Expert Systems with Applications 40 (2013) 6823–6836

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Mining frequent patterns and association rules using similarities

Ansel Y. Rodríguez-González [a,*], José Fco. Martínez-Trinidad [a], Jesús A. Carrasco-Ochoa [a], José Ruiz-Shulcloper [b]

[a] Department of Computer Sciences, Institute of Astrophysics, Optics and Electronics (INAOE), Luis Enrique Erro 1, Tonantzintla, Puebla 72840, Mexico
[b] Advanced Technologies Applications Center (CENATAV), 7th Avenue 21812 between 218 and 222, Siboney Neighborhood, Playa, Havana City 12200, Cuba

ARTICLE INFO

Keywords:
Data mining
Frequent patterns
Association rules
Mixed data
Similarity functions
Downward closure property

ABSTRACT

Most of the current algorithms for mining association rules assume that two object subdescriptions are similar when they are exactly equal, but in many real world problems some other similarity functions are used. Commonly these algorithms are divided in two steps: Frequent pattern mining and generation of interesting association rules from frequent patterns. In this work, two algorithms for mining frequent similar patterns using similarity functions different from the equality are proposed. Additionally, the *GenRules* Algorithm is adapted to generate interesting association rules from frequent similar patterns. Experimental results show that our algorithms are more effective and