bar{c}}^t$ ). After that, for the next period of time  $t+1$ , EPs are extracted ( $EP_c^{t+1}$  and  $EP_{\bar{c}}^{t+1}$ ) but before storing them, EPs extracted in the period of time  $t$  and  $t+1$  are ranked using the quality measure strength [123] with the aim of selecting only those EPs with high discriminate power. This procedure is repeated for each block coming from the data stream. The authors showed that EPDSminer allows for obtaining better classification results than using the C4.5 decision tree as a base classifier for the tested sampling and sliding window techniques.

In 2006, an algorithm for fast discovery and the generalization of strong jumping emerging patterns (SJEP) was proposed [59]. SJEP allows for handling some level of noise in the training data. For doing that, SJEP mines EPs supporting objects for all classes. The primary conditions are: (i) the pattern should support significantly more objects for a class than the remaining classes; (ii) the number of objects supported by the pattern in the remaining classes (considered noisy objects) should be lower than a predefined threshold. A similar solution was previously introduced in KORA-3 [184]. A drawback of SJEP is that it generates many patterns, even in small databases. As a consequence, the output model is too difficult to be understood by experts [71].

Also in 2006, a proposal for mining patterns from one-class problems (OCLEP) was introduced [33]. In the first stage of OCLEP, the original dataset is transformed by using an initial discretization, resulting in  $D$ . After that, patterns are extracted using a bagging method. OCLEP executes  $m$  times the function  $BoderDiff(t, T)$  for  $t \in D$  and  $T \subseteq D - \{t\}$  as follows: if  $D$  is small, let  $m = 100$  and each  $T$  should be a random subset of  $D - \{t\}$  such as  $|T|$  is in the range of [200, 800]. OCLEP allows achieving good detection accuracy while keeping the false positive rate low, in comparison to other state-of-the-art approaches. Nevertheless, OCLEP does not improve the results obtained by ocSVM [165]. Limitations of OCLEP include the need for initial discretization and depending on the number of subsamples selected (e.g., range from 200 to 800), the proposal extracts more or less the same number of patterns. More recently, in 2018, the OCLEP+, a modified version of OCLEP, was presented [51]. The main change does not rely on the mining stage, but in the classification stage for that reason, we will explain this difference in Section 2.3. An important drawback of OCLEP is that the mining stage proposed [33] is complex to understand, and the source code is not available.

In 2007, DEP was introduced as an algorithm for mining EPs in class imbalance problems [5]. DEP creates balanced subsamples containing all the objects from the minority class and a subset of objects from the majority class. Then, from each subsample, the EPs for the minority class are extracted using an exhaustive-search-based algorithm, which was not mentioned by the authors. In this way, many EPs for the minority class are extracted, and consequently,



they are not overwhelmed by the number of EPs from the majority class. DEP has the same drawback presented by EPRC when a resampling method is used. DEP could extract non-representative patterns of the problem since these could be EPs for only the subsample used instead of all training dataset.

In the same year (2007), an algorithm for discovering Jumping Emerging Patterns with Negation (JEPNs) was proposed [175]. A JEP is labeled as *jumping emerging pattern with negation* when it contains items with negated values, which were not extracted from the training dataset, i.e., a negated value represents the idea that such a value does not appear in an object. From this, Terlecki and Walczak argued that a *negated pattern* is obtained by changing each positive item of a given pattern to the corresponding negative item and vice versa [177]. The authors of [175] used JEPProducer [107] for mining JEPs and JEPNs from many databases. The authors showed that negative knowledge could be a useful addition to solutions based on positive patterns. The main drawback of using JEPNs is that the output model is more difficult to understand by an expert.

After, in 2008, the authors of [176] proposed (Top- $k$  JEPs) an algorithm for discovering emerging patterns. Top- $k$  JEPs is a modification of the algorithm proposed by [59]. The authors argued that those SJEPs with high support are the most discriminative patterns; as a consequence, they proposed a method for only keeping those  $k$  patterns with the highest supports. For doing that, the authors proposed to use the same mining strategy proposed by [59] but using new pruning conditions. The two pruning conditions proposed by Top- $k$  JEPs are the following. First, for pruning, on each node, it checks if the pattern associated with the node is not considered  $D$ -discernibility minimal. Let  $p$  be the pattern associated to the node,  $p$  is  $D$ -discernibility minimal if and only if:

$$\forall_{z \subset PS} support(p, \bar{C}_i, C_i) < support(z, \bar{C}_i, C_i) \quad (1)$$

In the above equation,  $support(p, \bar{C}_i, C_i)$  denotes the support of the pattern  $p$  for the negative class. Usually, in pattern discovery, the pattern's positive class refers to the class where that pattern covers more objects regarding the remaining classes (or negative class),  $\bar{C}_i$ . The second condition is by using a minimum support threshold for pruning. The main idea is to obtain the best  $k$  patterns much faster than SJEP but using the

same tree structure. The authors proposed to dynamically grow the threshold by storing the patterns in a heap by non-ascending support. Then if the size of the heap is equal to  $k$ , the support threshold is raised to the support of the first element of the queue plus one. The main handicap of Top- $k$  JEPs is that it continues using an initial discretization procedure before extracting patterns.

Later, during the same year (2008), an algorithm for mining patterns, labeled by authors as: jumping emerging pattern with occurrence count, was introduced [97]. The main idea of this proposal is to remove the initial discretization presented by the algorithms mentioned above for mining patterns. The authors showed that their proposal allowed to improve significantly the accuracy of a pattern-based classifier, in a particular image classification task, regarding another algorithm for mining patterns using the initial discretization. The main drawback of this proposal is that the number of extracted patterns increased significantly for those using initial discretization; in some cases, the number of patterns increased from 9 to 14,444. This result goes against the position of creating an understandable model for the experts.

In 2014, a novel approach for mining strong jumping emerging patterns based on BSC-tree (DGCP-Tree) was proposed [115]. The authors proposed a new data structure called as: Dynamically Growing Contrast Pattern Tree (DGCP-Tree), which is inspired by the BSC-Tree structure proposed by [92] for mining association rules. DGCP-Tree allows storing bit strings, each one representing the objects covered by an item. The main advantage of this method is that it allows mining emerging patterns during tree induction. For doing that, first, items with a length of value one are mined for each class. After, they are sorted by using the Growth Rate as the quality measure for patterns. Then, a procedure for growing the tree is executed by copying those right sibling nodes as the children of the node that is being processed, if they cover new objects and their support is greater than the threshold. The authors showed in their experiments that DGCP-Tree allows improving the computational time for extracting EPs and they allow improving the classification results regarding other well-known algorithms for mining EPs. Although it is true that DGCP-Tree improves the computational complexity, its main drawback is the initial discretization of all numerical features.

After, in 2015, an algorithm for mining strong emerging patterns in streamwise features (DFP-SEPSF) was proposed [4]. An emerging pattern  $ep$  having  $n$  items is labeled as Strong Emerging Pattern (SEP) when it satisfies the following conditions:

(i)  $support(ep, C_i, \bar{C}_i) \geq (k^n) / |C_i|$  and (ii) Growth Rate  $\gg 1$ ; where  $k$  the average number of values for an item and  $C_i$  is a class. As the length of an EP is limited by the support constraint then, its length is at most  $l_{max} = \log(|D|) / \log(k)$ . Consequently, the mining process does not extract SEPs with length greater than  $l_{max}$ . Usually, the training dataset contains all the feature set available before mining pattern. However, in some practical scenario, the number of training examples is fixed while the number of features grows, which is labeled as *streamwise features* [206]. Exhaustive-search-based algorithms are not suitable for working in problems with streamwise features. For that reason, the authors of [4] proposed to a bottom-up approach to constructing an Unordered Dynamic Frequent Pattern tree (UDFP-tree) and an Ordered Dynamic Frequent Pattern tree (ODFP-tree). As each pattern can be extracted from a FP-tree structure, the authors suggested applying three steps sequentially to convert the FP-tree in an UDFP-tree for managing streamwise features. First, some constraints are applied when a feature arrives. If a feature value is satisfied by one of the constraints, then it is removed. Otherwise, the feature value is transformed into several new nodes. In this process, new nodes are added to external nodes as their parents. Then, an updating process determines which nodes of the next feature are added to which external nodes. Finally, patterns are extracted by making a conditional database for each item in the header table. The conditional database is a sub-database or a sub-dataset which consists of a set of frequent items occurring with the same suffix pattern. The conditional database contains several paths which are created by the following item's node-links. A path is started at a node and ended up at the root for each node-link. In the experimental result, the authors of [4] showed that DFP-SEPSF allows obtaining better classification results than four other pattern-based algorithms and four other popular algorithms not based on CPs using 30 databases.

In 2016, WBEPM miner, an algorithm for mining EPs from imbalanced data streams was proposed

by [35]. The authors proposed a new type of pattern labeled as *Balanced Emerging Pattern* (BEP), which has a better adaptability on the imbalanced datasets. A pattern  $p$  is considered as bep if and only if:

$$support(p, C_i, \bar{C}_i) > \theta \text{ and:} \\ \delta \leq \frac{a * a^{k * support(p, \bar{C}_i, C_i)}}{support(p, C_i, \bar{C}_i)} \quad (2)$$

In the above equation,  $k$  is the balance factor,  $\delta$  the minimum contrast coefficient,  $a$  is the correlative correction parameter,  $\theta$  is the minimum support threshold, and  $support(p, C_i, \bar{C}_i)$  and  $support(p, \bar{C}_i, C_i)$  are the support of the pattern  $p$  for the classes  $C_i$  and  $\bar{C}_i$ , respectively. Similar to DEP, WBEPM creates several balanced subsamples but using a sliding window mechanism. EPs are extracted from each subsample by using a variant of the algorithm proposed by [56] (EJEPS). The experimental results show that the EPs mined by WBEPM attain better classification results than those obtained by EJEPS. Nevertheless, the WBEPM miner was proposed for working with imbalanced data streams, and it is not able to work with other problems. Also, WBEPM continues using an exhaustive-search-based algorithms for mining EPs, which needs an initial discretization of all numerical features.

Also in 2016, the authors of [114] introduced a single-scan algorithm for mining diverse types of correlation patterns. The authors introduced a new type of correlation pattern, labeled as pan-correlation patterns, to maximize the sequence of coherent data movements in one pattern. A *Pan-Correlation Pattern* (PCP) consists of a maximized sub-list of variables, where all the listed variables are associated with a segment of time points having the same length. The authors use an initial discretization of data for converting the pan-correlation mining problem into a sequential pattern mining problem. After that, the authors use a generalized representation of positive patterns and the opposite-mirror copy of the original sequential data set for mining patterns using an exhaustive-search-based algorithm, which the authors did not mention. As was stated in [113], these patterns are meant to work with unsupervised data where the features are measurements at different instants of

time (e.g., the measurements of sensors at different moments in time).

In 2016, an algorithm for mining patterns in data streams using reconfigurable hardware (SysTreeMiner) was proposed by [23]. SysTreeMiner uses a Landmark Window Model [36] and reconfigurable hardware [39] for mining several times a top-1 pattern from data streams. This proposal uses an algorithm similar to proposed by [176] (Top-k JEPs) for obtaining the top-1 pattern at each window of the data streams. The main advantage of this proposal is that it simultaneously uses Depth First Search (DFS) and Breadth First Search (BFS) traversals for analyzing the systolic tree structure, which allows a high level of parallelism for mining the top-1 pattern from each sliding windows. In the experimental results, the authors showed how the hardware architecture allows extracting all patterns correctly from data streams with a significant speed-up over other software-based implementations.

In 2017, an algorithm for mining patterns on data streams using hashing and a lexicographic order in hardware (LexOrdMiner) was proposed [25]. LexOrdMiner is a modification of SysTreeMiner, which includes a hashing procedure and lexicographic order for dealing with such cases where the number of items is large and objects, coming from a transactional data source, are short. The architecture proposed by LexOrdMiner performs one order of magnitude faster than other popular software-based baseline algorithms. Although SysTreeMiner outperforms LexOrdMiner for classification.

Later, in 2018, the authors of [24] proposed to extend the SysTreeMiner algorithm. This extension includes a new algorithm for the sliding window model and pre-processing stage. The authors showed how their proposal can improve the mining process when a device with no resource restrictions is used. As a consequence, their proposal can handle pattern with many items. Also, the authors proposed new parallel algorithms for mining patterns on data streams. None of the proposals of [24] compared their results against the algorithm (EPDSminer) proposed by [7] in the same context.

After, in the same year in (2018), a modification of the Apriori Algorithm (D2P-Apriori). D2P-Apriori modifies the Apriori algorithm for satis-

fying the high-performance requirement by using GPUs was proposed [188]. The authors proposed to use a dynamic bitmap queue data structure to avoid starting a CUDA kernel for redundant times and the Graph-join way is designed in parallel to generate candidates to realize a better performance than state-of-the-art serial implementations. The authors showed how D2P-Apriori improves other three state-of-the-art serial implementations. Nevertheless, they did not compare their results against the algorithms proposed by [24] in the same context.

In 2019, an algorithm for mining patterns on data Stream (FCI-Outlier) was introduced [83]. The authors proposed to apply FCI-Outlier in the outlier context. For doing this, first, all maximal patterns are mined using the CLOSET algorithm proposed by [149]. After that, FCI-Outlier defines three outlier factors to measure the abnormal degree of each object. In their experiment, the authors showed that FCI-Outlier improves results obtained by other three state-of-the-art algorithms. Nevertheless, FCI-Outlier presents more computational cost than others compared algorithms, and it is not clear why the authors proposed to use CLOSET instead of the improved version CLOSET+ [185].

In 2019, HashEclat was proposed [199] as a modification of the Eclat algorithm proposed by [198]. The main motivation of [199] is that Eclat is inefficient for computing the intersection sizes of itemsets. As a consequence, the authors of [199] proposed to calculate the approximate set intersection by using hash functions [178]. The authors selected MinHash proposed by [21], which can be used to estimate the similarity between two sets quickly. HashEclat adjusts some input values (like the minimum support threshold) to accelerate the speed of the mining stage, but it sacrifices the number of patterns mined. As a consequence, it is reflected in their classification results, which are worse than those achieved by Eclat. The authors used only seven datasets for testing their proposal, which is very poor for testing a pattern mining algorithm.

In 2019, an algorithm for mining the top-k patterns from large-scale data streams (Floating Top-k) was proposed [169]. A data stream is labeled as *large-scale data stream* when it contains a large number of distinct objects, and it is generally expensive, or

even infeasible, to maintain a counter for each distinct object at some time interval. Mining patterns from this type of problem is challenging nowadays due to the reasons mentioned above (a.k.a. *Windowed Top-k Frequent Items*). For example, one may want to query, in real time, about the top-10 most frequent hashtags in the past hour. The authors of [169] proposed to manage all window sizes dynamically within an upper bound  $W$ . The authors proposed the function *aggregate value* for grouping the items, from any windows  $w \in W$ , in a probabilistic way, i.e., items are grouped proportional to the total count of items in that window. For example, suppose that  $(I_1, I_2, I_3, I_2, I_3, I_1, I_2, I_4)$  are items arriving in increasing time order for  $|w| = 9$ , which are given the random levels  $(2, 3, 1, 0, 3, 1, 4, 2)$ , respectively. Then, the aggregate values of  $(I_1, I_2, I_3, I_4)$  (i.e., maximum levels) would be  $(2, 4, 3, 2)$  for  $w$ . Therefore,  $(2, 4, 3, 2)$  is an approximate indication of the frequency of the items and is suitable for obtaining the top-k frequent items. After that, the top-k frequent items are mined by using a reverse chronological order and maintain a vector of  $k$  tuples that have the highest levels for each  $w \in W$ . Floating Top-k have a  $O(\log(n))$  space complexity making it highly scalable for high-rate data streams with dynamic items and arbitrary-size windows. The authors showed how Floating Top-k improves other well-known algorithms for mining patterns from data streams. Nevertheless, Floating Top-k was not compared against the algorithms proposed by [23–25] in this context.

Recently, in 2019, the authors of [26] proposed an in-depth analysis of the LexOrdMiner algorithm proposed in [25]. In this analysis, the authors extended the literature review to include several works about mining frequent itemsets from data streams, which were not included in [25]. Also, they showed a detailed explanation of LexOrdMiner, including examples and new figures. Finally, the authors performed new experiments to demonstrate the viability of LexOrdMiner by using many databases and comparing it with other popular hardware-based baseline algorithms for mining patterns. Nevertheless, the authors did not compare LexOrdMiner against D2P-Apriori or FCI-Outlier, which were also proposed for working with data streams.

As we have stated before, many exhaustive-search-based algorithms rely on a tree structure (see Algorithm 1 for a general pseudocode as the one in [73, 86]) for mining CPs.

---

**Algorithm 1:** General scheme for mining contrast patterns using a tree structure.

---

**input** :  $D$ - a training dataset,  $q$ - a quality measure for patterns (usually the Growth Rate [49] measure is used),  $t$ - a tree traversal strategy, and  $p$ - a pruning strategy.

**output:**  $PS$ - a set of contrast patterns.

$R \leftarrow$  Initialize a null root node.

**foreach**  $o \in D$  **do**

Sort the items of  $o$  according to  $q$ .

Add  $o$  to  $R$  as nodes in a depth-first way;

**foreach** *existing node* in  $R$  **do**

Update the counter of occurrences;

**end**

**end**

$PS \leftarrow$  The set of contrast patterns mined using the tree traversal strategy  $t$  and the pruning strategy  $p$ .

**return**  $PS$

---

Exhaustive-search-based algorithms for mining patterns is the foundation for stating the pattern-based classification. They began the idea of obtaining understandable models for experts. Nevertheless, exhaustive-search-based algorithms present as the main drawback the initial discretization of all numeric features. As many authors argued [70, 71, 121, 124, 127], the initial discretization is not a feasible solution because of information loss, the limited possibility of transformed data interpretation, and the computational time required for carrying out this task. It is important to highlight that discretizing a numerical feature without considering the values of other features could hide important relations in the objects of a class, producing undesirable results [70, 71]. For example, the authors of [71] showed how the exhaustive-search-based algorithm proposed by [59], which applies an a priori discretization, is unable to find patterns in

the wpbc database; taken from UCI Machine Learning Repository [53].

Another drawback of the exhaustive-search-based algorithms for mining patterns is that only extract patterns having items of the form  $[f_i = v_j]$ ; in this way, they are subtracting patterns' discriminative power by not using other remaining relational operators, like  $\{\in, \notin, \neq, \leq, >\}$ . As a consequence, for avoiding these drawbacks, several CP mining algorithms based on decision trees were introduced.

### 2.1.2 Decision tree-based algorithms for mining contrast patterns

Mining CPs from decision trees can be better than using exhaustive-search-based algorithms for three reasons. First, the local discretization performed by decision tree with numeric features has proved better results than using a priori global discretization [70, 71]. Second, decision trees contain a small proportion of candidate features even in longer tree paths, which significantly reduces the search space of potential patterns and generate a small collection of high-quality patterns [70, 71, 156]. Third, CP mining algorithms based on decision trees can handle missing values by introducing a penalizing factor in the measure for evaluating candidate splits [71].

As we have stated in Section 2.1.1, mining CPs using exhaustive-search-based algorithms is a challenging problem because of the high-computational cost due to the exponential number of candidate patterns [61, 81, 86, 172, 197]. For example, as was stated by [187], several exhaustive-search-based algorithms for mining EPs have a computational cost equal to  $O((n * m)^2)$  where  $n$  and  $m$  are the number of objects and number of features in the training dataset, respectively. However, notice that those mining algorithms based on decision trees obtain a small collection of high-quality CPs, which reduces the computational cost significantly [70, 71, 127]. In this way, mining CPs from decision trees takes  $O(t * m * n \log_2(n))$ , where  $n$  and  $m$  are the number of objects and the number of features, respectively, in the training dataset and  $t$  is the number of decision trees to be built.

In our review, we found that algorithms based on decision trees for mining CPs follow two main steps. First, inducing many diverse decision trees, like the

traditional procedure proposed by [157], where a quality measure for evaluating splits and stop conditions are used. Second, extracting CPs from each induced decision tree, where each pattern contains items using all relational operators stated in Section 1, which coming from a path from the root node to a leaf node; i.e., every path from the root to a leaf determines conjunction of items, which forms a pattern. For more detail, we provide algorithms 2 and 3.

---

#### Algorithm 2: General scheme for mining contrast patterns from decision trees.

---

**input** : D- a database, K- number of decision trees to be induced, Z- a strategy for creating diversity.  
**output**: PS- a set of contrast patterns.  
 $PS \leftarrow \emptyset;$   
**while** Number of induced decision trees  $\leq K$  **do**  
    DT  $\leftarrow$  Build a decision tree by using the dataset D and the strategy Z;  
     $PS \leftarrow PS \cup \text{ExtractPatterns}(\text{DT.RootNode});$   
**end**  
**return** PS

---



---

#### Algorithm 3: ExtractPatterns - Recursive pattern extraction from a decision tree.

---

**input** : R- a decision tree node (Initially, the root node).  
**output**: PS- a set of CPs.  
 $PS \leftarrow \emptyset;$   
**foreach**  $child \in R.children$  **do**  
    **if**  $child$  is a leaf node **then**  
        Create a pattern  $p$  collecting the items from the root node to the  $child$  node;  
         $p.class \leftarrow$  assigning the class with more objects into the  $child$  node;  
        **if**  $p$  is contrast pattern **then**  
             $PS \cup \{p\};$   
        **end**  
    **end**  
    **else**  
        ExtractPatterns( $child$ );  
    **end**  
**end**  
**return** PS

---

As far as we know, the first contrast pattern mining algorithm based on decision trees was proposed

by [71], which was labeled as Logical Complex Mine (LCMine). LCMine is based on decision trees by using a deterministic procedure. LCMine generates diversity by using a set of vectors ranging from  $\vec{v} = (1, 1, \dots, 1)$  to  $\vec{v} = (k_1, k_2, \dots, k_m)$ , each one of a different decision tree. In this way, LCMine selects the best  $k$  splits in first levels and the best split in lower levels of the decision tree. For example, for  $k = \{5, 4, 3, 2\}$ , LCMine creates  $(5 \cdot 4 \cdot 3 \cdot 2) = 120$  different decision trees, from which patterns are extracted. LCMine allows obtaining better classification results than other exhaustive-search-based algorithms.

Later, in the same year (2010), an algorithm for mining EPs (CEPM), which improves LCMine, was proposed by [67]. CEPM induces several decision trees, similar to LCMine, but the measure used for evaluating splitting criteria (Information Gain proposed by [157]) is weighted taking into account each object of the training dataset. In other words, CEPM uses a boosting approach for weighing the objects after each tree induction. The authors of [67] proposed to extract emerging patterns when the candidate splits are generated if a condition is met. The condition claims that a pattern can be extracted from any candidate split that generates nodes with at least  $\mu$  objects in one class and at most one object in the remaining classes. After that, the best  $k$  splits are expanded, updating object weights after each induction. CEPM extract a collection of high-quality EPs, which allow attaining better classification than other popular state-of-the-art classifiers.

During the same year (2010), the algorithm EPRFm was proposed by [186], which extracts CPs from the decision trees created with Random Forest [20]. For inducing decision trees, EPRFm selects a random subset of features at each node in the decision tree as proposed by [20]. Then, the best feature of the selected subset is used to build the node. The authors showed that EPRFm obtains a collection of high-quality patterns, which allows good classification results using a video database.

After, in 2011, FEPM was introduced [68] as an algorithm for mining a new kind of pattern, named Fuzzy Emerging Pattern (FEP). A *fuzzy pattern* is a pattern containing conjunctions of selectors [ $Feature \in FuzzySet$ ], where  $\in$  is the membership of the feature value to FuzzySet. This way, an object satisfies a given pattern to a certain degree according to the degree the object feature values

satisfy the item expressed in the pattern. For example, [ $Temperature \in hot$ ]  $\wedge$  [ $Humidity \in normal$ ] is a fuzzy pattern describing the weather in a fuzzy domain. For mining fuzzy patterns, first, this procedure creates a fuzzification for all features. For non-numeric features, a collection of singleton fuzzy sets is created, i.e., for each different value, a fuzzy set having membership 1 for that value, and 0 for the remaining values is created. For numeric features, a traditional fuzzification method is applied. After that, authors use a fuzzy variant of the ID3 method [155] for building a set of different fuzzy decision trees, from where several fuzzy patterns are extracted. Authors showed in their experiment that fuzzy patterns allow obtaining better classification results than other popular state-of-the-art classifiers and SJEP. Nevertheless, the authors did not compare their proposal against LCMine. Fuzzy patterns allow describing the output model in a language closer to experts than those models described by non-fuzzy patterns; nevertheless, this approach has been little studied.

In 2015, Delete Best Feature (DBF) for mining CPs was introduced [70]. DBF removes from the training samples those features used in the root node of previously induced trees until no feature remains. The main idea of DBF is to alleviate the common problem that very discriminant items are contained in most of the mined patterns, creating many duplicate patterns. Nevertheless, in those databases with few features, DBF generates a small set of decision trees, which sometimes is not enough for extracting high-quality patterns. Also, in some databases, DBF produces models that allow obtaining lower classification results regarding other popular state-of-the-art classifiers, like the C4.5 classifier [157].

In the same paper mentioned above, the authors proposed another algorithm for mining CPs, which was labeled as Delete Best Property (DBP) [70]. DBP relaxes the limitation presented in DBF by deleting the whole feature while avoiding the repetition of a very discriminative item in many resultant patterns. In short, DBP forbids the items appearing in every root node to appear in remaining induced trees. The main drawback of DBP is that in databases with numerical features, it produces a second best item almost identical to the best one, but with a slightly different cut point. As a consequence, a collection of similar patterns is obtained by using this procedure.

For solving the drawback presented by DBF and DBP, the authors of [70] proposed a new algorithm for mining CPs which the authors named as Delete Best Property by Level (DBPL). The idea is similar to DBP, but it forbids a given item to appear in the future trees only in the tree level that it appeared in the previously generated trees. In this way, an item is forbidden at a level only if it has been previously forbidden at the child level, avoiding items in child nodes to be forbidden too fast. DBPL allows improving the results obtained by using DBF and DBP, but it does not improve other proposals of the same authors.

In 2015, the authors of [70] proposed RFm, an algorithm similar to the one proposed by [186] for mining EPs using the Random Forest algorithm proposed by [20]. The main difference is that RFm selects a random subset of features with size  $\log_2 |Features|$  at each node in the decision tree. Then, the best feature of the selected subset is used to build the node. Although in the original papers, proposed by [20] and [186], each forest is built based on a bagged version of the training set, the authors of [70] did not consider this procedure for avoiding hidden dependencies in their result. In [70], RFm obtained a collection of high-quality patterns, which allows better classification results than those patterns obtained by DBF, DBP, and DBPL. Nevertheless, RFm was not compared with EPRFm.

Another proposal of [70] is to use the decision tree-based bagging algorithm (Bagging) [19] for extracting CPs. This procedure creates diversity by building each decision tree with a bootstrap replicate of the training set. Bootstrap replicates are built by randomly sampling the training set with replacement, usually the 66.6% of the training dataset until an equal number of instances as the training set is obtained. Based on this procedure, the extracted CPs allow one to obtain better classification results than DBF, DBP, and DBPL. However, it does not improve the results obtained by using RFm.

Furthermore, the authors of [70] proposed to use the Random Subspaces algorithm [12] for extracting CPs. This procedure is similar to Random Forest, but it selects a random feature subset to build each decision tree. Authors claim that for extracting CPs using this procedure, the best size for the feature subset should be  $\frac{|Features|}{2}$ . This procedure for mining CPs allows obtaining better classification results than DBF, DBP,

and DBPL but it did not improve the results attained by Bagging and RFm.

The last proposal of [70] is to extract CPs from Random Split, a procedure for building decision trees proposed by [43]. Random Split is similar to other proposal mentioned above, but it selects one of the best 20 candidates splits at each node of the decision tree. This proposal for extracting CPs allows obtaining better classification results than DBF, DBP, and DBPL but it did not improve the results attained by Random Subspaces, Bagging, and RFm.

In 2016, SMOTE-TL+LCMine was proposed [124] as an algorithm for mining CPs in class imbalance problems. The authors conducted an in-depth study of the impact of resampling methods for cp-based classifiers in imbalanced databases. From this study, the authors proposed to use the resampling method SMOTE-TL, at the data level, before applying LCMine for extracting CPs. The authors showed in their experiments that LCMine is affected by imbalanced databases. Also, the authors showed how not all resampling methods allow improving the results of LCMine when they are applied before extracting CPs.

In 2017, Hellinger Random Forest (HRFm) was introduced [127] as an algorithm for mining contrast pattern in class imbalance problems. The authors proposed to modify the RFm [70] for building decision trees using the Hellinger distance [37] as the measure for evaluating splitting criteria instead of the traditional Information Gain measure [157]. The authors claim that the Hellinger distance is unaffected by the class imbalance problem because it rewards those candidate splits that maximize the TPR (True Positive Rate) while minimizing the FPR (False Positive Rate). Authors argued that using this proposal, CPs of the minority class do not become overwhelmed by CPs from the majority class. HRFm extracts contrast pattern that allows attaining better classification results than several popular state-of-the-art classifiers, based and not based on patterns, which are designed for class imbalance problems. The main drawback of HRFm is that only work with two-class problems due to the limitation of the selected version of Hellinger distance, which only evaluates candidate splits for two-class problems.

In 2017, EvAEP, an evolutionary fuzzy system for mining EPs was proposed by [75]. EvAEP relies on the hybridization between a fuzzy system and a learning process based on evolutionary

computation. EvAEP uses an evolutionary algorithm with a codification “chromosome = Rule” where only the antecedent part of the rule is represented for obtaining items similar to those formed by patterns. For numerical features, EvAEP uses a fuzzy representation with fuzzy sets composed by linguistic labels defined with uniform triangle forms. EvAEP relies on a mono-objective approach with an iterative rule learning, obtaining for each class the best individuals in an iterative process. In this way, EvAEP extracts EPs until a non-emerging pattern is obtained. EvAEP stops if all objects for the class are covered by the patterns extracted previously or a pattern with null support is obtained. The main drawback of EvAEP is that it is well-known that evolutionary-based approaches present a high computational complexity [145].

In 2017, the authors of [74] proposed EvAEFP-Spark, which is a modification of EvAEFP for working with big data problems. EvAEFP-Spark only uses the MapReduce paradigm [41] for alleviating the computational complexity presented by EvAEFP. EvAEFP-Spark uses the advantage of the MapReduce approach where the population on a given generation is sent to the partitions, and for each individual, a partial confusion matrix is calculated. After that, EvAEFP-Spark summarizes all individual’s confusion matrices to get the final confusion matrix. Although the results obtained by EvAEFP-Spark improves those obtained by EvAEFP for big data problems, this proposal seems straightforward.

In 2018, MOEA-EFEP, a multi-objective evolutionary algorithm for extracting fuzzy emerging patterns was proposed [72]. MOEA-EFEP uses a similar approach to presented by EvAEFP, where a “chromosome=rule” approach including both the antecedent and consequent of the rule to extract knowledge for all the classes in a single execution is used. In a similar way to EvAEFP, MOEA-EFEP uses fuzzy logic for the representation of numerical features by means of linguistic labels. The main differences of MOEA-EFEP against EvAEFP is that it uses the consequent of rules for extracting items. Also, MOEA-EFEP uses three filtering methods to keep only high-quality patterns, minimal EPs, maximal EPs, and EPs with confidence higher than 60% (these filtering methods will be explained in Section 2.2). The authors showed in their experiment that MOEA-EFEP improves EvAEFP and FEPM using many databases. It is important to high-

light that the authors did not comment on how their proposal converts the fuzzy rule into fuzzy emerging patterns.

In some practical cases, it is incorrect to evaluate the classifier by taking into account only the number of cases which are correctly classified because of the cost of misclassification for each class is different. This type of problem is known as cost-sensitive problems where a cost matrix governs cost-sensitive problems, and as a consequence, this type of problem cannot be compared against non-cost-sensitive problems [95]. For example, in a two-class problem using the cost-sensitive approach, the classification of a query object is optimal for the class  $C_i$  if and only if the misclassification cost is less than the misclassification cost of classifying the same object in the other class  $\bar{C}_i$ .

Based on this approach, in 2019, an algorithm for discovering cost-sensitive patterns in class imbalance problems was introduced by [130]. A *Cost-Sensitive Pattern* (CSP) is a type of pattern whose support, in each class, is multiplied by the corresponding cost in the class coming from a cost matrix. In this way, a CSP has a cost associated with each class, and it should be interpreted as the misclassification cost for each class. The authors proposed to extract a collection of CSPs by using a variant of RFm proposed by [70] but using a new measure for evaluating splitting criteria, which takes into account the different cost associated to each problem’s class. In the experiments, the authors have shown how their proposal allows obtaining lower misclassification cost than other well-known classifiers (based and not based on CPs) combined with Meta-cost [45]. The main handicap of mining CSPs is that it relies on a cost matrix, which commonly should be provided by experts in the application domain.

In some datasets, the classes cannot be easily separated using CPs with univariate items. Based on this fact, recently, [27] proposed an algorithm for mining multivariate CPs based on multivariate decision trees. A *Multivariate Contrast Pattern* (MCP) is a pattern represented by conjunctions of items, where items can be either univariate items (stated in Section 1) or multivariate items. A *multivariate item* allows linear combinations of numerical features of the form  $\sum w_i f_i \leq v_j$  or  $\sum w_i f_i > v_j$ , where  $w_i \in \mathfrak{R}$ ,  $\forall i \in \{1, \dots, k\}$ . For mining MCPs, the authors modified the HRFm algorithm proposed by [127] for working with multi-class problems and for inducing multivariate



decision trees using oblique splits [85], which refers the oblique hyperplane used, or multivariate splits [22] coming from the use of multiple features in the split. In this way, their proposal extracts a collection of MCPs that allows obtaining better classification results than HRFm and other popular state-of-the-art classifiers not based on patterns. Nevertheless, the authors did not show any comparison with the proposal of [70] for mining fuzzy CPs, which seems to be a suitable comparison because both approaches can generate patterns containing not traditional items.

From our review of the algorithms for mining CPs based on decision trees, we see that those proposals based on decision trees extract a small collection of high-quality CPs that allows obtaining better classification results than those obtained with exhaustive-search-based algorithms. We also notice that those CP mining algorithms based on resampling methods, as well as boosting and bagging algorithms, modify the original dataset; as a result, the extracted patterns could be fictitious or they may not cover any object of the original dataset. Consequently, these patterns could not be useful for explaining the results in terms of the original problem's classes.

Furthermore, from this section, we can see that the algorithms for mining fuzzy CPs and mining MCPs provide a different way for solving the limitation of the traditional form for inducing decision trees, which only builds splits involving a single feature. Consequently, an interesting window to explore is

to develop algorithms for mining fuzzy multivariate patterns.

Finally, in Fig. 2, we show the relationships among the types of patterns most used throughout our review, which were stated in this section. Note that for a dataset, a set of patterns can be extracted from which only a subset is considered CPs. From CPs, a set of minimal and maximal CPs are identified, and after that, there exist different types of CPs such as FEPs, MCPs, and CSPs.

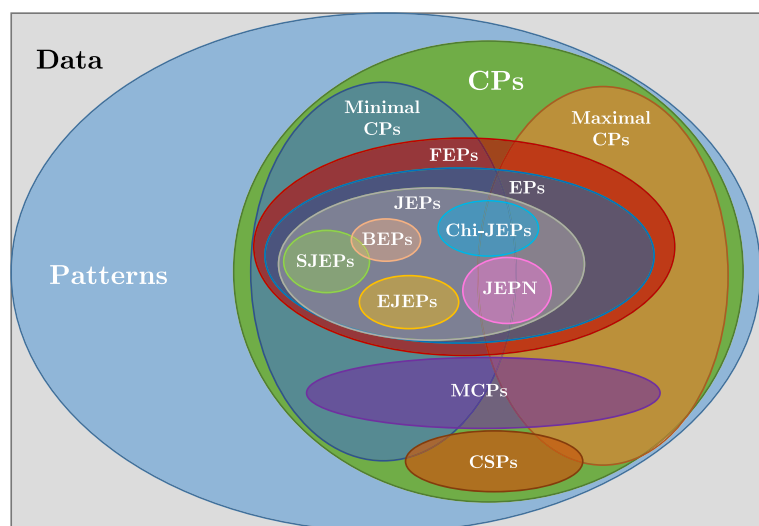
## 2.2 Filtering Contrast Patterns

Commonly, algorithms for mining CPs extract a large set of patterns from a training dataset. Therefore, an important task is to select those patterns with high discriminative ability for supervised classification [56, 57, 69, 70, 121, 124, 127, 187]. To carry out this task, many patterns filtering algorithms have been proposed on in the literature. Based on our review, we split this section into two: Section 2.2.1 algorithms based on set theory and Section 2.2.2 algorithms based on quality measures for patterns.

### 2.2.1 Based on set theory

Many authors [56, 57, 69, 187] propose to eliminate duplicate and specific patterns, as well as removing redundant items from patterns, to obtain a collection of patterns having a high discriminative ability.

**Fig. 2** A graph for representing the relationships among different types of patterns and their coverage for a set of hypothetical data



A basic strategy for obtaining patterns having fewer items is to remove redundant items. The main idea is to compare items for finding generalizations of them and, as a result, obtaining short patterns. An item  $I_1$  is more general than another item  $I_2$  if all objects fulfilling  $I_1$  also fulfill  $I_2$ , but not all objects fulfilling  $I_2$  fulfill  $I_1$ ; in other words,  $I_2$  is redundant with  $I_1$ . If two items in a pattern are redundant, the most general item is eliminated. An example of a pattern having redundant items is:  $[Age > 37] \wedge [Age > 40]$ , which is simplified to  $[Age > 40]$ ; since persons older than 40 are also older than 37.

Another filtering strategy is to remove duplicate patterns. Since patterns are extracted from several decision trees by using the same training dataset or using resampling approaches, many patterns containing the same items and covering the same objects (a.k.a. *duplicate patterns*) can be extracted. In order to reduce the size of the outcome, only one pattern is selected from those containing the same items while covering the same objects.

A strategy widely used is to remove specific patterns (a.k.a. *removing maximal patterns*). A *maximal pattern* is a pattern whose super-patterns are no longer CPs. Usually, maximal patterns contain several items, as a consequence, they are very specific and cover only a few objects. Let  $P_1$  and  $P_2$  two patterns from the same class,  $P_1$  is more specific than  $P_2$  if  $P_1$  contains all the items in  $P_2$  and at least one more. For example, let  $P_1 = [Age \leq 37] \wedge [Color = White] \wedge [Cuban = True]$  and  $P_2 = [Age \leq 37] \wedge [Color = White]$  be two patterns from the same class. Since all the items belonging to  $P_2$  also belong to  $P_1$  but  $P_1$  has one more item, then  $P_1$  is more specific. Therefore, as  $P_1$  is more specific than  $P_2$  and, both are patterns from the same class, then  $P_1$  should be removed.

An alternative for removing specific multivariate CPs was proposed by [27], which generalizes the univariate item relations to multivariate items. Let be  $I_1$  and  $I_2$  two multivariate items containing the same relational operator ( $\leq$  or  $>$ ), and the same features having non-zero coefficients ( $w_i \neq 0$ ). Then, a ratio ( $r$ ) between the corresponding feature coefficients of  $I_1$  and  $I_2$  is computed, and if the difference is less than an acceptable error  $\epsilon$ , then  $I_2$  is multiplied by the ratio  $r$  and after, the same filtering strategy mentioned before for univariate items is applied.

The aforementioned strategy for removing specific patterns was used by [56] with the aim of obtaining

a collection of minimal patterns. A *minimal pattern* is defined as a CP whose sub-patterns are not CP anymore. This type of patterns is the most general cp. Minimal patterns are interesting for describing a model because generally they contain a low number of items and higher support [56, 57, 69, 187].

It is important to highlight that the algorithms mentioned above are the basis for building other filtering algorithms based on set theory [56, 57]. For example, Essential Jumping Emerging Patterns (EJEPs) or Strong Jumping Emerging Patterns (SJEPs) are obtained by the intersection of the a set of minimal EPs and the set of JEPs [57].

The computation cost for all the filtering algorithms mentioned above is the following:  $O(n)$  for removing redundant items and  $O(n^2)$  for the remaining strategies, where  $n$  is the number of CPs.

### 2.2.2 Based on quality measures

Many pattern filtering algorithms are based on a quality measure for patterns. A *Quality Measure* (QM) assigns a higher value to a pattern when it better discriminates objects of a class from objects of other classes [10, 65, 121, 123]. Consequently, a QM allows generating a pattern ranking based on the discriminative power of the patterns, which can be used for selecting the best patterns for a pattern-based classifier [65, 90, 121, 123, 144]. We can say that a quality measure  $Q_1$  has *better behavior* than another quality measure  $Q_2$  if, at the classification stage, the patterns selected from the ranking induced by  $Q_1$  provide better classification result than those coming from the ranking induced by  $Q_2$ .

Based on previous studies [66, 76, 77, 123, 136], quality measures for patterns can be categorized into two groups:

**Objective:** They are based on probabilities and statistics. The aim is to evaluate the ability of a pattern for discriminating objects in a class from objects in other classes [136, 137].

**Subjective:** They are based on a subjective criterion issued by an expert in the application domain [111, 147].

Objective measures are the most used in algorithms for filtering CPs because they do not take into account neither the context of the application domain nor the

goals and background knowledge of experts [66, 77, 123]. Then, since subjective measures are based on a specific criterion issued by an expert in the application domain, which is not available in any repository, it is very complicated to develop pattern filtering methods based on subjective measures. As a consequence, as far as we know, there are no pattern filtering algorithms based on subjective measures.

An *objective quality measure* can be defined as a function  $q(p, C_i, \bar{C}_i) \rightarrow R$ , which assigns a higher value to a pattern  $p$  when it better discriminates objects in a class  $C_i$  from objects in the remaining problem classes  $\bar{C}_i$  (The classes form a partition of the universe  $D = C_i \cup \bar{C}_i, C_i \cap \bar{C}_i = \emptyset$ ) [66, 123].

There are several studies [65, 66, 121, 123] about the effect of QMs for CPs in both balanced and imbalanced problems. In these studies, the authors evaluated the behavior of more than 50 QMs and two pattern filtering algorithms ( $k$  best and covering) based on these QMs.

The first algorithm selects the  $k$ -best patterns by class from the ranking produced by applying a given QM. [65] proposed to select a percentage of all extracted pattern instead of using a fixed number because the number of extracted patterns varies a lot from a dataset to the others; as a consequence, selecting a fixed number of patterns could provide a small collection or, on the contrary, a big collection of patterns. [65] claimed that using  $k = 10\%$ , without taking into account the class, is suitable for obtaining good classification results. Nevertheless, [124] showed that selecting only 10% of CPs could lead to low accuracy at the classification stage, especially in class imbalance problems. Consequently, [124] proposed to select  $k = 10\%$ ,  $k = 50\%$ , and  $k = 80\%$  by class. The authors showed that using  $k = 50\%$  and  $k = 80\%$  allows better classification results than using  $k = 10\%$ .

The second algorithm selects a subset of the best patterns covering all the objects of the training sample. In this algorithm, for each object of the training sample, the best pattern covering the object is selected (only if this CP is associated with the same class that the object has, and this CP has not been previously selected).

Notice that both filtering algorithms have lower computational costs  $O(n)$  than those based on set theory. The pseudocodes for both pattern selection methods are shown in Algorithm 4 and Algorithm 5, respectively.

---

**Algorithm 4:** Algorithm for filtering the  $k$  best patterns.

---

**input :**  $P$ - a set of patterns,  $q$ - a quality measure,  $s$ - a parameter indicating selection by class,  $k$ - a number of patterns  
**output:**  $R$ - a set of selected patterns

```

 $R \leftarrow \emptyset$ 
if  $s == true$  then
  foreach  $c \in Classes$  do
     $PS \leftarrow$  patterns of  $c$  sorted using  $q$ 
    Select the  $k$  best patterns from  $PS$  and add them to  $R$ ;
  end
end
else
   $PS \leftarrow P$  sorted using  $q$ 
   $R \leftarrow$  Selecting the  $k$  best patterns from  $PS$ 
end
return  $R$ 

```

---



---

**Algorithm 5:** Algorithm for filtering patterns considering their covering.

---

**input :**  $P$ - a set of patterns,  $q$ - a quality measure,  $T$ - a training sample  
**output:**  $R$ - a set of selected patterns

```

 $PS \leftarrow P$  sorted using  $q$ 
 $R \leftarrow \emptyset$ 
foreach  $o \in T$  do
  Search for the first pattern  $p$  in  $PS$  that covers  $o$  and it is associated with the same class as  $o$ 
  if  $p \notin R$  then
     $R \leftarrow R \cup \{p\}$ 
  end
end
return  $R$ 

```

---

The main drawback of both methods ( $k$  best and covering) is that they output a collection of high-quality patterns taking into account the training dataset, but that collection could have low quality for objects in the testing dataset. Hence, the authors of [66] proposed to evaluate the quality measures for CPs by using query objects of the testing dataset. For doing that, in [66] proposed the following measure:

$$PQ(p, C_i, \bar{C}_i) = \frac{\text{support}(p, C_i, \bar{C}_i) + 0.01}{(\text{support}(p, C_i, \bar{C}_i) + \text{Support}(p, \bar{C}_i, C_i)) + 0.02} \quad (3)$$

where  $p$  is a pattern,  $C_i$  is the positive class, and  $\bar{C}_i$  the negative class (i.e., all other classes different to  $C_i$ ).

In [66] studied the correlation between the quality of the CPs extracted from the training dataset and the

quality of those extracted from the testing dataset. The main findings of this study were: (i) there are not QMs having an average correlation higher than 0.52, this is a small value; (ii) the behavior of a quality measure is highly dependent on the database; and (iii) most of the QMs have a very similar average correlation.

In [112] proposed two quality measures for assessing sets of CPs, which are assessing the diversity of a collection of CPs. The first one (IO) measures the number of items overlapping among the CPs. For IO, lower values are more desirable:

$$IO(PS) = \frac{2}{k(k+1)} \sum_{i=1}^{|PS|} \sum_{j=i+1}^{|PS|} ovi(p_i, p_j) \quad (4)$$

where  $PS$  is a set of contrast patterns and  $ovi$  measures the number of items shared by  $p_i$  and  $p_j$ .

The second proposal of [112] measures the number of objects covered by the set of CPs. For that measure, lower values are more desirable:

$$DO(PS) = \frac{2}{k(k+1)} \sum_{i=1}^{|PS|} \sum_{j=i+1}^{|PS|} ovo(p_i, p_j) \quad (5)$$

where  $PS$  is a set of contrast patterns and  $ovo$  measures the number of objects covered by  $p_i$  that are also covered by  $p_j$ .

The measure IO was developed for CPs extracted by using an initial discretization of the numeric features, which have items of the form  $[f_i = v_j]$ . Hence, IO is not suitable for CPs extracted from decision trees because they do not use an initial discretization of the numeric features; and consequently, they contain relational operator different to  $=$ , which are not straightforward for comparing. For example, it is not straightforward to decide that two patterns  $p_1 = [air\_pressure < 22.5]$  and  $p_2 = [air\_pressure < 22.4]$  are sharing the same item. Although the difference between their values is very small, it is highly dependent on the nature of the problem from the CPs were extracted.

From our review and several early studies [65, 66, 121, 123], we can conclude that there is no consensus about which is the best pattern filtering method between  $k$ -best and covering. Also, we can conclude that Jaccard [174] is the best QM for ranking patterns for supervised classification in class imbalance problems. On the other hand, Lift [77] is the best QM for

ranking patterns for supervised classification in balanced problems. Some authors [123] claim that QMs can be evaluated in isolation or combined with other measures by comparing their values with an estimated quality of CPs. Although, some authors [66] claim that combining two QMs can lead to better results than using their components and any other single QM.

From the related works in this section, we can conclude that pattern filtering algorithms have proven to reduce the number of patterns from the original collection, helping to obtain an understandable model. Nevertheless, the number of patterns selected by using these algorithms is not small enough (e.g., 10 patterns) to be both simple to understand and highly accurate for classification.

### 2.3 Supervised Classifiers Based on Contrast Patterns

After mining and filtering CPs, the next stage is classification. Algorithms for classification based on CPs are responsible for searching the best strategy for combining the information provided by a collection of CPs and so building accurate models based on CPs. Usually, the computational cost for pattern-based classifiers is  $O(n)$ , where  $n$  is the number of CPs covering a query object. Algorithm 6 shows a general scheme of supervised classification based on CPs; where, from now and on, we will call *scoring function* to the  $\sigma$  function and, *votes integration*, to the  $\phi$  function.

---

#### Algorithm 6: General scheme of supervised classification based on CPs.

---

**input** :  $PS$ - a set of CPs,  $C$ - a set of classes,  $o$ - a query object,  $\sigma$  - a scoring function,  $\phi$  - a function for integrating the votes per class.

**output**:  $C_i$  - a class belonging to  $C$ .

**foreach**  $C_i \in C$  **do**  
 $R[C] = \sigma(o, PS, C_i, C)$ , where  $\sigma$  is a function  
 such as  $\sum_{\substack{p \in PS, \\ p \text{ covers } o}} support(p, C_i, \bar{C}_i)$ ;

**end**

$C_i = \phi(R, C)$ , where  $\phi$  is a function such as  
 $\operatorname{argmax}_{C_i \in C} (R[C_i])$ ;

**return**  $C_i$

---

One of the first algorithms proposed for classification is based on a scoring function known as *Classification by Aggregating*. This function was

adopted by many cp-based classifiers, such as KORA-3 [17], KORA-Ω [184], EPRFm [186], LCMine [71], CEMP [67], and CAEP (Classification by Aggregating Emerging Patterns) proposed by [52]. This function uses the *support* quality measure [2, 49] for classifying a query object into the class having the highest sum of support. This sum is computed using all patterns covering the object to be classified. Let  $p$  be a pattern, and  $D$  be a dataset; then, the support of  $p$  is the fraction resulting from dividing the number of objects in  $D$  described by  $p$  by the total number of objects in  $D$  [49]. The mathematical form of this function appears in (6).

$$\sigma(o, PS, C_i, C) = \sum_{\substack{p \in PS, \\ p \text{ covers } o}} support(p, C_i, \bar{C}_i) \quad (6)$$

where  $support(p, C_i, \bar{C}_i)$  is the support in the class  $C_i$  of the CP  $p$  covering the query object  $o$ , and  $PS$  is the set of CPs for all the classes in  $C$ .

The main drawback of using a scoring function based only on the support for classification is that the support is a QM affected by the class imbalance problem [124]. As a consequence, classifiers based solely on support could bias their classification results toward the majority class.

In 2000, the authors of [201] proposed Information-Based Classification by Aggregating Emerging Patterns (iCAEP), a classification strategy based on EPs for dealing with large-volume high-dimensional datasets. iCAEP relies on two QMs for patterns to classify a query object using a collection of high-quality patterns. To do this, first, EPs are ranked in descending order according to their number of items (*Length* stated in [33]). For patterns having the same number of items, the Growth Rate [49] is used as a second ordering criterion. Then, for each class, iCAEP iteratively selects (according to the ranking) patterns, until all the features of the dataset appear in at least one item of the selected patterns. Finally, according to each subset of EPs, the query object is classified into the class having the highest sum of supports.

In 2003, the authors of [57] proposed BCEP, a classifier based on EPs and the Bayes theorem. A scoring function based on Bayes theorem assigns a class  $C_i$  to a query object  $o$  if it maximises (7):

$$P(C_i|o) = \frac{P(o, C_i)}{P(o)} = \frac{P(C_i)P(o|C_i)}{P(o)} \quad (7)$$

where  $P(a|b)$  denotes the conditional probability of  $a$  given  $b$  and probabilities are estimated from the training dataset.

Since it is very difficult in practice to calculate the probability  $P(o, C_i)$ , some approximations can be used. In this way, BCEP incorporates the use of a set of emerging patterns  $EP$  to derive a product approximation of  $P(o, C_i)$  using the chain rule of probability:

$$P(ep_1, ep_2, \dots, ep_n) = P(ep_1)P(ep_2|ep_1) \dots P(ep_n|ep_1, \dots, ep_{n-1})$$

where  $ep_i \in EP$ . Using this scoring function based on Bayes theorem, the authors showed that BCEP improves the results of the classification obtained by CAEP.

In 2006, a one-class classifier based on length statistics of EPs (OCLEP) was proposed [33]. In this type of classifiers, an object can be classified as outlier (masquerader) or normal. To classify an object  $o$ , first, OCLEP applies the function *BorderDiff*( $o, T$ ), as we stated in Section 2.1.1, to get a set  $P$  of minimal JEPs for  $o$  against  $T$ . If the dataset  $D$  is small, then  $T = D$  and  $avgLen(o) = avgLen(BorderDiff(o, D))$ . Otherwise, they proposed to use 20 random samples ( $T_1, \dots, T_{20}$ ) of  $D$ , and then  $avgLen(o) = \sum_{i=1}^{i=20} avgLen(BorderDiff(o, T_i))/20$ ; where, given a non-empty set  $P$  of patterns, let the average length be:

$$avgLen(o) = \frac{\sum_i |P_i| * i}{\sum_i |P_i|} \quad (8)$$

where  $|P_i|$  is the count of EPs with length  $i$  covering the query object  $o$ . Notice that the length of a pattern is computed as the number of items contained in that pattern.

After that, a cut-off threshold should be computed. Let  $a$  and  $b$  be the minimum and maximum of the average lengths. Then, any number  $x$  satisfying  $a \leq x \leq b$  can be used as a cut-off threshold during the classification stage. Finally, if  $avgLen(o) < x$  then, OCLEP classifies  $o$  as an outlier (masquerader); otherwise,  $o$  is classified into the normal class, which is the only class used in the training stage. Notice that a cut-off close to  $a$  will lead to a low false positive rate while a cut-off close to  $b$  will lead to a high false positive rate. As a consequence, the authors recommend selecting a cut-off point close to  $a$ . The main drawback of OCLEP is

the number of input values for classifying an object. The authors recommended using specific values in their experiment, but the classification results are not consistent when the input values are changed. Also, OCLEP does not improve significantly the classification results obtained by ocSVM [165]. Another handicap of OCLEP is its computational time at the classification stage. Since for classifying each query object, OCLEP needs to extract EPs that differentiate that query object from objects previously classified into the normal class [51].

Later, in 2006, the authors of [55] proposed to use the bagging approach using EP-based classifiers. For doing that, the authors proposed a scoring function favoring the support instead the growth rate for a subset of EPs. This function is defined as:

$$\sigma(o, PS, C_i, C) = \sum_{\substack{p \in PS \\ p \text{ covers } o}} \frac{\text{support}(p, C_i, \bar{C}_i)}{\text{support}(p, C_i, \bar{C}_i) + \text{support}(p, \bar{C}_i, C_i)} \quad (9)$$

where  $o$  is a query object,  $PS$  is a set of EPs,  $C_i$  is the positive class and  $\bar{C}_i$  the negative class (or the complement of  $C_i$ ). In short, this scoring function computes, for a set of EPs, the impact of classifying a query object  $o$  by using the EP's support in class  $C_i$  divided by the support across all classes in  $C$ . The authors showed that their proposal, using (9) jointly to ensembles of EP-based classifiers, improves CAEP and other popular state-of-the-art classifiers not based on EPs.

In 2011, a Fuzzy Emerging Pattern-based Classifier (FEPC) was introduced [68]. The authors modified the quality measure *support* for making it suitable for fuzzy problems. In this way, an object supports every fuzzy pattern to some degree according to its membership to the fuzzy items of the pattern. Then, the individual fuzzy support of an object  $o$  for a given fuzzy pattern  $F$  is defined as the minimum membership ( $\mu$ ) of all its feature values in their respective fuzzy sets:

$$fsup(o, F) = \min_{f \in F} \{\mu_f(o)\}. \quad (10)$$

Then, the support of a fuzzy pattern  $F$  in a class  $C_i$  is the sum of the individual fuzzy support for all objects in  $C_i$ :

$$FSup(F, C_i) = \sum_{o \in C_i} fsup(o, F). \quad (11)$$

Based on (10) and (11), the authors defined a Fuzzy Emerging Pattern (FEP) as a fuzzy pattern  $F$  with  $Trust(F) > 0.5$ :

$$Trust(F) = \frac{\max_{C_i} FSup(F, C_i)}{\sum_{C_i} FSup(F, C_i)}. \quad (12)$$

Finally, for classification, the authors proposed a graph structure where all patterns are organized into a graph. Each extracted FEP is represented as a node and there exists a arch from node  $P_1$  to node  $P_2$  if the pattern contained in  $P_1$  is more specific than the pattern contained in  $P_2$ . Similar to the classification strategy proposed in CAEP, FEPC assigns to a query object  $o$  the class with the highest total vote. For computing the votes per class of the query object  $o$ , FEPC evaluates the patterns with no ancestors. If an evaluated pattern matches  $o$  (with fuzzy support above a certain threshold) the vote to its class is increased with its *Trust*, while all its descendants are discarded. Otherwise, all immediate descendants are evaluated in the same way. The process ends when every node has been evaluated or discarded.

Usually, algorithms for mining patterns in class imbalance problems extract several patterns having high support for the majority class and only a few emerging patterns, having low support, for the minority class [116–118, 124, 127]. This makes that some pattern-based classifiers, which are based only on the sum of supports, such as LCMine, CAEP, Apriori, and iCAEP, become biased toward the majority class [127]. For solving this problem, at the classification stage, in [127] proposed PBC4cip. PBC4cip weights the sum of supports in each class taking into account the imbalance of the classes in the training dataset. The main idea is that, at the classification stage, those patterns having low support for the minority class do not become overwhelmed by those patterns having high support for the majority class. In this way, PBC4cip weights the sum of support for those patterns covering an object to be classified, by value  $w_c$  that takes into account the patterns in the class, their support, and the class imbalance, according to the following expression:

$$w_{C_i} = \left(1 - \frac{|C_i|}{|D|}\right) / \sum_{p \in PC_i} support(p, C_i, \bar{C}_i) \quad (13)$$

where  $|C_i|$  represents the number of objects belonging to the class  $C_i$ ,  $|D|$  is the number of objects in the training dataset,  $P_{C_i}$  is the set of EPs mined for the class  $C_i$ , and  $support(p, C_i, \bar{C}_i)$  is the support of the pattern  $p$  into the class  $C_i$ .

Notice, that the term  $(1 - |C_i|/|D|)$ , proposed by PBC4cip, allows rewarding the sum of supports computed for the minority class, which usually is low, since the smaller the value of  $|C_i|$ , the higher the value of this term. On the contrary, this term punishes the sum of supports computed for the majority class, which usually is high, since the higher the value of  $|C_i|$ , the lower the value of this term. Additionally, the term  $\sum_{p \in P_{C_i}} support(p, C_i, \bar{C}_i)$  is used for normalizing the sum of supports in each class regarding the support of all patterns of the same class. In this way, the weight, defined by PBC4cip, aims to overcome the bias of the classifier to the majority class, by assigning a higher weight for the minority class.

In 2018, OCLEP+ was proposed [51] as an extension of the OCLEP. The main difference is how OCLEP+ uses the lengths of EPs and how the length is computed. OCLEP+ is based on the minimal length of JEPs instead of using the average length of JEPs. The authors claim that the EPs differentiating objects of different classes are often short, whereas the EPs differentiating objects of a common class are often long. As a consequence, the authors proposed to use minimal length of JEPs during the classification stage. In their experimental results using a few databases, the authors showed that OCLEP+ improves OCLEP and ocSVM using different kernels. The main drawback of OCLEP+ is the tuning-up of the input values because they have an impact on the classification results. The authors showed how some input values could improve the classification results from 59.65 to 80.61 of accuracy.

Recently, in 2019, a modification of CAEP for class imbalance problems by using Cost-Sensitive Patterns (CSPs) was proposed by [130]. CACSP computes the sum of cost per class of all patterns covering a query object, and then, that class where the misclassification cost reaches the minimum cost will be the class of the query object. In case of ties, the classifier assigns the minority class to the query object. The authors decided to assign the minority class to the query object in the case of ties because those objects belonging to the minority class have higher misclassification cost. The scoring function used in CACSP is defined as:

$$\sigma(o, PS, C_i, C) = \sum_{\substack{p \in PS, \\ p \text{ covers } o}} support(p, C_i, \bar{C}_i) * m(C_i) \quad (14)$$

where  $support(p, C_i, \bar{C}_i)$  is the support in the class  $C_i$  of the CP  $p$  covering the query object  $o$ ,  $PS$  is the set of CPs for all the classes, and  $m(C_i)$  is the misclassification cost associated to the class  $C_i$ .

In 2019, the authors showed that their proposal obtains lower misclassification cost [45] than two other well-known classifiers based on CPs and ten popular classifiers not based on CPs using 95 imbalanced datasets and five cost matrices.

From all papers reviewed in this section, we can conclude that there are many proposals for the supervised classification based on CPs (see Table 2). However, several of these proposals used a scoring function for combining the information provided by a collection of patterns; as a consequence, some problems, like the class imbalance problem, can negatively affect the classification results of these proposals based on scoring functions. On the other hand, classification based on fuzzy patterns seems to be promising for obtaining good classification results, but it has been little studied.

The OCLEP classifier has shown good classification results in a few databases. The main drawback is that OCLEP relies on the length quality measure and, in some context, the patterns mined from exhaustive-search-based algorithms have many more items than those patterns coming from decision tree-based algorithms. However, OCLEP has only been tested using patterns mined from exhaustive-search-based algorithms; then its classification results could change when using patterns mined from other mining algorithms, which do not use an initial discretization.

Furthermore, it is important to highlight that the classification stage needs a collection of high-quality patterns for obtaining good classification results, but usually, it does not happen because the before stages (mining and filtering) cannot provide this collection of high-quality patterns.

From all reviewed paper in Section 2, we have collected their titles and abstracts to generate a word cloud to capture the most studied topics. To build the word cloud presented in Fig. 3, we first removed the stop words most commonly used in the English language such as “the”, “of”, and “and”; then we have lemmatized each word. The size of each word into





- classification while providing an easy-to-understand classification model. A limitation of these patterns is that the quality and amount of patterns extracted highly depends on the way that we measure the “significant” differences of support.
- (b) *Chi-Emerging Pattern*: A contrast pattern where the “significant” difference of support is evaluated by the  $X^2$  test [168] which must be greater than a threshold. There are studies [65, 66, 76, 77, 121, 123] showing that there are several quality measures better than  $X^2$  for evaluating the patterns for supervised classification.
- (c) *Emerging Pattern*: This is a type of contrast pattern where the “significant” difference of support is evaluated by the quality measure Growth Rate which must be greater than a threshold. There are studies [65, 66, 76, 77, 121, 123] showing that there are several quality measures better than Growth Rate for evaluating the patterns for supervised classification.
- (d) *Strong Jumping Emerging Pattern*: This is a type of emerging pattern where the number of objects supported by the pattern in the classes of low support (considered noisy objects) should be lower than a predefined threshold. Computing this threshold is challenging; besides, there are papers [69, 70] showing that the classification is better when using diverse patterns; i.e., patterns tolerating both low and high amount of noisy objects.
- (e) *Jumping Emerging Pattern*: This is a particular case of strong jumping emerging pattern where the pattern supports objects of only one class. When the classes are hard-to-separate, the mining algorithms might extract jumping emerging patterns with many items which are hard-to-understand by users. These long patterns might be too specific (supports a few objects); hence, overfitting the classifiers.
- (f) *Jumping Emerging Pattern with Negation*: This is a particular case of jumping emerging pattern where the patterns contain items having negated values; i.e., values that do not appear in any object. The problem with these patterns is that it is hard to determine which items to negate and how many patterns must be negated to achieve high accuracy of classification.
- (g) *Essential Jumping Emerging Pattern*: A jumping emerging pattern is considered essential when its value of Growth Rate is higher than some threshold. Finding the correct threshold for each dataset could be challenging.
- (h) *Balanced Emerging pattern*: This is an emerging pattern designed to deal with class imbalance problems. This type of emerging pattern must be compliant with some thresholds associated to the imbalanced of the classes.
- (i) *Cost-sensitive Pattern*: This is a contrast pattern whose support per class is weighted according to some cost matrix. The accuracy of classifying with these patterns highly depends on the quality of the end user’s cost matrix.
- (j) *Fuzzy Emerging Pattern*: A fuzzy emerging pattern supports every object with some degree of membership to each class of the problem. An advantage of such a pattern is that it is expressed in a language closer to the human’s natural language. The quality of a fuzzy pattern highly depends on the quality of the features’ fuzzifying algorithms.
- (k) *Multivariate Contrast Pattern*: This is a contrast pattern which items can be either univariate items (stated in Section 1) or multivariate items. These patterns allow for better performance of classification, but they are more difficult to understand.

### T3 Filtering method:

- (a) **Based on set theory**: These algorithms select a subset of patterns based on operations among the sets of objects that the patterns cover. These algorithms might be slow if the amount of objects is large. An advantage of these algorithms is that they return a subset of patterns having low redundancy among them.

- (b) **Based on quality measures:** These algorithms rank the patterns according to some quality measures and then apply a greedy strategy to select a subset of patterns. These algorithms are faster than those based on set theory, but they might return a set of patterns having high redundancy among them.

Table 3 classifies the reviewed papers according to the proposed taxonomies. While in [46] extensively reviews the algorithms with mining strategies based on exhaustive search, we reviewed some of those

algorithms which were milestones. Additionally, we reviewed the algorithms that mine contrast patterns from decision trees.

From our study, we highlight the following:

- The algorithms that mine patterns from decision trees are more accurate than those based on exhaustive search.
- The algorithms based on fuzzy contrast patterns or multivariate contrast patterns achieve higher accuracies.
- There is a need for creating more algorithms that mine contrast patterns from Big Data.

**Table 2** Summary of the algorithms based on contrast patterns classified according to five taxonomies

Ref.	Mining strategy	Type of pattern	Filtering method
[17]	Exhaustive search	Jumping Emerging Pattern	–
[184]	Exhaustive search	Jumping Emerging Pattern	–
[49]	Exhaustive search	Emerging Pattern	–
[107]	Exhaustive search	Minimal Jumping Emerging Pattern	–
[11]	Exhaustive search	Emerging Pattern	–
[56]	Exhaustive search	Essential Jumping Emerging Pattern	–
[57]	Exhaustive search	Essential Jumping Emerging Pattern	–
[58]	Exhaustive search	Chi-Emerging Pattern	–
[6]	Exhaustive search	Balanced Emerging Pattern	–
[106]	Exhaustive search	Emerging Pattern	Based on set theory
[7]	Exhaustive search	Emerging Pattern	Based on quality measure
[59]	Exhaustive search	Strong Jumping Emerging Patterns	–
[5]	Exhaustive search	Emerging Patterns	–
[175]	Exhaustive search	Jumping Emerging Pattern with Negation	–
[176]	Exhaustive search	Jumping Emerging Pattern with Negation	Based on quality measure
[97]	Exhaustive search	Jumping Emerging Pattern	–
[115]	Exhaustive search	Strong Jumping Emerging Pattern	–
[4]	Exhaustive search	Strong Jumping Emerging Pattern	–
[35]	Exhaustive search	Balanced Emerging Pattern	–
[71]	Decision trees	Contrast Patterns	Based on set theory
[67]	Decision trees	Contrast Patterns	Based on set theory
[186]	Decision trees	Contrast Patterns	–
[68]	Decision trees	Fuzzy Emerging Pattern	–
[70]	Decision trees	Contrast Pattern	–
[124]	Decision trees	Contrast Pattern	–
[127]	Decision trees	Contrast Pattern	Based on set theory
[75]	Exhaustive search	Fuzzy Emerging Pattern	–
[74]	Exhaustive search	Fuzzy Emerging Pattern	–
[72]	Exhaustive search	Fuzzy Emerging Pattern	Based on quality measure
[130]	Decision trees	Cost-sensitive Pattern	Based on set theory
[27]	Decision trees	Multivariate Contrast Pattern	Based on set theory

## 2.5 Scientometric Study

Over the years, many journals have been indexed in databases, and the indexation is determined by well-defined and quantifiable criteria, such as acceptance and rejection rates, promptness of publication, and high frequency of citation by other journals (impact); among others [153].

Scientometrics is about measuring the progress of science. It includes quantitative studies of scientific activities, including publication and so overlaps bibliometrics to some extent [87]. Hence, a scientometric study should be based on reliable databases, like SCOPUS<sup>2</sup>, which index the published papers on different journals.

As far as we know, there is not any scientometric study about the classification based on CPs. From this study, we can determine the impact of the papers derived from this area for the international research community. Also, we can identify the top-contributing institutions and authors, the publications generating more citations and if they were published in congress or journal, and the number of generated citations by paper; among others. From this study, the international research community can know, in a summarized way, relevant information to connect authors, networking, and research entities, which are specialized in cp-based classification.

In our study, data acquisition was designated to extract information from SCOPUS, a database of peer-reviewed literature. For this extraction, we use the Application Programming Interfaces (APIs) provided by Elsevier Developers<sup>3</sup>, which allow obtaining up to 6,000 results<sup>4</sup> for each query. We have created a query in SCOPUS for all those papers talking about “Contrast Pattern”, “Frequent Item”, “Emerging Pattern”, or “Association Rule”, and these were published in the computer science area. For each paper collected using the query mentioned above, we have extracted several features such as title, the name of authors, affiliations, number of citations, source of publication, and keyword indexation; among others.

Figure 4 shows a line graph regarding the number of published papers in the cp-based classification

topic since 1971 to January 2020. From Fig. 4, we can see how the cp-based classification topic has been gaining interest in the last years for the international research community. Also, we can notice that from 1997 this topic began an exponential growth regarding the number of published papers. For the last 10 years, on average, more than 800 papers are published in both conferences and journals. Furthermore, from this figure, we can observe that the number the published paper during January 2020 is equal to all papers published in 1997.

Figure 5 shows a donut graph regarding the number of papers published into the top-10 conferences and journals for the cp-based classification topic. In the same vein, Fig. 6 shows a donut graph regarding the number of published papers, in the cp-based classification topic, for each type of document indexed by SCOPUS.

From Figs. 5 and 6, we can observe that most of the papers appear in Lecture Note in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), which publishes conference proceedings. Also, from Fig. 5, we can notice that other specialized journals such as Expert System With Applications, Information Science, and IEEE Transactions On Knowledge And Data Engineering are into the top-10 journal list for publishing papers related to contrast pattern-based classification.

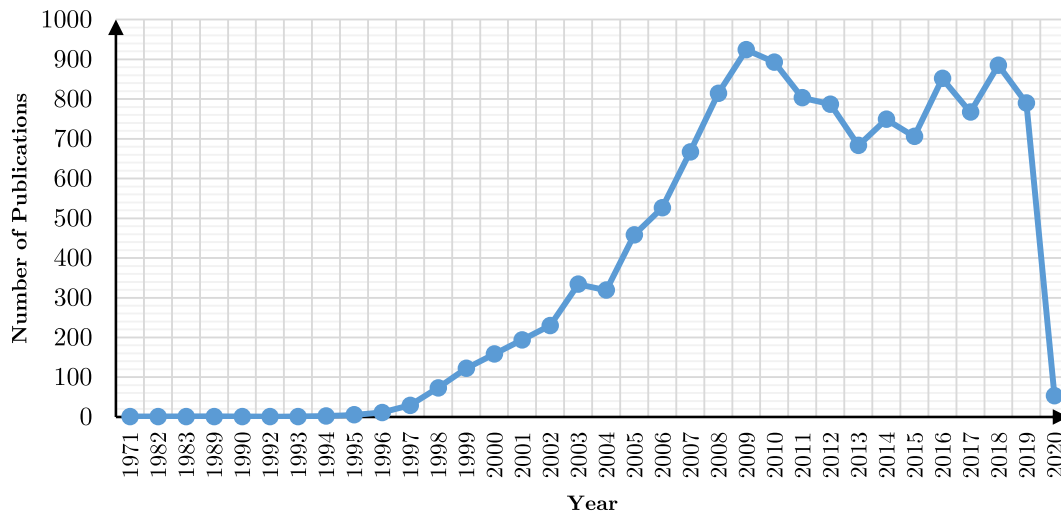
Figure 7 shows a donut graph representing the percentage of published papers in the cp-based classification topic regarding the top-10 subjects indexed by SCOPUS. From Fig. 7, we can see that the most prominent subjects for publishing papers of CPs are: Computer Science, Mathematics, and Engineering, which mainly contains theoretical papers. Also, we can notice that there are other suitable subjects to publish those works applying contrast pattern-based classification in real-world problems, such as Decision Sciences, Medicine, Physics and Astronomy and Materials Science. These results support our reviewed papers in Section 2.1, Sections 2.2 and 2.3.

An important point is to know the most prominent countries, as well as their affiliated institutions, publishing works in the cp-based classification topic. Hence, in Fig. 8, we have created a thermal-world map for showing the number of published papers per

<sup>2</sup><https://www.scopus.com>

<sup>3</sup>[http://dev.elsevier.com/sc\\_apis.html](http://dev.elsevier.com/sc_apis.html)

<sup>4</sup>[http://dev.elsevier.com/tecdoc\\_developer\\_faq.html](http://dev.elsevier.com/tecdoc_developer_faq.html)



**Fig. 4** Number of published papers in the contrast pattern-based classification topic since January 1971 to January 2020

country. From this figure, we can notice that China, India, United States, Taiwan, Japan, Australia, and France (in this order) are the countries generating more papers about the cp-based classification topic. Also, from Fig. 9, we can see that the top-10 institutions generating papers in this topic come from China; although, it is essential to highlight that Universidad de Granada (Spain) is making significant advances in the topic, specifically in the association rule area.

Finally, from our scientometric study, we can conclude that contrast pattern-based classification is gaining interest in the international research community, where more than 700 papers, on average, are published each year from 2007 to 2019. The main subjects where these papers were published are computer science, mathematics, and engineering. On the other hand, the primary source for publishing papers of pattern-based classification is conference proceeding (62% of all published papers). Additional, the most prominent countries working in this topic are China, India, United States, Taiwan, Japan, Australia, and France; in this order.

An essential component of any research area is to prove the proposed algorithms in practical domains, where the applied context requires to adjust these proposed algorithms for obtaining both good performance and high accuracy. For that reason, in the next section, we review several papers applying the cp-based classification on real-world problems.

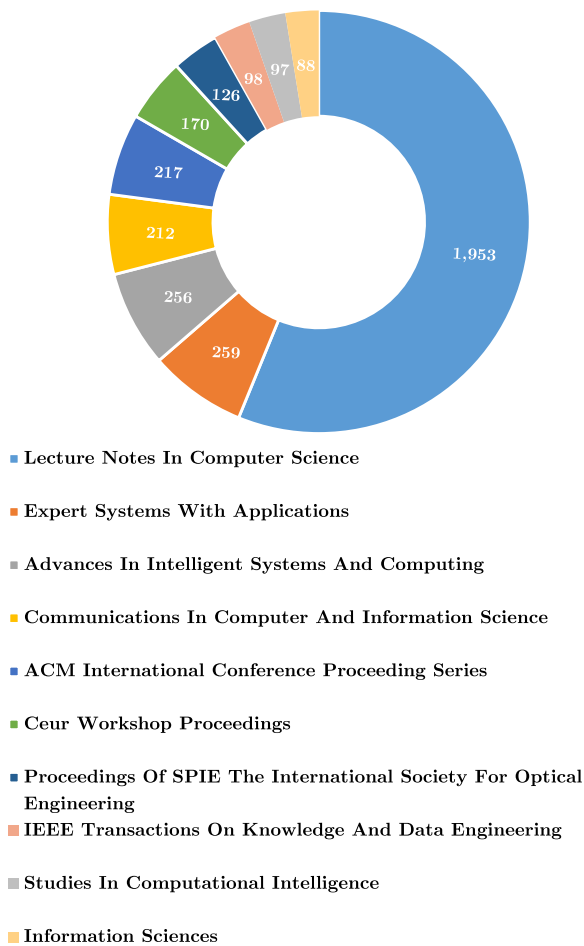
### 3 Applying Contrast Pattern-Based Classification to Real-World Problems

In this section, we review 36 of the most outstanding papers applying the cp-based classification to real-world problems. Papers were ordered chronologically as follow:

In 2001, a system for detecting changes in consumer habits was proposed [170]. The consuming habits are represented with association rules and EPs. The authors proposed a measure to quantify the degree of change of the patterns and to filter the patterns. The authors created two proprietary datasets with discretized features. Apriori mines the patterns which are not compared with the results of any other algorithm.

After, in 2004, the authors of [50] use EPs to analyze gene expression profiles for the diagnosis of a disease state. The authors use the classifier Prediction by Collective Likelihood (PCL) on a single dataset and compare only with C4.5, Naive Bayes (NB), and (SVM). The authors discretize the feature using an algorithm that they designed specifically for this dataset. A border-based algorithm extracts the patterns which are not compared against the results of different mining algorithms.

In 2009, the authors of [194] use EPs to detect metamorphic malware. To classify an object, the classifier aggregates a score based on Support and Growth Rate of the EPs (with Support higher than 0.04) covering the object. The authors reported results of



**Fig. 5** Number of papers published in the top-10 conferences and journals based on the contrast pattern-based classification topic

a single dataset without comparing it to any other classifier.

Next, the authors of [163] mine EPs from proteins to map the original proteins' structure [140] into a new representation where the EPs are the new features. Then, traditional classifiers predict the protein class. The results show that SVM outperforms C4.5, NB, and Bayesian Network (BN) when using the features based on EPs. The authors do not describe the characteristics of the used dataset neither the evaluation protocol that they use.

After, a classifier based on a tree of EPs for myocardial ischemia diagnosis is proposed by [152]. The authors mine EPs from a tree that they build from discretized features. The proposal outperforms the classifiers C4.5, Classification based on Multiple Class-Association Rules (CMAR) [109], and

Classification based on Predictive Association Rules (CPAR) [196] using a proprietary dataset; but no statistical test corroborates the results.

Later, the authors of [148] mine EPs and use CAEP for predicting non-safe power lines. The authors create a dataset which is not publicly available. The results showed that CAEP outperforms C4.5, SVM, NB, BN, CMAR, and CPAR; but there is not any statistical test supporting the results.

In 2010, EPs are used by [152] to describe motifs of support for Barack Obama in the 2008 presidential election. This research uses EPs not for classification, but for explaining a phenomenon. The authors filter the patterns with a Growth Rate threshold of 15. This work is an excellent example of how EPs are good starting points for data visualization. The authors create a dataset, with discrete features, which is not public. There is not any comparison with any other algorithm for mining CPs.

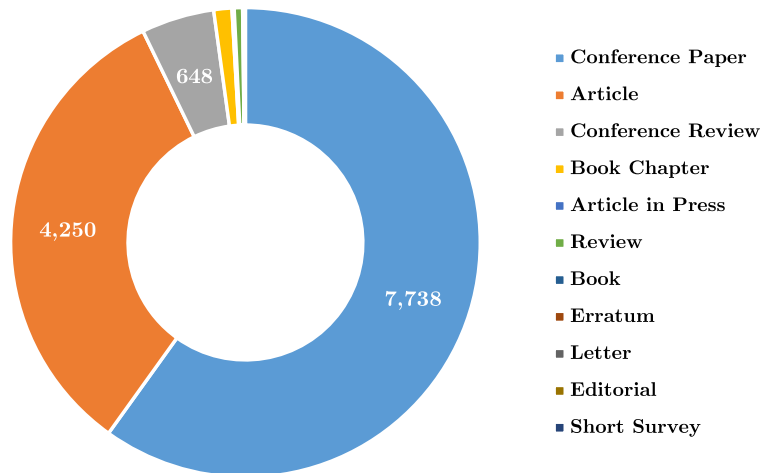
After, the authors of [204] proposed NRMINDER, an exhaustive-search-based algorithm (see Fig. 2.1.1) for discovering diagnostic gene CPs from microarray data. The mining algorithm internally filters the patterns according to their support and confidence. The authors classify new objects according to the CP that supports the object and maximizes a proposed score measure. The authors compare their proposal only with the algorithm FEAT [64] using three public dataset, but there is no statistical test supporting the conclusions.

In 2012, the authors of [108] use EPs and a classifier named PCL for profiling subtypes of childhood acute lymphoblastic leukemia patients. The authors test the proposal in a single public dataset showing that it outperforms SVM, k-NN, NB, and C4.5. No statistical test corroborates the classifiers differences in this research.

Next, in [132], the authors compute JEPs for finding genes with high discriminating power in microarray datasets and then they build a classifier based on the votes of the found genes. There is not any comparison with any other classifier.

After, in [98], the authors create three subtypes of JEPs (EPs with occurrence counts, spatial EPs, and jumping emerging substrings) for classifying multimedia data. Their classification strategy bases on the sum of the supports of the minimal patterns by class. The authors test their proposal in a public dataset showing that their proposal outperforms C4.5 and

**Fig. 6** Number of published papers in the contrast pattern-based classification topic regarding the type of documents



SVM. They do not use any statistical test neither compare with the most accurate image classifiers: convolutional neural networks.

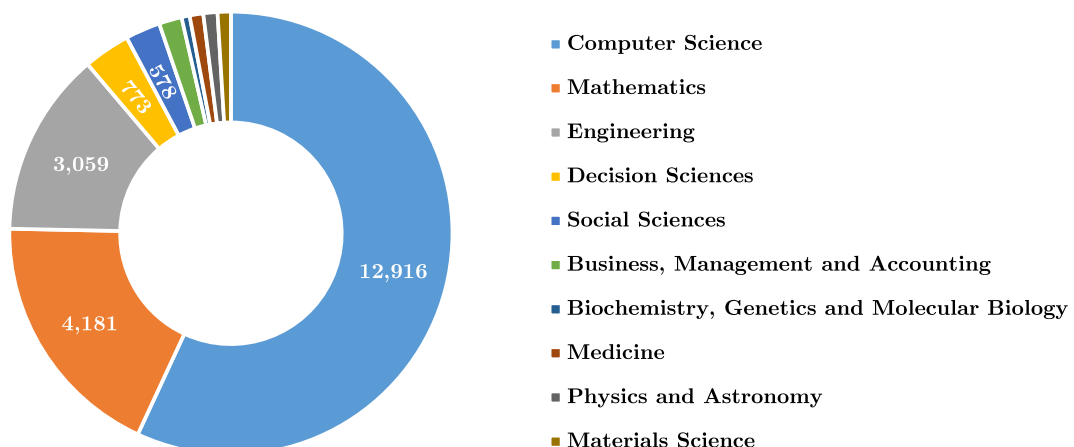
Later, the authors of [16] define Frequent Emerging Molecular Pattern (FEMP) for classifying molecules as toxic or non-toxic. A FEMP is a pattern from a two-classes chemical dataset with a frequency and growth rate higher than some predefined thresholds. The authors experiment with a public dataset, but there is not any comparison with other algorithms, and there is not any statistical test on the results. Similar to EPs, FEMP are mined from discrete features.

In 2013, the authors of [159] mine fuzzy EPs for modeling slope units affected or not affected by landslides. The authors use FISpro tool [80] for classification based on the fuzzy EPs. An interesting characteristic of this algorithm is that the authors use

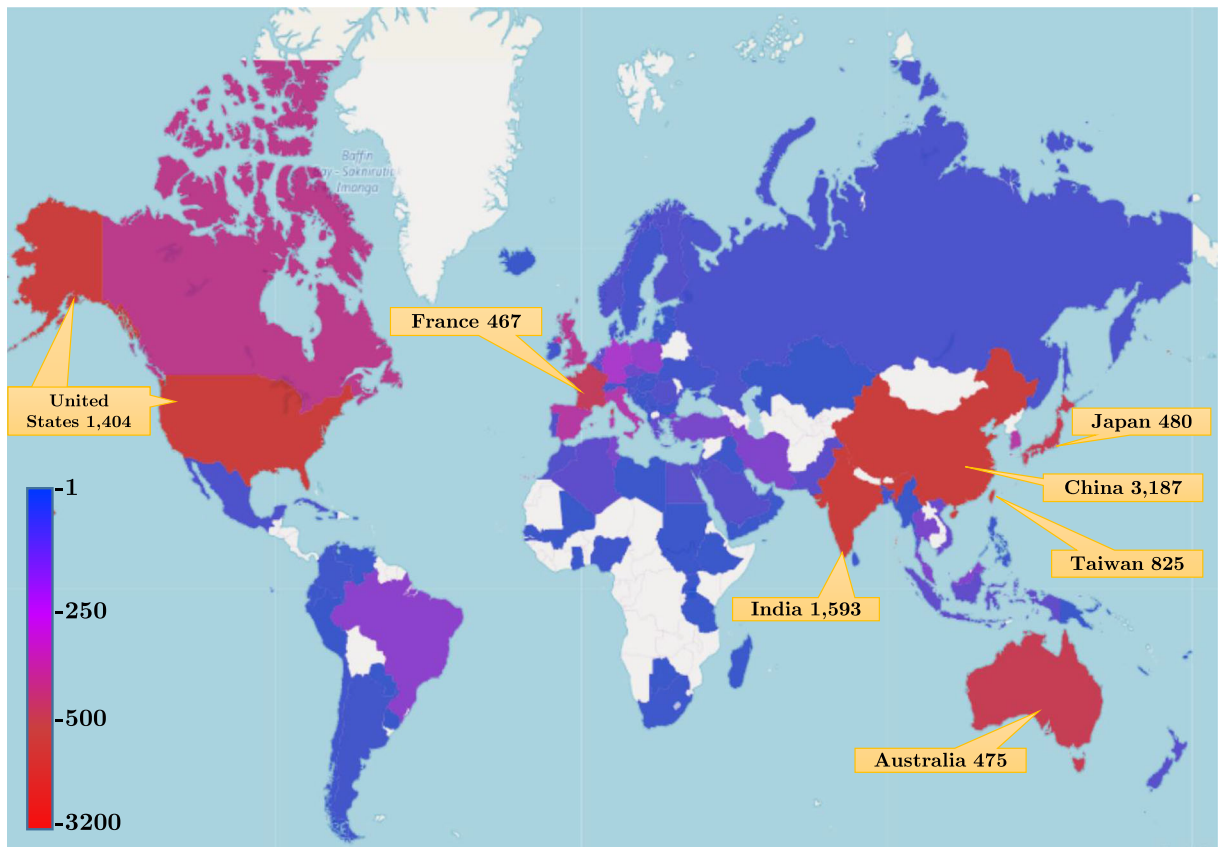
several clustering algorithms and validity indexes to determine the best partition by feature. The authors work with a single dataset without comparing with any classifier or performing any statistical test.

Subsequently, in [167] their authors designed an algorithm for mining EPs which form hierarchical clusters of compounds for toxicity predictions. The authors test their proposal in two public datasets, but they do not perform the traditional protocols for comparing supervised classifiers. The paper has not any comparison with any other EPs mining algorithm or supervised classifier. The authors do not perform any statistical test of the results.

In the same year, the authors of [1] proposed a rule-based classification approach for phishing detection. The algorithm works with multi-label rules; this is similar to the CPs based classification because



**Fig. 7** Number of published papers in the contrast pattern-based classification topic regarding the top-10 subjects

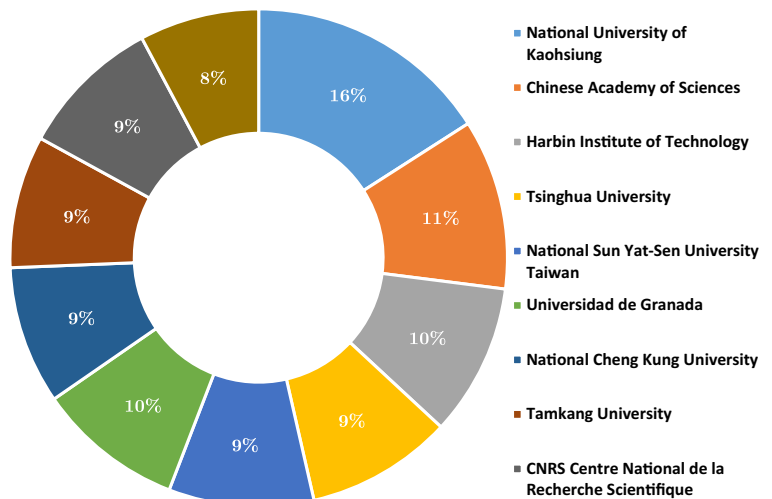


**Fig. 8** Thermal-world map for showing the activity by country in the contrast pattern-based classification topic since January 1971 to December 2019

every CPs may vote for multiple classes. The authors create new features from a public dataset, but the new dataset is not public. The experiments, after discretizing the features, show that the proposal

outperforms C4.5, CBA [110], PART [63], RIPPER [38], MCAR [179], MMAC [180]; but there is not any statistical test supporting the results.

**Fig. 9** Percentage of published papers in the contrast pattern-based classification topic regarding the top-10 institutions. Collected since January 1971 to December 2019



**Table 3** A taxonomy of the existing approaches for pattern-based classification, which is divided in the main stages of this topic: mining, filtering, and classification. For each approach, advantages and drawbacks are shown

Stage	Approach	Refs.	Advantages	Drawbacks
Mining	Exhaustive search	[2, 4–7, 11, 15, 17, 23–26, 33, 35, 49, 51, 56–59, 82, 83, 91, 97, 106, 107, 114, 115, 149, 169, 175, 176, 184, 188, 198, 199]	<ul style="list-style-type: none"> <li>– In small databases, it can find all possible patterns.</li> </ul>	<ul style="list-style-type: none"> <li>– High-computational cost.</li> </ul>
	Decision trees	[67, 68, 70–72, 74, 75, 124, 127, 130, 186]	<ul style="list-style-type: none"> <li>– Low-computational cost.</li> <li>– It does not need an initial discretization for numeric features.</li> </ul>	<ul style="list-style-type: none"> <li>– It can not handle missing values.</li> <li>– Initial discretization of all numeric features.</li> <li>– Only can extract patterns having items of the form <math>[f_i = v_j]</math>.</li> <li>– The limited possibility of transformed data interpretation by the initial discretization.</li> <li>– Unable to find patterns in some databases. For example, the wpbc database; taken from UCI Machine Learning Repository [53].</li> <li>– It is unable to find all possible patterns from a dataset.</li> <li>– The extracted patterns could not be useful for explaining the results in terms of the original problem's classes when they were obtained by using bagging or boosting method.</li> </ul>
			<ul style="list-style-type: none"> <li>– It can handle missing values.</li> <li>– It obtains a small collection of high-quality patterns</li> <li>– It obtains patterns with more discriminative power by using other relational operators, like <math>\{\in, \notin, =, \neq, \leq, &gt;\}</math>.</li> <li>– A good interpretation because the features are not transformed by using any initial discretization.</li> </ul>	
Filtering	Set theory	[56, 57, 69, 187]	<ul style="list-style-type: none"> <li>– It evaluates the patterns based on their items instead of their support.</li> </ul>	<ul style="list-style-type: none"> <li>– High-computational cost: <math>O(n^2)</math>.</li> </ul>



**Table 3** (continued)

Stage	Approach	Refs.	Advantages	Drawbacks
	Quality measure	[66, 66, 76, 77, 111, 123, 136, 137, 147]	<ul style="list-style-type: none"> <li>– Low-computational cost: <math>O(n)</math>.</li> </ul>	<ul style="list-style-type: none"> <li>– It can obtain minimal patterns, which are interesting for describing a model because, in general, they contain a low number of items and higher support.</li> <li>– Subjective measures are based on a specific criterion issued by an expert in the application domain, which is not available in any repository.</li> </ul>
Classification	Unweighted score	[17, 33, 51, 52, 55, 57, 67, 71, 184, 186, 201]	<ul style="list-style-type: none"> <li>– It obtains a small collection of high-quality patterns based on a training dataset.</li> <li>– It is easy to compute and understand.</li> </ul>	<ul style="list-style-type: none"> <li>– The collection of patterns could have low quality for objects in the testing dataset.</li> <li>– It is biased by the quality measure used for computing the score.</li> </ul>
	Weighted score	[68, 127, 130]	<ul style="list-style-type: none"> <li>– It has reported good classification results on problems with balanced classes.</li> <li>– It is suitable for handling class imbalance problems.</li> </ul>	<ul style="list-style-type: none"> <li>– It has obtained bad classification results on class imbalance problems.</li> <li>– The weight is based on the distribution of objects by classes of the training dataset, which could be different from the testing dataset.</li> </ul>

In 2015, the authors of [105] proposed to mine EPs for profiling users of hotels. The goal of the method is to generate insights to help tourism managers address travelers' concerns. The authors create a dataset that is not public, so their experiments cannot be replicated. The research does not include any comparisons with other EPs mining algorithms, and it does not perform any statistical test.

After, during the same year (2015), the authors of [40] use EPs and JEPs for identifying clusters of compounds in datasets about Ames mutagens. The authors modify a public dataset, but the new dataset is kept proprietary. Their research does not compare with previous works neither includes any statistical test of the results.

In 2016, the authors of [141] use CPs to distinguish groups of pieces within a music corpus. The authors mine different types of patterns on different datasets without methodological support for this decision. In the same vein, they compute the p-value of the patterns for some datasets, and it is not clear the justification of this decision. There is not any comparison among the mining strategies, neither any statistical test comparing different algorithms.

After, in [125], the authors proposed to use CPs to model failures in Temporary Immersion Bioreactors (TIBs). The authors use the classifier iCAEP based on CPs to predict the failures in the TIBs. The authors used eight databases and four classifiers based on patterns in their experimental setup. In [125] showed that their proposal, using bagging miner jointly with iCAEP, obtains significantly better classification results than other tested contrast pattern-based classifiers.

Next, the authors of [205] mine CPs from structured and sequential data to analyze taxpayer behavior. The authors mine the CPs from a tree that they create using discrete data that takes into account the temporal dependencies among objects. The authors experiment with three proprietary datasets without comparing with any other classifier. No statistical test supports the results.

Afterward, in [29], the authors proposed to mine EPs from gamers of Defense Of The Ancients 2. The idea is to study the behavior of expert gamers, so the new ones may reduce the learning curve by following the patterns of expert gamers. The authors create a proprietary dataset, do not compare their results with

other algorithms, and do not perform any statistical test that corroborates the experimental results.

Subsequently, the authors of [160] use EPs to find combinations of drugs and medical events that are associated with the development of myocardial infarction. The authors create a single proprietary dataset with custom features where they test only their proposal. No statistical test is used, and no comparison with previous works is performed.

In 2017, the authors of [31] use CPs to model changes in network traffic. They introduce patterns's quality measures to reflect the interpretability of patterns for security managers. The authors filter the patterns based on growth rate and class coverage differentiation. The authors evaluate their patterns only based on their respective support and do not perform any classification experiment. There is no evaluation of other mining algorithms or type of patterns, and there is no statistical test evaluating the mined patterns.

In the same year, the authors of [154] mine EPs to identify sets of metabolite biomarkers that are useful to discriminate diseased from healthy subjects. The authors use a public dataset which features they discretize to mine EPs. The research includes a comparison with SVM, Random Forest, and a Neural Network; but no statistical test shows a significant difference among the classifiers.

After, in 2018, the authors in [173] mine closed frequent patterns to build features that are the input of Random Forest for complex activity recognition in smart homes. The authors test their proposal in a public dataset where they compare with Hidden Markov Models (HMM) [14], BN, NB, SVM, and C4.5; but no statistical test validates the results.

Next, in [44], the authors proposed a generic framework for mining frequent itemsets from auto-generated transactional databases of event logs. Similar to the rest of the algorithms than mine frequent itemsets, this research requires discretizing the features. The proposal compares with Episode-mining [102],  $\alpha$  algorithm [183], ImprovedInductive-miner [103], Declare-miner [131], and Episode-miner [102] using two public datasets; but perform no statistical test over the results.

Afterward, the authors of [128] mine CPs from web navigation logs to train PBC4cip to detect bots. The authors used LCMine, Bagging, and Random

Forest miners, three state-of-the-art datasets, and two datasets collected by themselves. The authors showed that their proposal for using the Random Forest miner jointly with the classifiers PBC4cip [127] obtains significantly better classification results than ten popular state-of-the-art classifiers. The authors carried out a validation of the extracted pattern together with the experts in the application area.

Next, the authors of [150] mine EPs from microblog streams. Then, the authors cluster the EPs to detect emerging topics. The authors create a proprietary dataset where they compare with online-LDA [100], SFSD [151], and HUPC [89]. No statistical test validates the results.

Subsequently, in [88], the authors proposed to mine frequent itemsets to discover meaningful patterns in alarm floods. The authors propose visualization techniques based on exiting plots to show alarm floods and alarm patterns. The authors create a proprietary dataset where no comparison with previous works appear neither any statistical test of significance of the results.

Later, in [189] use EPs to describe product sales trends in dynamic markets. The authors create two proprietary datasets where they test their proposal without comparing with any other algorithm. No statistical test validates the results.

Recently, in [122], the authors proposed to extract CPs from sentiment features from tweets of political figures related to the presidential election in Mexico, 2018. The author used the Random Forest miner for extracting contrast patterns and for generating diversity, they evaluated each binary candidate split at each decision tree level using four different measures (Hellinger distance, Bhattacharyya, information gain, and Gini index). In [122], all extracted patterns were used for describing the tweets together to a visual model instead of classification. The authors did not use any statistical method for validating their results because they used the expert opinions.

During the same year, the authors of [126] extends their earlier work [125] showing that the classifier PBC4cip outperforms other 11 classifiers. The authors used the eight datasets used in [125] and new datasets. The main difference is that the proposal of [126] can extract patterns for identifying six types of failures instead of two failures detected by [125]. The authors

showed that their proposal obtains significantly better classification results than other 11 state-of-the-art classifiers.

After, in [28] proposed a protocol for frequent items discovery in unstructured P2P networks. The authors created software, which they made public, that simulates the necessary data to test their proposal. The authors do not compare with other works neither perform a statistical test to support their results.

Next, in [193] extracted rare-unusual association rules from a stroke medical dataset to provide meaningful knowledge to the user domain. The authors extract rules from a single dataset, which is proprietary. There is not any statistical test evaluating the significance of the rules against any other mining algorithm.

Afterward, in [54], their authors proposed to extract association rules to describe the most common conditions of accidents in France. The authors use Apriori to mine rules from a dataset that they created and made public. The proposal is not compared with any other algorithm, and there is no statistical test corroborating the results.

Later, the authors of [129] extract CPs from tweets text to describe the behavior of legitimate and automated bot accounts. The authors used three different algorithms for mining contrast patterns by using the option of filtering. They tested their results in four datasets describing tweets issued by political figures. The authors showed that their proposal of using the Random Forest miner jointly with the classifiers PBC4cip [127] obtains significantly better classification results than ten popular state-of-the-art classifiers. The main difference of this work with the proposed by [122] is that [129] proposed a new feature representation based on the frequency of the issued tweets.

In summary, CPs-based algorithms are as accurate as traditional classifiers (e.g., SVM, C4.5, k-NN, NB, BN) but most of the applications test on a single dataset, which is not public. Most of the works do not use any statistical test, and they compare with three or fewer classifiers. The works of Loyola-González et al. [126, 129] stand out as exception because they use multiple datasets (some of them are public), they use statistical tests, and they compare with at least ten different classifiers.

## 4 Trends and Opportunities

In this section, we describe potential research directions for supervised classification, based on our reviews in Section 2. First, in Section 4.1, we discuss a number of existing CP-based approaches, including their limitations. Next, in Section 4.2, we identify under-explored areas and potential approaches that warrant further exploration.

### 4.1 Trends

From data of the last decade, we observed that many different approaches for each stage of CP-based supervised classification have been presented. Similarly, a number of exhaustive-search-based algorithms for extracting patterns from data streams or imbalanced databases have also been recently proposed. Although these proposals solve interesting problems, they continue to use initial discretization for all numerical features. As we have previously discussed, there are several limitations associated with the use of initial discretization [70, 71, 127]:

- Information loss caused by the limited possibility of transformed data interpretation.
- Additional and high-computational times for carrying out this task.
- Undesirable results, such as the omission of patterns, because the values of other features could hide essential relations in the objects of a class.
- All extracted patterns only have items of the form  $[f_i = v_j]$  and do not take into account other relational operators, subtracting discriminative power from patterns.
- The features containing missing values cannot be effectively handled; consequently, the extracted patterns may not be representative of the problem.

We also noted that there have been attempts to design approaches for extracting CPs from decision trees. Such approaches remove the need for initial discretization, and hence reduce the search space of potential patterns significantly. In addition, such approaches can also handle missing values, and generate a small collection of high-quality patterns. It has also been shown in the literature that patterns extracted from decision trees can help achieve better classification results than those extracted from exhaustive-search-based algorithms [70, 71, 127].

At the data level, other trending approaches, such as bagging and resampling methods, have been used for extracting CPs. The main idea is to use the advantages of these approaches for dealing with noisy objects and imbalanced databases before (or during) the mining stage. Researchers [5, 70, 124] have shown that applying these approaches allow one to obtain CPs that improve the classification results. However, these approaches could promote the extraction of patterns which are not representative of the problem. For example, if a CP is extracted using only a subset of the training dataset, then the particular CP is only representative for the particular subset and not of the original training dataset. Even more, a CP could stop being a CP to be only a pattern because of the support of that CP could significantly change when it is updated by using the original training dataset.

For the filtering stage, we observed that approaches based on set theory are essential in this stage, due to their ability in obtaining a subset of high-quality patterns (like the subset of minimal patterns). This allows one to obtain good classification results. However, in some contexts, these approaches may obtain a subset of patterns that do not cover all objects of the training dataset. Consequently, classification results may degrade. On the other hand, pattern filtering approaches based on QMs can obtain a collection of high-quality patterns, which covers all objects of the training dataset (e.g., the covering algorithm). However, filtering approaches based on QMs cannot guarantee that a QM for ranking CPs obtains the same behavior for two different problems; hence, selecting a QM as the best one for ranking patterns in any context would produce undesired results.

Filtering algorithms allow having a set of CPs with significantly fewer patterns than using all extracted patterns, but their main drawback is that the number of selected patterns is yet impractical to be analyzed by experts; depending on the database, it ranges from 100 to several thousand of patterns. The fewer patterns, the more understandable the model. Although usually, the fewer patterns, the worse classification results. A possible explanation is that the mining stage cannot provide a set of high-quality patterns; hence, more patterns are necessary for obtaining good classification results.

Finally, at the classification stage, we have observed that several pattern-based classifiers are using a scoring function for integrating the votes

coming from a collection of patterns. As we have stated in Section 2.3, those classifiers using a scoring function based on the support can be affected by some practical problems, for example, the class imbalance problem [124]. As a consequence, these classifiers could bias their classification results toward one class while avoiding the remaining classes

On the one hand, some classifiers introduced a variant to weight the scoring function, like PBC4cip [127], BCEP [57], and CACSP [130], to avoid the bias classification presented by traditional scoring functions. These variants have shown high classification results than other proposals. The main idea is that those patterns with low support for one class do not become overwhelmed by those patterns with high support for the remaining classes. This is possible by rewarding the support of the class with fewer patterns while punishing the support of those patterns belonging to the remaining classes.

On the other hand, there exist the FEPC classifier [68], which modifies the support with the aim of making it suitable for fuzzy problems. In this way, an object supports a fuzzy pattern according to the membership degree of that object for each item belonging to that pattern. The FEPC classifier has shown better classification results than other cp-based classifiers. A possible explanation is that FEPC takes into account the relationships among items based on a membership degree, avoiding the crisp way used by other cp-based classifiers (which could be very restrictive).

Other pattern-based classifiers showing good classification results are OCLEP and OCLEP+. These classifiers rely on the quality measure length (i.e., the number of items of a pattern) for building a one-class classification model. OCLEP and OCLEP+ were tested using only an exhaustive-search-based algorithm and only a few databases; consequently, their classification results could change when they use patterns extracted from other mining algorithms, which not use an initial discretization. The main reason is that in some context, patterns extracted from exhaustive-search-based algorithms have more items than those patterns coming from decision trees based algorithms, due to some filtering strategies cannot be applied (e.g., removing redundant items). As a consequence, both OCLEP and OCLEP+ must be tested using suitable experimentation in order to show their performances as a pattern-based classifier for one-class problems.

## 4.2 Challenges

An important key in any paper is to show future research directions with the aim of promoting the development of new proposals. In this section, we state some of those proposal based on CPs that have been little studied as well their possible improving items. Also, we point out possible approaches for exploring in the supervised classification based on CPs.

From our review, fuzzy pattern-based classifiers have shown to make consistently more accurate predictions than other popular non-fuzzy pattern-based classifiers. However, this approach has been little studied; only three classifiers based on fuzzy patterns have been proposed [68]. Fuzzy patterns look more similar to the language used by experts than other types of patterns. This is possible thanks to the use of linguistic hedges (e.g., “very,” “often,” and “somewhat”), which are commonly used for fixing the discretization of continuous features. An interesting point to explore is to develop new linguistic hedges, which can be extracted from interaction with experts. Another point to explore is to modify the algorithm proposed by [68] for mining fuzzy patterns from decision trees in order to use a variant of the C4.5 algorithm [157] instead of the current variant using ID3. As was stated by [157], the C4.5 algorithm improves the ID3 algorithm significantly.

Another interesting key is that pattern-based classification for one-class problems has not been enough studied. For example, OCLEP+ [51] is a recent pattern-based proposal for one-class problems. Nevertheless, as we have stated before, OCLEP+ continues using an initial discretization before mining CPs, which produces many handicaps. A possible solution is to use a One-Class Random Forest (OCRF) as proposed by [42] for inducing several decision trees and after that, mining CPs from these decision trees; in this way, the initial discretization is removed.

In the same vein, many algorithms for mining CPs from data streams have been proposed [23–26, 83]. Nevertheless, they continue using an initial discretization and an exhaustive-search-based algorithm for mining CPs, which are limiting the power of these proposals. Hence, an alternative for solving it is to use a strategy for building fast decision trees, like the proposal of [195] using an adaptive tie threshold and an

incremental pruning, or another one proposed by [104] for building fast decision trees in one-class problems.

As we have stated in Section 2.1.2, CP-based classification has reported good classification results in several contexts by using CPs extracted from decision trees, but an interesting point to explore is to build new decision trees induction procedures specifically designed for extracting CPs. A starting point could be the modification of the measure for evaluating candidate splits, the pruning procedure, and the stop conditions in the tree induction procedure with the aim of producing trees fulfilling the condition of CP in any path from the root node to the leaf nodes during the induction procedure and not after that.

As was argued by López et al. [116], there are data intrinsic characteristics such as noisy data, overlapping among the classes, and data fragmentation that is affecting the classification results of the cp-based classifiers on imbalanced databases. Hence, another line of future research could be to focus on detecting these intrinsic characteristics before mining CPs in order to improve the cp-based classification. In the same vein, from several experiments conducted in [121, 123, 124, 127], we can notice that there are some databases where every cp-based classifier attains good classification results; on the contrary, there are other databases where the classification results are bad. It could be vital to find an automated procedure to detect those cases, based on some database information. A meta-analysis study could be an initial point for carrying it out.

Pattern-based classifiers have reported good classification results, in class imbalance problems, when patterns are extracted from a resampled database [5, 6], an interesting point to explore is how to create resampling methods designed specifically for improving the task of CP mining, even if these methods are not good for improving other classification tasks. The main challenge for that is to extract a small collection of high-quality CPs, which should be representative of all training dataset. A starting point could be updating the support of all patterns extracted from each subsample regarding the original training dataset and, in this way, only those patterns fulfilling the condition of CPs are selected in the final collection.

In [27] showed that CPs extracted from multivariate decision trees allow obtaining better classification results than those CPs extracted from univariate decision trees. An important point to explore is

to extract CPs from multivariate decision trees with higher polynomial degrees than those tested by [27] to find if it can improve the obtained classification results. However, it is essential to continue obtaining an understandable pattern-based model from decision trees, which could be affected by using high polynomial degrees or using data transformation, like the rotation forest algorithm proposing by [161]. Obtaining an understandable model helps model credibility and even acceptability in practical contexts. Hence, it would be interesting to explore using fuzzy multivariate patterns for classification. The main idea is to extract multivariate patterns from other algorithms for inducing fuzzy decision trees such those reviewed by [99] or fusing polynomial, fuzzy logic, and decision tree structures like the one proposed by [139].

As far as we know, the first algorithm for mining cost-sensitive patterns was proposed by [130]. It looks an attractive idea to be deeply studied due to in practical problems the experts need to rely on a cost matrix, which is commonly depending on the problem. Providing an understandable model for this type of problems would help to experts to take actions by using discriminate power of the pattern jointly the associated cost matrix. An alternative to explore is to use the test cost approach, which computes the expected cost by adding the cost associated with each feature used in the decision nodes traversed from the root to the leaves for classifying a query object. The main handicap of this type of problem is to find the cost matrices associated with each feature of the problem.

Filtering CPs by using quality measures for patterns has shown interesting results. However, further work is needed to analyze which quality measures can create a ranking of patterns in a similar way as an expert in the domain application does. It would help to select a collection of high-quality pattern similar to how the experts do it in the application domain.

The most interesting point to explore at the filtering stage is to obtain a small subset of high-quality patterns that meet the following conditions: (i) allows obtaining good classification results and (ii) providing a reasonable amount of patterns to be interpreted by experts in the application domain. Usually, from our review, we notice that when the first condition is fulfilled, then the second one fails by far.

The main advantages of cp-based classification are the good classification results and the possibility of

providing an understandable model, being this latter the desired property in practical domains.

As we have stated in Section 3, several papers are applying the cp-based classification to real-world problems. However, nowadays, with the advancement of technology, there are emerging new real-world problems where the contrast pattern-based classification has been little (or none) applied.

The technology revolution has been facilitating the generation of several data using digital devices, resulting in what has been called “big data” [3]. On the one hand, big data can be obtained from stored data in disks where a few cp-based proposals have shown their advances in this topic; like the one proposed by [74]. Nevertheless, there are not recent studies for making the cp-based classification suitable to big data. An alternative to do it is to use the MapReduce paradigm [41]. MapReduce not only allows dealing with Big Data, but it is also a paradigm allowing the creation of scalable methods [192].

On the other hand, big data can be provided from sensors or frameworks in the form of data streams [166]. For this type of problem, there are several theoretical proposals such proposed by [23–26] and [169]. Nevertheless, these algorithms only were proved using synthetic data (or no real data streams), and commonly there are new real-world problems where these algorithms should be proved.

Some interesting practical contexts for applying the cp-based classification from data streams are:

- One may want to query, in real time, about the top-2 most frequent political hashtags of twitter [164] every 10 minutes, to predict a political campaign.
- Obtaining, in real time, the top-5 frequent products consulted by users in Amazon.com every minute, to promote sales.
- Selecting, in real time, the top-3 frequent news from Facebook.com, Google News, and CNN, to analyze trending news.

Notice that from the items mentioned above the main advantage of the cp-based classification is to provide an explainable model, where the final user can take actions based on the patterns’s interpretability.

Finally, from this section, we can conclude that there are several works in the cp-based classification topic, but they leave an improving room at each stage; mining, filtering, and classification. For every stage, we have provided future research directions to

increase the obtained results in the pattern-based classification. Additionally, we have promoted some new real-world applications for applying the pattern-based classification, which could be suitable for showing both discriminative power and interpretability of CPs in practical contexts.

## 5 Conclusion

The renewed interest in artificial intelligence approaches, such as those based on machine learning or deep learning, is partly due to their applications in a broad range of applications such as parallel and distributed systems. Therefore, in this paper, we performed an in-depth review and a scientometric study of CP-based supervised classification approaches. In addition to the findings (e.g. taxonomy) discussed in this paper and summarized in the below bullet-points, potential research opportunities were also presented in Section 4.

### Advantages:

- Based on our scientometric study, we observed that pattern-based classification continues to attract interest from the research community, with several hundreds publications annually between 2017 and 2019.
- CPs are useful for applications where end-users require pattern explanation.
- CP-based techniques based on decision trees have been shown to obtain better classification results.
- Fuzzy CP-based classifiers have shown better classification results, in comparison to other classifiers based on patterns. However, this type of classifiers should be studied further, for example by exploring the potential of utilizing linguistic hedges (may be more related to language used by experts than other types of patterns).
- From our taxonomy, we concluded that most of the existing algorithms extract CPs based on exhaustive search, despite algorithms based on decision trees having better accuracy and Area Under the Curve (AUC).

**Disadvantages:**

- CP-based techniques using initial discretization appear to suffer from limitations such as information loss, high-computational time, not capable of finding patterns, subtracting discriminative power from patterns, and difficulty in handling missing values. Such limitations may be overcome when CPs are extracted from decision trees.
- Several filtering algorithms for CPs have been proposed, but the number of patterns provided by filtering algorithms is still impractical to be analyzed by experts in the application domain.
- From our review, we observed that there are fewer proposals related to the filtering stage than the mining and classification stages. Hence, developing new filtering methods for contrast patterns also appear to be a topic that requires further attention.
- CP-based classifiers that use a scoring function as the classification strategy may suffer from class imbalance problems. Consequently, there have been attempts to design CP-based classifiers using a weighted scoring function.

**Notations**

The notations used in this paper are listed here.

API	Application Programming Interface
CEP	Bayes-based Classification using Emerging Patterns
BEP	Balanced emerging pattern
BFS	Breadth First Search
BN	Bayesian Network
BSC-tree	Bit String Compression tree
CACSP	Classification by Aggregating Cost-Sensitive Patterns
CAEP	Classification by Aggregating Emerging Patterns
CEPM	Cascade Emerging Pattern Miner
CMAR	Classification based on Multiple Class-Association Rules

CPAR	Classification based on Predictive Association Rules
CP	Contrast pattern
CSP	Cost-sensitive pattern
D2D-Apriori	Dynamic queue & Deep Parallel Apriori
DBF	Delete Best Feature
DBP	Delete Best Property
DBPL	Delete Best Property by Level
DFS	Depth First Search
DGCP-Tree	Dynamically Growing Contrast Pattern Tree
EP	Emerging Pattern
EPRFm	Emerging Patterns with Random Forest miner
EJEP	Essential Jumping Emerging Pattern
EvAEP	Evolutionary Algorithm for Extracting Emerging Patterns
EvAEFP-Spark	Evolutionary Algorithm for Extracting Emerging Patterns using Spark
HRFm	Hellinger Random Forest miner
FCI-Outlier	An Frequent Closed Itemset-based Outlier detecting approach
FEP	Fuzzy Emerging Pattern
FEPm	Fuzzy Emerging Pattern Miner
FEPC	Fuzzy Emerging Pattern-based Classifier
FPR	False Positive Rate
HMM	Hidden Markov Models
iCAEP	Information-based Classification by Aggregating Emerging Patterns
JEP	Jumping Emerging Pattern
JEPN	Jumping Emerging Pattern with Negation
LCMine	Logical Complex Miner
LexOrdMiner	Lexicographic Order Miner
MCP	Multivariate contrast pattern
MOEA-EFEP	Multi-Objective Evolutionary Algorithm for Extracting Fuzzy Emerging Patterns
NB	Naive Bayes



OCLEP	One-class Classification using Length statistics of Emerging Patterns
OCRf	One-Class Random Forest
ocSVM	one-class Support Vector Machine
ODFP-tree	Ordered Dynamic Frequent Pattern tree
PCP	Pan-Correlation Pattern
PBC4cip	Pattern-based Classification for class imbalanced problems
kg	Kilogram
RFm	Random Forest miner
SMOTE-TL+LCMine	Synthetic Minority Over-sampling Technique using Tomek Links before applying Logical Complex Miner
SVM	Support Vector Machine
UDFP-tree	Unordered Dynamic Frequent Pattern tree
QM	Quality Measure
SEP	Strong Emerging pattern
SJEP	Strong Jumping Emerging Pattern
SysTreeMiner	Systolic Tree Miner
TPR	True Positive Rate
WBPEM	Windows Balanced Emerging Pattern Miner

## References

1. Abdelhamid, N., Ayesh, A., Thabtah, F.: Phishing detection based associative classification data mining. *Expert Syst. Appl.* **41**(13), 5948–5959. <https://doi.org/10.1016/j.eswa.2014.03.019>. <http://www.sciencedirect.com/science/article/pii/S0952197614001481> (2014)
2. Agrawal, R., Imieliński, 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, SIGMOD '93*, pp. 207–216. ACM, New York. <https://doi.org/10.1145/170035.170072> (1993)
3. Alam, S., Yao, N.: Big data analytics, text mining and modern english language. *J. Grid Comput.* **17**(2), 357–366 (2019). <https://doi.org/10.1007/s10723-018-9452-4>
4. Alavi, F., Hashemi, S.: Dfp-sepsf: A dynamic frequent pattern tree to mine strong emerging patterns in streamwise features. *Eng. Appl. Artif. Intell.* **37**, 54–70. <https://doi.org/10.1016/j.engappai.2014.08.010>

5. Alhammady, H.: A Novel Approach for Mining Emerging Patterns in Rare-Class Datasets. In: Sobh, T. (ed.) *Innovations and Advanced Techniques in Computer and Information Sciences and Engineering*, pp. 207–211. Springer, Netherlands (2007)
6. Alhammady, H., Ramamohanarao, K., Zhang, C.: The Application of Emerging Patterns for Improving the Quality of Rare-Class Classification. In: Dai, H., Srikant, R. (eds.) *Advances in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science*, vol. 3056, pp. 207–211. Springer, Berlin (2004)
7. Alhammady, H., Ramamohanarao, K.: Mining Emerging Patterns and Classification in Data Streams. In: *The IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*, pp. 272–275. <https://doi.org/10.1109/WI.2005.96> (2005)
8. Alkadi, O.S., Moustafa, N., Turnbull, B., Choo, K.R.: An ontological graph identification method for improving localization of ip prefix hijacking in network systems. *IEEE Trans. Inf. Forensic. Secur.* **15**, 1164–1174 (2020). <https://doi.org/10.1109/TIFS.2019.2936975>
9. An, Z., Deng, W., Hu, J., Zhong, Y., Zhao, Y.: Apa: Adaptive pose alignment for pose-invariant face recognition. *IEEE Access* **7**, 14653–14670 (2019). <https://doi.org/10.1109/ACCESS.2019.2894162>
10. Bailey, J.: Statistical Measures for Contrast Patterns. In: Dong, G., Bailey, J. (eds.) *Contrast Data Mining: Concepts, Algorithms, and Applications*, chap. 2, pp. 13–20. Chapman & Hall/CRC, USA (2012)
11. Bailey, J., Manoukian, T., Ramamohanarao, K.: Fast Algorithms for Mining Emerging Patterns. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) *Principles of Data Mining and Knowledge Discovery*, pp. 39–50. Berlin, Springer (2002)
12. Barandiaran, I.: The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(8) (1998)
13. Barman, S., Shum, H.P.H., Chattopadhyay, S., Samanta, D.: A secure authentication protocol for multi-server-based e-healthcare using a fuzzy commitment scheme. *IEEE Access* **7**, 12557–12574 (2019). <https://doi.org/10.1109/ACCESS.2019.2893185>
14. Baum, L.E., Petrie, T.: Statistical inference for probabilistic functions of finite state markov chains. *Ann. Math. Stat.* **37**(6), 1554–1563. <http://www.jstor.org/stable/2238772> (1966)
15. Bayardo Jr., R.J.: Efficiently mining long patterns from databases. In: *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, SIGMOD '98*, pp. 85–93. ACM, New York. <https://doi.org/10.1145/276304.276313> (1998)
16. Cuissart, B., Poezevara, G., Lepailleur, A., Bureau, R.: Emerging Patterns as Structural Alerts for Computational Toxicology. In: Dong, G., Bailey, J. (eds.) *Contrast Data Mining: Concepts, Algorithms, and Applications, Data Mining and Knowledge Discovery Series*, chap. 19, pp. 269–282. USA, Chapman & Hall/CRC (2012)

17. Bongard, M.: Solution to geological problems with support of recognition programs. *Sov Geol.* **6**, 33–50 (1963)
18. Bouadjeneq, M.R., Hacid, H., Bouzeghoub, M.: Social networks and information retrieval, how are they converging? a survey, a taxonomy and an analysis of social information retrieval approaches and platforms. *Inf. Syst.* **56**(0), 1–18 (2016)
19. Breiman, L.: Bagging predictors. *Mach. Learn.* **24**(2), 123–140 (1996). <https://doi.org/10.1007/BF00058655>
20. Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32. <https://doi.org/10.1023/A:1010933404324> (2001)
21. Broder, A.Z., Charikar, M., Frieze, A.M., Mitzenmacher, M.: Min-wise independent permutations. *J. Comput. Syst. Sci.* **60**(3), 630–659 (2000). <https://doi.org/10.1006/jcss.1999.1690>. <http://www.sciencedirect.com/science/article/pii/S002200009916902>
22. Brodley, C.E., Utgoff, P.E.: Multivariate decision trees. *Mach. Learn.* **19**(1), 45–77 (1995). <https://doi.org/10.1007/BF00994660>
23. Bustio, L., Cumplido, R., Hernández, R., Bande, J.M., Feregrino, C.: Frequent Itemsets Mining in Data Streams Using Reconfigurable Hardware. In: Ceci, M., Loglisci, C., Manco, G., Masciari, E., Ras, Z.W. (eds.) *New Frontiers in Mining Complex Patterns*, pp. 32–45. Springer International Publishing (2016)
24. Bustio-Martínez, L., Cumplido, R., Hernández-León, R., Bande-Serrano, J.M., Feregrino-Urbe, C.: On the design of hardware-software architectures for frequent itemsets mining on data streams. *J. Intell. Inf. Syst.* **50**(3), 415–440 (2018). <https://doi.org/10.1007/s10844-017-0461-8>
25. Bustio-Martínez, L., Cumplido, R., Letras-Luna, M., Uribe, C.F., Hernández-León, R., Bande-Serrano, J.M.: Approximate Frequent Itemsets Mining on Data Streams Using Hashing and Lexicographic Order in Hardware. In: 2017 IEEE 8th Latin American Symposium on Circuits Systems (LASCAS), pp. 1–4. <https://doi.org/10.1109/LASCAS.2017.7948076> (2017)
26. Bustio-Martínez, L., Letras-Luna, M., Cumplido, R., Hernández-León, R., Feregrino-Urbe, C., Bande-Serrano, J.M.: Using hashing and lexicographic order for frequent itemsets mining on data streams. *J. Parallel Distrib. Comput.* **125**, 58–71. <https://doi.org/10.1016/j.jpdc.2018.11.002>. <http://www.sciencedirect.com/science/article/pii/S0743731518308244> (2019)
27. Cañete-Sifuentes, L., Monroy, R., Medina-Pérez, M.A., Loyola-González, O., Voronisky, F.V.: Classification based on multivariate contrast patterns. *IEEE Access* **7**(1), 55744–55762 (2019). <https://doi.org/10.1109/ACCESS.2019.2913649>
28. Cafaro, M., Epicoco, I., Pulimeno, M.: Mining frequent items in unstructured p2p networks. *Fut. Gener. Comput. Syst.* **95**, 1–16. <https://doi.org/10.1016/j.future.2018.12.030>. <http://www.sciencedirect.com/science/article/pii/S0167739X18315838> (2019)
29. Cavadenti, O., Codocedo, V., Boulicaut, J., Kaytoue, M.: What Did I Do Wrong in My Moba Game? Mining Patterns Discriminating Deviant Behaviours. In: 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA), pp. 662–671. <https://doi.org/10.1109/DSAA.2016.75> (2016)
30. Cervantes, B., Gómez, F., Monroy, R., Loyola-González, O., Medina-Pérez, M.A., Ramírez-Márquez, J.: Pattern-Based and Visual Analytics for Visitor Analysis on Websites. *Appl. Sci.* **9**(18). <https://doi.org/10.3390/app9183840>. <https://www.mdpi.com/2076-3417/9/18/3840> (2019)
31. Chavary, E.A., Erfani, S.M., Leckie, C.: Summarizing significant changes in network traffic using contrast pattern mining. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM '17, pp. 2015–2018. ACM, New York. <https://doi.org/10.1145/3132847.3133111> (2017)
32. Chen, C.H. *Handbook of Pattern Recognition and Computer Vision*, 5th edn. World Scientific Publishing Co., Inc., River Edge (2016)
33. Chen, L., Dong, G.: Masquerader Detection Using Oclep: One-Class Classification Using Length Statistics of Emerging Patterns. In: Seventh International Conference on Web-Age Information Management Workshops, pp. 5–5. <https://doi.org/10.1109/WAIMW.2006.19> (2006)
34. Chen, T.T., Lee, S.J.: A weighted ls-svm based learning system for time series forecasting. *Inform. Sci.* **299**(0), 99–116 (2015)
35. Chen, X., Liu, Z.: Finding Contrast Patterns in Imbalanced Classification based on Sliding Window. In: Zhu, S.H. (ed.) Proceedings of the 4th International Conference on Mechanical Materials and Manufacturing Engineering (MMME 2016), *Advances in Engineering Research*, vol. 79, pp. 161–166. Atlantis Press (2016). <https://doi.org/10.2991/mmme-16.2016.36>
36. Cheng, J., Ke, Y., Ng, W.: A survey on algorithms for mining frequent itemsets over data streams. *Knowl. Inf. Syst.* **16**(1), 1–27 (2008). <https://doi.org/10.1007/s10115-007-0092-4>
37. Cieslak, D.A., Chawla, N.V.: Learning decision trees for unbalanced data. In: Daelemans, W., Goethals, B., Morik, K. (eds.) Proceedings of the Machine Learning and Knowledge Discovery in Databases: European Conference (ECML PKDD 2008), pp. 241–256. Springer, Berlin (2008). [https://doi.org/10.1007/978-3-540-87479-9\\_34](https://doi.org/10.1007/978-3-540-87479-9_34)
38. Cohen, W.W.: Fast effective rule induction. In: Proceedings of the Twelfth International Conference on International Conference on Machine Learning, ICML'95, pp. 115–123. Morgan Kaufmann Publishers Inc., San Francisco. <http://dl.acm.org/citation.cfm?id=3091622.3091637> (1995)
39. Compton, K., Hauck, S.: Reconfigurable computing: A survey of systems and software. *ACM Comput. Surv.* **34**(2), 171–210 (2002). <https://doi.org/10.1145/508352.508353>
40. Coquin, L., Canipa, S.J., Drewe, W.C., Fisk, L., Gillet, V.J., Patel, M., Plante, J., Sherhod, R.J., Vessey, J.D.: New structural alerts for Ames mutagenicity discovered using emerging pattern mining techniques. *Toxicol. Res.* **4**, 46–56 (2015). <https://doi.org/10.1039/C4TX00071D>
41. Dean, J., Ghemawat, S.: Mapreduce: Simplified data processing on large clusters. *Commun. ACM* **51**(1), 107–113 (2008). <https://doi.org/10.1145/1327452.1327492>
42. Désir, C., Bernard, S., Petitjean, C., Heutte, L.: One class random forests. *Pattern Recogn.* **46**(12), 3490–3506. <https://doi.org/10.1016/j.patrec.2013.08.025>

- [doi.org/10.1016/j.patcog.2013.05.022](https://doi.org/10.1016/j.patcog.2013.05.022). <http://www.science/article/pii/S003132031300246X> (2013)
43. Dietterich, T.G.: Ensemble Methods in Machine Learning. In: Multiple Classifier Systems SE - 1, Lecture Notes in Computer Science, vol. 1857, pp. 1–15. Springer, Berlin. [https://doi.org/10.1007/3-540-45014-9\\_1](https://doi.org/10.1007/3-540-45014-9_1) (2000)
  44. Djenouri, Y., Belhadi, A., Fournier-Viger, P.: Extracting useful knowledge from event logs: A frequent itemset mining approach. *Knowl.-Based Syst.* **139**, 132–148. <https://doi.org/10.1016/j.knosys.2017.10.016>. <http://www.sciencedirect.com/science/article/pii/S0950705117304860> (2018)
  45. Domingos, P.: MetaCost: a general method for making classifiers cost-sensitive. In: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 155–164. ACM, San Diego (1999)
  46. Dong, G.: Preliminaries. In: Dong, G., Bailey, J. (eds.) *Contrast Data Mining: Concepts, Algorithms, and Applications*, Data Mining and Knowledge Discovery Series, chap. 1, pp. 3–12. USA, Chapman & Hall/CRC (2012)
  47. Dong, G.: Exploiting the power of group differences: Using patterns to solve data analysis problems. *Synth. Lect. Data Min. Knowl. Discov.* **11**(1), 1–146 (2019). <https://doi.org/10.2200/S00897ED1V01Y201901DMK016>
  48. Dong, G., Bailey, J.: *Contrast Data Mining: concepts, Algorithms, and Applications*, 1st edn. Chapman & Hall/CRC (2012)
  49. Dong, G., Li, J.: Efficient mining of emerging patterns: discovering trends and differences. In: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '99, pp. 43–52. ACM, New York (1999)
  50. Dong, G., Li, J., Wong, L.: The Use of Emerging Patterns in the Analysis of Gene Expression Profiles for the Diagnosis and Understanding of Diseases. In: *New Generation of Data Mining Applications*, Chap. 14, pp. 331–354. Wiley, New York (2004)
  51. Dong, G., Pentukar, S.K.: Oclep+: One-class anomaly and intrusion detection using minimal length of emerging patterns. arXiv:1811.09842 (2018)
  52. Dong, G., Zhang, X., Wong, L., Li, J.: Caep: Classification by aggregating emerging patterns. In: Arikawa, S., Furukawa, K. (eds.) *Proceedings of the Second International Conference on Discovery Science (DS'99)*, pp. 30–42. Springer, Berlin (1999). [https://doi.org/10.1007/3-540-46846-3\\_4](https://doi.org/10.1007/3-540-46846-3_4)
  53. Dua, D., Graff, C.: UCI machine learning repository. <http://archive.ics.uci.edu/ml> (2019)
  54. El Mazouri, F.Z., Abounaima, M.C., Zenkouar, K.: Data mining combined to the multicriteria decision analysis for the improvement of road safety: case of france. *J. Big Data* **6**(1), 5 (2019). <https://doi.org/10.1186/s40537-018-0165-0>
  55. Fan, H., Fan, M., Ramamohanarao, K., Liu, M.: Further Improving Emerging Pattern Based Classifiers via Bagging. In: Ng, W.K., Kitsuregawa, M., Li, J., Chang, K. (eds.) *Advances in Knowledge Discovery and Data Mining*, pp. 91–96. Springer, Berlin (2006)
  56. Fan, H., Kotagiri, R.: An efficient single-scan algorithm for mining essential jumping emerging patterns for classification. In: Chen, M.S., Yu, P.S., Liu, B. (eds.) *Advances in Knowledge Discovery and Data Mining: Proceedings of the 6th Pacific-Asia Conference (PAKDD 2002)*, pp. 456–462. Springer, Berlin (2002). [https://doi.org/10.1007/3-540-47887-6\\_45](https://doi.org/10.1007/3-540-47887-6_45)
  57. Fan, H., Ramamohanarao, K.: A bayesian approach to use emerging patterns for classification. In: *Proceedings of the 14th Australasian Database Conference - Volume 17, ADC '03*, pp. 39–48. Australian Computer Society, Inc., Darlinghurst (2003)
  58. Fan, H., Ramamohanarao, K.: Efficiently Mining Interesting Emerging Patterns. In: Dong, G., Tang, C., Wang, W. (eds.) *Advances in Web-Age Information Management*, pp. 189–201. Springer, Berlin (2003)
  59. 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.* **18**(6), 721–737 (2006). <https://doi.org/10.1109/TKDE.2006.95>
  60. Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continuous-valued attributes for classification learning. In: *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pp. 1022–1027, San Francisco (1993)
  61. Feng, M., Dong, G.: Incremental Maintenance of Emerging Patterns. In: Dong, G., Bailey, J. (eds.) *Contrast Data Mining: Concepts, Algorithms, and Applications*, Data Mining and Knowledge Discovery Series, chap. 7, pp. 69–86. Chapman & Hall/CRC (2012)
  62. Fournier-Viger, P., Lin, J.C.W., Vo, B., Chi, T.T., Zhang, J., Le, H.B.: A survey of itemset mining. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **7**(4), e1207 (2017). <https://doi.org/10.1002/widm.1207>
  63. Frank, E., Witten, I.H.: Generating accurate rule sets without global optimization. In: *Proceedings of the Fifteenth International Conference on Machine Learning, ICML '98*, pp. 144–151. Morgan Kaufmann Publishers Inc., San Francisco. <http://dl.acm.org/citation.cfm?id=645527.657305> (1998)
  64. Gao, C., Wang, J., He, Y., Zhou, L.: Efficient mining of frequent sequence generators. In: *Proceedings of the 17th International Conference on World Wide Web, WWW '08*, pp. 1051–1052. ACM, New York. <https://doi.org/10.1145/1367497.1367651> (2008)
  65. García-Borroto, M., Loyola-González, O., Martínez-Trinidad, J., Carrasco-Ochoa, J.: Comparing Quality Measures for Contrast Pattern Classifiers. In: Ruiz-Shulcloper, J., Sanniti di Baja, G. (eds.) *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications SE - 39, Lecture Notes in Computer Science*, vol. 8258, pp. 311–318. Springer, Berlin (2013)
  66. García-Borroto, M., Loyola-González, O., Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A.: Evaluation of quality measures for contrast patterns by using unseen objects. *Expert Syst. Appl.* **83**, 104–113. <https://doi.org/10.1016/j.eswa.2017.04.038>. <http://www.sciencedirect.com/science/article/pii/S0957417417302877> (2017)
  67. García-Borroto, M., Martínez-Trinidad, J., Carrasco-Ochoa, J.: Cascading an Emerging Pattern Based Classifier. In: Martínez-Trinidad, J., Carrasco-Ochoa, J., Kittler, J. (eds.) *Advances in Pattern Recognition*,

- Lecture Notes in Computer Science, vol. 6256, pp. 240–249. Springer, Berlin (2010). [https://doi.org/10.1007/978-3-642-15992-3\\_26](https://doi.org/10.1007/978-3-642-15992-3_26)
68. García-Borroto, M., Martínez-Trinidad, J., Carrasco-Ochoa, J.: Fuzzy emerging patterns for classifying hard domains. *Knowl. Inf. Syst.* **28**(2), 473–489 (2011). <https://doi.org/10.1007/s10115-010-0324-x>
  69. García-Borroto, M., Martínez-Trinidad, J., Carrasco-Ochoa, J.: A survey of emerging patterns for supervised classification. *Artif. Intell. Rev.* **42**(4), 705–721 (2014)
  70. García-Borroto, M., Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A.: Finding the best diversity generation procedures for mining contrast patterns. *Expert Syst. Appl.* **42**(11), 4859–4866 (2015). <https://doi.org/10.1016/j.eswa.2015.02.028>. <http://www.sciencedirect.com/science/article/pii/S0957417415001359>
  71. García-Borroto, M., Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A., Medina-Pérez, M.A., Ruiz-shulcloper, J.: LCMine: An efficient algorithm for mining discriminative regularities and its application in supervised classification. *Pattern Recogn.* **43**(9), 3025–3034 (2010). <https://doi.org/10.1016/j.patcog.2010.04.008>
  72. García-Vico, A.M., Carmona, C.J., González, P., del Jesus, M.J.: Moea-efep: Multi-objective evolutionary algorithm for extracting fuzzy emerging patterns. *IEEE Trans. Fuzzy Syst.* **26**(5), 2861–2872 (2018). <https://doi.org/10.1109/TFUZZ.2018.2814577>
  73. García-Vico, A.M., Carmona, C.J., Martín, D., García-Borroto, M., Del Jesus, M.J.: An overview of emerging pattern mining in supervised descriptive rule discovery: taxonomy, empirical study, trends, and prospects. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **8**(1) e1231–n/a. <https://doi.org/10.1002/widm.1231> (2017)
  74. García-Vico, A.M., González, P., del Jesus, M.J., Carmona, C.J.: A First Approach to Handle Fuzzy Emerging Patterns Mining on Big Data Problems: The EvaeFP-Spark Algorithm. In: *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1–6. <https://doi.org/10.1109/FUZZ-IEEE.2017.8015673> (2017)
  75. García-Vico, A.M., Montes, J., Aguilera, J., Carmona, C.J., del Jesus, M.J.: Analysing Concentrating Photovoltaics Technology through the Use of Emerging Pattern Mining. In: Graña, M., López-Guede, J.M., Etxaniz, O., Herrero, Á., Quintián, H., Corchado, E. (eds.) *International Joint Conference SOCO'16-CISIS'16-ICEUTE'16*, pp. 334–344. Springer International Publishing (2017)
  76. Geng, L., Hamilton, H.: Choosing the Right Lens: Finding What is Interesting in Data Mining. In: Guillet, F., Hamilton, H. (eds.) *Quality Measures in Data Mining, Studies in Computational Intelligence*, vol. 43, pp. 3–24. Springer, Berlin (2007)
  77. Geng, L., Hamilton, H.J.: Interestingness Measures for Data Mining: A Survey. *ACM Comput. Surv.* **38**(3), 1–32 (2006)
  78. GhasemiGol, M., Ghaemi-Bafghi, A., Takabi, H.: A comprehensive approach for network attack forecasting. *Comput. Secur.* **58**(0), 83–105 (2016)
  79. González-Soler, L.J., Chang, L., Hernández-Palancar, J., Pérez-Suárez, A., Gomez-Barrero, M.: Fingerprint Presentation Attack Detection Method Based on a Bag-Of-Words Approach. In: Mendoza, M., Velastín, S. (eds.) *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pp. 263–271. Springer International Publishing, Cham (2018)
  80. Guillaume, S., Charnomordic, B.: Learning interpretable fuzzy inference systems with fispro. *IEEE Access* **181**(20), 4409–4427 (2019). <https://doi.org/10.1016/j.ins.2011.03.025>
  81. Han, J., Cheng, H., Xin, D., Yan, X.: Frequent pattern mining: current status and future directions. *Data Min. Knowl. Disc.* **15**(1), 55–86 (2007)
  82. 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, SIGMOD '00*, pp. 1–12. ACM, New York. <https://doi.org/10.1145/342009.335372> (2000)
  83. Hao, S., Cai, S., Sun, R., Li, S.: Fci-Outlier: an Efficient Frequent Closed Itemset-Based Outlier Detecting Approach on Data Stream. In: Sun, Y., Lu, T., Xie, X., Gao, L., Fan, H. (eds.) *Computer Supported Cooperative Work and Social Computing*, pp. 176–187. Springer, Singapore (2019)
  84. Hassanien, A.E., Al-Shammari, E.T., Ghali, N.I.: Computational intelligence techniques in bioinformatics. *Comput. Biol. Chem.* **47**(0), 37–47 (2013)
  85. Heath, D.G., Kasif, S., Salzberg, S.: Induction of oblique decision trees. In: *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pp. 1002–1007 (1993)
  86. Cheng, H., Han, J., Yan, X., Yu, P.S.: Efficient Direct Mining of Selective Discriminative Patterns for Classification. In: Dong, G., Bailey, J. (eds.) *Contrast Data Mining: Concepts, Algorithms, and Applications, Data Mining and Knowledge Discovery Series*, chap. 5, pp. 39–58. USA, Chapman & Hall/CRC (2012)
  87. Hood, W., Wilson, C.: The literature of bibliometrics, scientometrics, and informetrics. *Scientometrics* **52**(2), 291–314. <https://doi.org/10.1023/A:1017919924342> (2001)
  88. Hu, W., Chen, T., Shah, S.L.: Detection of frequent alarm patterns in industrial alarm floods using itemset mining methods. *IEEE Trans. Ind. Electron.* **65**(9), 7290–7300 (2018). <https://doi.org/10.1109/TIE.2018.2795573>
  89. Huang, J., Peng, M., Wang, H.: Topic detection from large scale of microblog stream with high utility pattern clustering. In: *Proceedings of the 8th Workshop on Ph. D. Workshop in Information and Knowledge Management*, pp. 3–10. ACM (2015)
  90. Huynh, X.H., Guillet, F., Blanchard, J., Kuntz, P., Briand, H., Gras, R.: A Graph-Based Clustering Approach to Evaluate Interestingness Measures: a Tool and a Comparative Study. In: Guillet, F., Hamilton, H. (eds.) *Quality Measures in Data Mining, Studies in Computational Intelligence*, vol. 43, pp. 25–50. Springer, Berlin (2007)
  91. Jia, Y., Bailey, J., Kotagiri, R., Leckie, C.: Pattern-based feature generation. *Feature Engineering for Machine Learning and Data Analytics*, 245 (2018)
  92. Jiang, X.R., Gruenwald, L.: Microarray gene expression data association rules mining based on bsc-tree and fis-tree. *Data Knowl. Eng.* **53**(1), 3–29. <https://doi.org/10.1016/j.datak.2004.06.011>. <http://www.sciencedirect.com/science/article/pii/S0169023X04001107>. *Biological Data Management* (2005)

93. Ho, K., Dong, R., Lee, G., Piao, M.: Emerging Pattern Based Prediction of Heart Diseases and Powerline Safety. In: Dong, G., Bailey, J. (eds.) *Contrast Data Mining: Concepts, Algorithms, and Applications*, Data Mining and Knowledge Discovery Series, chap. 23, pp. 329–336. USA, Chapman & Hall/CRC (2012)
94. Khemchandani, R., Sharma, S.: Robust least squares twin support vector machine for human activity recognition. *Appl. Soft Comput.* **47**, 33–46 (2016)
95. Kim, J., Choi, K., Kim, G., Suh, Y.: Classification cost: An empirical comparison among traditional classifier, Cost-Sensitive Classifier, and MetaCost. *Expert Syst. Appl.* **39**(4), 4013–4019. <https://doi.org/10.1016/j.eswa.2011.09.071>. <http://www.sciencedirect.com/science/article/pii/S0957417411013947> (2012)
96. Knobbe, A., Crémilleux, B., Fürnkranz, J., Scholz, M.: From Local Patterns to Global Models: The LeGo Approach to Data Mining. In: *International Workshop from Local Patterns to Global Models (ECML'08)*, pp. 1–16. Lego (2008)
97. Kobylński, Ł., Walczak, K.: Jumping Emerging Patterns with Occurrence Count in Image Classification. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) *Advances in Knowledge Discovery and Data Mining*, pp. 904–909. Springer, Berlin (2008)
98. Kobylński, Ł., Walczak, K.: Emerging Patterns and Classification for Spatial and Image Data. In: Dong, G., Bailey, J. (eds.) *Contrast Data Mining: Concepts, Algorithms, and Applications*, Data Mining and Knowledge Discovery Series, chap. 20, pp. 285–302. USA, Chapman & Hall/CRC (2012)
99. Kotsiantis, S.B.: Decision trees: a recent overview. *Artif. Intell. Rev.* **39**(4), 261–283 (2013). <https://doi.org/10.1007/s10462-011-9272-4>
100. Lau, J.H., Collier, N., Baldwin, T.: On-Line Trend Analysis with Topic Models: # Twitter Trends Detection Topic Model Online. In: *24Th International Conference on Computational Linguistics*, pp. 1519–1534, Mumbai (2012)
101. Le, T., Vo, B.: The lattice-based approaches for mining association rules: a review. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **6**(4), 140–151 (2016). <https://doi.org/10.1002/widm.1181>
102. Leemans, M., van der Aalst, W.M.: Discovery of Frequent Episodes in Event Logs. In: *International Symposium on Data-Driven Process Discovery and Analysis*, pp. 1–31. Springer (2014)
103. Leemans, S.J., Fahland, D., der Aalst, W.M.V.: Scalable process discovery and conformance checking. *Softw. Syst. Model.* **17**(2), 599–631 (2018)
104. Li, C., Zhang, Y., Li, X.: Ocvfdt: One-class very fast decision tree for one-class classification of data streams. In: *Proceedings of the Third International Workshop on Knowledge Discovery from Sensor Data, SensorKDD '09*, pp. 79–86. ACM, New York. <https://doi.org/10.1145/1601966.1601981> (2009)
105. Li, G., Law, R., Vu, H.Q., Rong, J., Zhao, X.R.: Identifying emerging hotel preferences using emerging pattern mining technique. *Tourism Manag.* **46**, 311–321. <https://doi.org/10.1016/j.tourman.2014.06.015>. <http://www.sciencedirect.com/science/article/pii/S0261517714001174> (2015)
106. Li, J., Dong, G., Ramamohanarao, K., Wong, L.: Deeps: A new instance-based lazy discovery and classification system. *Mach. Learn.* **54**(2), 99–124. <https://doi.org/10.1023/B:MACH.0000011804.08528.7d> (2004)
107. Li, J., Ramamohanarao, K., Dong, G.: The space of jumping emerging patterns and its incremental maintenance algorithms. In: *Proceedings of the Seventeenth International Conference on Machine Learning, ICML '00*, pp. 551–558. Morgan Kaufmann Publishers Inc., San Francisco. <http://dl.acm.org/citation.cfm?id=645529.657809> (2000)
108. Li, J., Wong, L.: Emerging Pattern Based Rules Characterizing Subtypes of Leukemia. In: Dong, G., Bailey, J. (eds.) *Contrast Data Mining: Concepts, Algorithms, and Applications*, Data Mining and Knowledge Discovery Series, chap. 15, pp. 219–232. Chapman & Hall/CRC (2012)
109. Li, W., Han, J., Pei, J.: Cmar: Accurate and efficient classification based on multiple class-association rules. In: *2001 IEEE International Conference on Data Mining, ICDM '01*, pp. 369–376. IEEE Computer Society, San Jose. <https://doi.org/10.1109/ICDM.2001.9895410>. <http://dl.acm.org/citation.cfm?id=645496.657866> (2001)
110. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In: *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, KDD'98*, pp. 80–86. AAAI Press. <http://dl.acm.org/citation.cfm?id=3000292.3000305> (1998)
111. Liu, B., Ma, Y., Wong, C., Yu, P.: Scoring the data using association rules. *Appl. Intell.* **18**(2), 119–135 (2003)
112. Liu, Q., Dong, G.: A Contrast Pattern Based Clustering Quality Index for Categorical Data. In: *Ninth IEEE International Conference on Data Mining*, pp. 860–865. <https://doi.org/10.1109/ICDM.2009.105> (2009)
113. Liu, Q., Ghosh, S., Li, J., Wong, L., Ramamohanarao, K.: Discovering pan-correlation patterns from time course data sets by efficient mining algorithms. *Computing* **100**(4), 421–437 (2018). <https://doi.org/10.1007/s00607-018-0606-9>
114. Liu, Q., Li, J., Wong, L., Ramamohanarao, K.: Efficient Mining of Pan-Correlation Patterns from Time Course Data. In: Li, J., Li, X., Wang, S., Li, J., Sheng, Q.Z. (eds.) *Advanced Data Mining and Applications*, pp. 234–249. Springer International Publishing (2016)
115. 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.* **45**(3), 598–615 (2014). <https://doi.org/10.1080/00207721.2012.724110>
116. López, V., Fernández, A., García, S., Palade, V., Herrera, F.: An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Inf. Sci.* **250**(0), 113–141 (2013)
117. López, V., Fernández, A., Herrera, F.: On the importance of the validation technique for classification with imbalanced datasets: Addressing covariate shift when data is skewed. *Inform. Sci.* **257**(0), 1–13 (2014)

118. López, V., Triguero, I., Carmona, C.J., García, S., Herrera, F.: Addressing imbalanced classification with instance generation techniques: IPADE-ID. *Neurocomputing* **126**(0), 15–28 (2014)
119. Loyola-González, O.: Black-Box vs. White-Box: Understanding Their Advantages and Weaknesses From a Practical Point of View. *IEEE Access* **7**(1), 154096–154113. <https://doi.org/10.1109/ACCESS.2019.2949286> (2019)
120. Loyola-González, O.: Understanding the Criminal Behavior in Mexico City through an Explainable Artificial Intelligence Model. In: Martínez-Villaseñor, L., Batyrshin, I., Marín-Hernández, A. (eds.) *Advances in Soft Computing*, pp. 136–149. Springer International Publishing, Cham (2019). [https://doi.org/10.1007/978-3-030-33749-0\\_12](https://doi.org/10.1007/978-3-030-33749-0_12)
121. Loyola-González, O., García-Borroto, M., Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A.: An empirical comparison among quality measures for pattern based classifiers. *Intell. Data Anal.* **18**(0), S5–S17. <https://doi.org/10.3233/IDA-140705>. <http://iospress.metapress.com/content/220668WV01J24660> (2014)
122. Loyola-González, O., López-Cuevas, A., Medina-Pérez, M.A., Camiña, B., Ramírez-Márquez, J.E., Monroy, R.: Fusing pattern discovery and visual analytics approaches in tweet propagation. *Inf. Fusion* **46**, 91–101. <https://doi.org/10.1016/j.inffus.2018.05.004>. <https://www.sciencedirect.com/science/article/pii/S1566253517307716> (2019)
123. Loyola-González, O., Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A., García-Borroto, M.: Effect of class imbalance on quality measures for contrast patterns: An experimental study. *Inf. Sci.* **374**, 179–192. <https://doi.org/10.1016/j.ins.2016.09.040>. <http://www.sciencedirect.com/science/article/pii/S0020025516309379> (2016)
124. Loyola-González, O., Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A., García-Borroto, M.: Study of the impact of resampling methods for contrast pattern based classifiers in imbalanced databases. *Neurocomputing* **175**(Part B), 935–947. <https://doi.org/10.1016/j.neucom.2015.04.120>. <http://www.sciencedirect.com/science/article/pii/S0925231215015908> (2016)
125. Loyola-González, O., Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A., Hernández-Tamayo, D., García-Borroto, M.: Detecting Pneumatic Failures on Temporary Immersion Bioreactors, vol. 9703, pp. 293–302. Springer International Publishing. [https://doi.org/10.1007/978-3-319-39393-3\\_29](https://doi.org/10.1007/978-3-319-39393-3_29) (2016)
126. Loyola-González, O., Medina-Pérez, M.A., Hernández-Tamayo, D., Monroy, R., Carrasco-Ochoa, J.A., García-Borroto, M.: A Pattern-Based Approach for Detecting Pneumatic Failures on Temporary Immersion Bioreactors. *Sensors* **19**(2). <https://doi.org/10.3390/s19020414>. <http://www.mdpi.com/1424-8220/19/2/414> (2019)
127. Loyola-González, O., Medina-Pérez, M.A., Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A., Monroy, R., García-borroto, M.: PBC4cip: A new contrast pattern-based classifier for class imbalance problems. *Knowl.-Based Syst.* **115**, 100–109 (2017). <https://doi.org/10.1016/j.knosys.2016.10.018>
128. Loyola-González, O., Monroy, R., Medina-Pérez, M.A., Cervantes, B., Grimaldo-Tijerina, J.E.: An Approach Based on Contrast Patterns for Bot Detection on Web Log Files. In: Batyrshin, I., Martínez-Villaseñor, M.d.L., Ponce Espinosa, H.E. (eds.) *Advances in Soft Computing*, pp. 276–285. Springer International Publishing (2018). [https://doi.org/10.1007/978-3-030-04491-6\\_21](https://doi.org/10.1007/978-3-030-04491-6_21)
129. Loyola-González, O., Monroy, R., Rodríguez, J., López-Cuevas, A., Israel, M.S.J.: Contrast pattern-based classification for bot detection on twitter. *IEEE Access* **7**, 45800–45817 (2019). <https://doi.org/10.1109/ACCESS.2019.2904220>
130. Loyola-González, O., Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A., García-Borroto, M.: Cost-sensitive pattern-based classification for class imbalance problems. *IEEE Access* **7**(1), 60411–60427 (2019). <https://doi.org/10.1109/ACCESS.2019.2913982>
131. Maggi, F.M., Mooij, A.J., van der Aalst, W.M.: User-Guided Discovery of Declarative Process Models. In: 2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), pp. 192–199. IEEE (2011)
132. Mao, S., Dong, G.: Discriminating Gene Transfer and Microarray Concordance Analysis. In: Dong, G., Bailey, J. (eds.) *Contrast Data Mining: Concepts, Algorithms, and Applications*, Data Mining and Knowledge Discovery Series, chap. 16, pp. 233–240. USA, Chapman & Hall/CRC (2012)
133. Martens, D., Baesens, B., Gestel, T.V., Vanthienen, J.: Comprehensible credit scoring models using rule extraction from support vector machines. *Eur. J. Oper. Res.* **183**(3), 1466–1476 (2007). <https://doi.org/10.1016/j.ejor.2006.04.051>
134. Martínez-Díaz, Y., Hernández, N., Biscay, R.J., Chang, L., Méndez-Vázquez, H., Sucar, L.E.: On fisher vector encoding of binary features for video face recognition. *J. Vis. Commun. Image Represent.* **51**, 155–161 (2018). <https://doi.org/10.1016/j.jvcir.2018.01.017>
135. Martínez-Díaz, Y., Méndez-Vázquez, H., López-Avila, L., Chang, L., Sucar, L.E., Tistarelli, M.: Toward More Realistic Face Recognition Evaluation Protocols for the Youtube Faces Database. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 526–5268. <https://doi.org/10.1109/CVPRW.2018.00082> (2018)
136. McGarry, K.: A survey of interestingness measures for knowledge discovery. *Kowl. Eng. Rev.* **1**(3), 39–61 (2005)
137. McGarry, K., Malone, J.: Analysis of rules discovered by the data mining process. In: Lotfi, A., Garibaldi, J. (eds.) *Applications and Science in Soft Computing*, *Advances in Soft Computing*, vol. 24, pp. 219–224. Springer, Berlin (2004). [https://doi.org/10.1007/978-3-540-45240-9\\_30](https://doi.org/10.1007/978-3-540-45240-9_30)
138. Mei, X., Lee, H.-L., Diao, K.-y., Huang, M., Lin, B., Liu, C., Xie, Z.: Artificial intelligence-enabled rapid diagnosis of patients with COVID-19. *Nature Medicine* **1–5** (2020)
139. Mugambi, E.M., Hunter, A., Oatley, G., Kennedy, L.: Polynomial-Fuzzy Decision Tree Structures for Classifying Medical Data. In: Coenen, F., Preece, A., Macintosh, A. (eds.) *Research and Development in Intelligent Systems XX*, pp. 155–167. Springer, London (2004)
140. Nauman, M., Ur Rehman, H., Politano, G., Benso, A.: Beyond homology transfer: Deep learning for automated

- annotation of proteins. *J. Grid Comput.* **17**(2), 225–237 (2019). <https://doi.org/10.1007/s10723-018-9450-6>
141. Neubarth, K., Conklin, D.: Contrast Pattern Mining in Folk Music Analysis, pp. 393–424. Springer International Publishing. [https://doi.org/10.1007/978-3-319-25931-4\\_15](https://doi.org/10.1007/978-3-319-25931-4_15) (2016)
  142. Noh, T., Walbert, T.: Chapter 6 - Brain Metastasis: Clinical Manifestations, Symptom Management, and Palliative Care. In: Schiff, D., van den Bent, M.J. (eds.) *Metastatic Disease of the Nervous System, Handbook of Clinical Neurology*, vol. 149, pp. 75–88. Elsevier (2018). <https://doi.org/10.1016/B978-0-12-811161-1.00006-2>
  143. Noor, K., Jan, T., Basher, M., Ali, A., Khalil, R.A., Zafar, M.H., Ashraf, M., Babar, M.I., Shah, S.W.: Performances enhancement of fingerprint recognition system using classifiers. *IEEE Access* **7**, 5760–5768 (2019). <https://doi.org/10.1109/ACCESS.2018.2879272>
  144. Novak, P.K., Lavrač, N., Webb, G.I.: Supervised Descriptive Rule Discovery: A Unifying Survey of Contrast Set, Emerging Pattern and Subgroup Mining. *J. Mach. Learn. Res.* **10**, 377–403 (2009)
  145. Oliveto, P.S., He, J., Yao, X.: Time complexity of evolutionary algorithms for combinatorial optimization: a decade of results. *Int. J. Autom. Comput.* **4**(3), 281–293 (2007). <https://doi.org/10.1007/s11633-007-0281-3>
  146. Onofri, L., Soda, P., Pechenizkiy, M., Iannello, G.: A survey on using domain and contextual knowledge for human activity recognition in video streams. *Expert Syst. Appl.* **63**(0), 97–111 (2016)
  147. Padmanabhan, B., Tuzhilin, A.: Knowledge refinement based on the discovery of unexpected patterns in data mining. *Decis. Support Syst.* **33**(3), 309–321 (2002)
  148. Park, J.H., Lee, H.G., Sohn, G.Y., Shin, J., Ryu, K.H.: Emerging Pattern Based Classification for Automated Non-Safe Power Line Detection. In: 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery, Vol. 1, pp. 169–173. <https://doi.org/10.1109/FSKD.2009.769> (2009)
  149. Pei, J., Han, J., Mao, R., et al.: Closet: an Efficient Algorithm for Mining Frequent Closed Itemsets. In: *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, Vol. 4, pp. 21–30 (2000)
  150. Peng, M., Ouyang, S., Zhu, J., Huang, J., Wang, H., Yong, J.: Emerging Topic Detection from Microblog Streams Based on Emerging Pattern Mining\*. In: 2018 IEEE 22Nd International Conference on Computer Supported Cooperative Work in Design ((CSCWD)), pp. 259–264. <https://doi.org/10.1109/CSCWD.2018.8465166> (2018)
  151. Petrović, S., Osborne, M., Lavrenko, V.: Streaming First Story Detection with Application to Twitter. In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 181–189. Association for Computational Linguistics (2010)
  152. Piao, M., Lee, H.G., Sohn, G.Y., Pok, G., Ryu, K.H.: Emerging Patterns Based Methodology for Prediction of Patients with Myocardial Ischemia. In: *Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, Vol. 1, pp. 174–178. <https://doi.org/10.1109/FSKD.2009.638> (2009)
  153. Pisoschi, A.M., Pisoschi, C.G.: Is open access the solution to increase the impact of scientific journals? *Scientometrics* **109**(2), 1075–1095 (2016). <https://doi.org/10.1007/s11192-016-2088-x>
  154. Poezevara, G., Lozano, S., Cuissart, B., Bureau, R., Bureau, P., Croixmarie, V., Vayer, P., Lepailleur, A.: A computational selection of metabolite biomarkers using emerging pattern mining: a case study in human hepatocellular carcinoma. *J. Proteome Res.* **16**(6), 2240–2249 (2017). <https://doi.org/10.1021/acs.jproteome.7b00054>
  155. Quinlan, J.R.: Induction of decision trees. *Mach. Learn.* **1**(1), 81–106 (1986). <https://doi.org/10.1007/BF00116251>
  156. Quinlan, J.R.: Generating Production Rules from Decision Trees. In: *Proceedings of the 10th International Joint Conference on Artificial Intelligence - Volume 1, IJCAI'87*, pp. 304–307. Morgan Kaufmann Publishers Inc., San Francisco. <http://dl.acm.org/citation.cfm?id=1625015.1625078> (1987)
  157. Quinlan, J.R.: *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc (1993)
  158. Rajaraman, A., Ullman, J.D.: *Mining of massive datasets*. Cambridge University Press, New York (2011)
  159. Rampini, A., Bordogna, G., Carrara, P., Pepe, M., Antoninetti, M., Mondini, A., Reichenbach, P.: Modelling Landslides' Susceptibility by Fuzzy Emerging Patterns, pp. 363–370. Springer, Berlin. [https://doi.org/10.1007/978-3-642-31325-7\\_48](https://doi.org/10.1007/978-3-642-31325-7_48) (2013)
  160. Reys, J.M., Aickelin, U., Hubbard, R.B.: Refining adverse drug reaction signals by incorporating interaction variables identified using emergent pattern mining. *Comput. Biol. Med.* **69**, 61–70 (2016)
  161. Rodriguez, J.J., Kuncheva, L.I., Alonso, C.J.: Rotation forest: a new classifier ensemble method. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(10), 1619–1630 (2006). <https://doi.org/10.1109/TPAMI.2006.211>
  162. Rodriguez-Ruiz, J., Monroy, R., Medina-Pérez, M., Loyola-González, O., Cervantes, B.: Cluster validation in clustering-based one-class classification. *Expert. Syst.* **36**(6), e12475 (2019). <https://doi.org/10.1111/exsy.12475>
  163. Saeed, K.E.K., Lee, H.G., Kim, W., Cha, E., Ryu, K.H.: Using Emerging Subsequence in Classifying Protein Structural Class. In: *Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, Vol. 1, pp. 349–353. <https://doi.org/10.1109/FSKD.2009.752> (2009)
  164. Saeed, Z., Abbasi, R.A., Maqbool, O., Sadaf, A., Razzak, I., Daud, A., Aljohani, N.R., Xu, G.: What's happening around the world? a survey and framework on event detection techniques on twitter. *J. Grid Comput.* **17**(2), 279–312 (2019). <https://doi.org/10.1007/s10723-019-09482-2>
  165. Schölkopf, B., Platt, J.C., Shawe-Taylor, J.C., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. *Neural Comput.* **13**(7), 1443–1471 (2001)
  166. Shah, A., Halim, Z.: On efficient mining of frequent itemsets from big uncertain databases. *J. Grid Comput.* **17**(4), 831–850 (2019). <https://doi.org/10.1007/s10723-018-9456-0>
  167. Sherhod, R., Judson, P.N., Hanser, T., Vessey, J.D., Webb, S.J., Gillet, V.J.: Emerging pattern mining to aid toxicological knowledge discovery. *J. Chem. Inf. Model.* **54**(7), 1864–1879. <https://doi.org/10.1021/ci5001828> (2014)

168. Sheskin, D.J.: Handbook of Parametric and Nonparametric Statistical Procedures, 4 edn. Chapman & Hall/CRC (2007)
169. Song, C., Liu, X., Ge, T., Ge, Y.: Top-k frequent items and item frequency tracking over sliding windows of any size. *Inf. Sci.* **475**, 100–120. <https://doi.org/10.1016/j.ins.2018.09.066>. <http://www.sciencedirect.com/science/article/pii/S0020025518307837> (2019)
170. Song, H.S., kyeong Kim, J., Kim, S.H.: Mining the change of customer behavior in an internet shopping mall. *Expert Syst. Appl.* **21**(3), 157–168. [https://doi.org/10.1016/S0957-4174\(01\)00037-9](https://doi.org/10.1016/S0957-4174(01)00037-9). <http://www.sciencedirect.com/science/article/pii/S0957417401000379> (2001)
171. Song, W., Liang, J.Z., Cao, X.L., Park, S.C.: An effective query recommendation approach using semantic strategies for intelligent information retrieval. *Expert Syst. Appl.* **41**(2), 366–372 (2014)
172. Szathmary, L., Napoli, A., Valtchev, P.: Towards Rare Itemset Mining. In: 2007. ICTAI 2007. 19Th IEEE International Conference On Tools with Artificial Intelligence, vol. 1, pp. 305–312 (2007)
173. Tabatabaee Malazi, H., Davari, M.: Combining emerging patterns with random forest for complex activity recognition in smart homes. *Appl. Intell.* **48**(2), 315–330 (2018). <https://doi.org/10.1007/s10489-017-0976-2>
174. Tan, P.N., Kumar, V., Srivastava, J.: Selecting the right objective measure for association analysis. *Inf. Syst.* **29**(4), 293–313. [https://doi.org/10.1016/S0306-4379\(03\)00072-3](https://doi.org/10.1016/S0306-4379(03)00072-3). <http://www.sciencedirect.com/science/article/pii/S0306437903000723>. *Knowledge Discovery and Data Mining (KDD 2002)* (2004)
175. Terlecki, P., Walczak, K.: Jumping emerging patterns with negation in transaction databases – classification and discovery. *Inf. Sci.* **177**(24), 5675–5690 (2007). <https://doi.org/10.1016/j.ins.2007.07.018>. <http://www.sciencedirect.com/science/article/pii/S0020025507003672>
176. Terlecki, P., Walczak, K.: Efficient Discovery of Top-K Minimal Jumping Emerging Patterns. In: Chan, C.C., Grzymala-Busse, J.W., Ziarko, W.P. (eds.) *Rough Sets and Current Trends in Computing*, pp. 438–447. Springer, Berlin (2008)
177. Terlecki, P., Walczak, K.: Relation between Jumping Emerging Patterns and Rough Set Theory. In: Dong, G., Bailey, J. (eds.) *Contrast Data Mining: Concepts, Algorithms, and Applications*, *Data Mining and Knowledge Discovery Series*, chap. 10, pp. 129–149. USA, Chapman & Hall/CRC (2012)
178. Teschner, M., Heidelberger, B., Müller, M., Pomerantes, D., Gross, M.H.: Optimized Spatial Hashing for Collision Detection of Deformable Objects. In: *Vmv*, Vol. 3, pp. 47–54 (2003)
179. Thabtah, F., Cowling, P., Peng, Y.: MCAR: Multi-Class Classification Based on Association Rule. In: *The 3Rd ACS/IEEE International Conference Oncomputer Systems and Applications*, 2005., pp. 33–. Cairo. <https://doi.org/10.1109/AICCSA.2005.1387030> (2005)
180. Thabtah, F.A., Cowling, P., Peng, Y.: MMAC: a New Multi-Class, Multi-Label Associative Classification Approach. In: *Fourth IEEE International Conference on Data Mining (ICDM'04)*, pp. 217–224, Brighton. <https://doi.org/10.1109/ICDM.2004.10117> (2004)
181. Thomas, S., Sarawagi, S.: Mining generalized association rules and sequential patterns using sql queries. In: *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, KDD'98*, pp. 344–348. AAAI Press. <http://dl.acm.org/citation.cfm?id=3000292.3000357> (1998)
182. Ting, D., Wei, S., Carin, L., Dzau, V., Wong, T.Y.: Digital technology and COVID-19. *Nature Medicine* **26**(4), 459–461 (2020)
183. van der Aalst, W., Weijters, T., Maruster, L.: Workflow mining: discovering process models from event logs. *IEEE Trans. Knowl. Data Eng.* **16**(9), 1128–1142 (2004). <https://doi.org/10.1109/TKDE.2004.47>
184. De la Vega-Doria, L., Carrasco-Ochoa, A., Ruiz-Shucloper, J.: Fuzzy Kora-W Algorithm. In: *6Th European Conf. on Intelligent Techniques and Soft Computer*, Germany, pp. 1190–1194 (1998)
185. Wang, J., Han, J., Pei, J.: Closet+: Searching for the best strategies for mining frequent closed itemsets. In: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '03*, pp. 236–245. ACM, New York. <https://doi.org/10.1145/956750.956779> (2003)
186. Wang, L., Wang, Y., Zhao, D.: Building Emerging Pattern (Ep) Random Forest for Recognition. In: *2010 IEEE International Conference on Image Processing*, pp. 1457–1460. <https://doi.org/10.1109/ICIP.2010.5653902> (2010)
187. Wang, L., Zhao, H., Dong, G., Li, J.: On the complexity of finding emerging patterns. In: *Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International*, vol. 2, pp. 126–129 (2004)
188. Wang, Y., Xu, T., Xue, S., Shen, Y.: D2p-Apriori: a Deep Parallel Frequent Itemset Mining Algorithm with Dynamic Queue. In: *Tenth International Conference on Advanced Computational Intelligence (ICACI 2018)*, pp. 649–654. <https://doi.org/10.1109/ICACI.2018.8377536> (2018)
189. Weng, C.H., Huang Tony, C.K.: Observation of sales trends by mining emerging patterns in dynamic markets. *Appl. Intell.* **48**(11), 4515–4529 (2018). <https://doi.org/10.1007/s10489-018-1231-1>
190. Wu, D., Wang, Z., Chen, Y., Zhao, H.: Mixed-kernel based weighted extreme learning machine for inertial sensor based human activity recognition with imbalanced dataset. *Neurocomputing* **190**(0), 35–49 (2016)
191. Wu, M.S.: Modeling query-document dependencies with topic language models for information retrieval. *Inform. Sci.* **312**(0), 1–12 (2015)
192. Wu, X., Zhu, X., Wu, G., Ding, W.: Data mining with big data. *IEEE Trans. Knowl. Data Eng.* **26**(1), 97–107 (2014). <https://doi.org/10.1109/TKDE.2013.109>
193. Wulandari, C.P., Ou-Yang, C., Wang, H.C.: Applying mutual information for discretization to support the discovery of rare-unusual association rule in cerebrovascular examination dataset. *Expert Syst. Appl.* **118**, 52–64. <https://doi.org/10.1016/j.eswa.2018.09.044>. <http://www.sciencedirect.com/science/article/pii/S0957417418306213> (2019)
194. Xue, J., Hu, C., Wang, K., Ma, R., Zou, J.: Metamorphic malware detection technology based on aggregating



- emerging patterns. In: Proceedings of the 2Nd International Conference on Interaction Sciences: Information Technology, Culture and Human, ICIS '09, pp. 1293–1296. ACM, New York. <https://doi.org/10.1145/1655925.1656162> (2009)
195. Yang, H., Fong, S.: Moderated Vfdt in Stream Mining Using Adaptive Tie Threshold and Incremental Pruning. In: Cuzzocrea, A., Dayal, U. (eds.) *Data Warehousing and Knowledge Discovery*, pp. 471–483. Springer, Berlin (2011)
  196. Yin, X., Han, J.: Cpar: Classification Based on Predictive Association Rules. In: 2003 SIAM International Conference on Data Mining, pp. 331–335, San Francisco. <https://doi.org/10.1137/1.9781611972733.40> (2003)
  197. Yu, K., Ding, W., Simovici, D.A., Wu, X.: Mining emerging patterns by streaming feature selection. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, pp. 60–68. ACM, New York (2012)
  198. Zaki, M.J.: Scalable algorithms for association mining. *IEEE Trans. Knowl. Data Eng.* **12**(3), 372–390 (2000)
  199. Zhang, C., Tian, P., Zhang, X., Liao, Q., Jiang, Z.L., Wang, X.: Hasheclat: an efficient frequent itemset algorithm. *International Journal of Machine Learning and Cybernetics*. <https://doi.org/10.1007/s13042-018-00918-x> (2019)
  200. Zhang, X., Dong, G.: Overview and Analysis of Contrast Pattern Based Classification. In: Dong, G., Bailey, J. (eds.) *Contrast Data Mining: Concepts, Algorithms, and Applications*, *Data Mining and Knowledge Discovery Series*, chap. 11, pp. 151–170. USA, Chapman & Hall/CRC (2012)
  201. Zhang, X., Dong, G., Ramamohanarao, K.: Information-Based Classification by Aggregating Emerging Patterns. In: Leung, K., Chan, L.W., Meng, H. (eds.) *Intelligent Data Engineering and Automated Learning — IDEAL 2000*. *Data Mining, Financial Engineering, and Intelligent Agents*, *Lecture Notes in Computer Science*, vol. 1983, pp. 48–53. Springer, Berlin (2000)
  202. Zhang, Y., He, D., Zhang, M., Choo, K.K.R.: A provable-secure and practical two-party distributed signing protocol for sm2 signature algorithm. *Front. Comput. Sci.* **14**(3), 143803. <https://doi.org/10.1007/s11704-018-8106-9> (2019)
  203. Zhao, X.M., Ngom, A., Hao, J.K.: Pattern recognition in bioinformatics. *Neurocomputing* **145**(0), 1–2 (2014)
  204. Zhao, Y., Li, Y., Wang, Z., Wang, G.: Finding Novel Diagnostic Gene Patterns Based on Interesting Non-Redundant Contrast Sequence Rules. In: *IEEE 11Th International Conference on Data Mining(ICDM)*, Vol. 00, pp. 972–981. <https://doi.org/10.1109/ICDM.2011.68> (2011)
  205. Zheng, Z., Wei, W., Liu, C., Cao, W., Cao, L., Bhatia, M.: An effective contrast sequential pattern mining approach to taxpayer behavior analysis. *World Wide Web* **19**(4), 633–651 (2016). <https://doi.org/10.1007/s11280-015-0350-4>
  206. Zhou, J., Foster, D.P., Stine, R.A., Ungar, L.H.: Streamwise feature selection. *J. Mach. Learn. Res.* **7**(Sep), 1861–1885 (2006)
  207. Zia, T., Razzaq, S.: Residual recurrent highway networks for learning deep sequence prediction models. *J. Grid Comput.* **18**(1), 169–176 (2020)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.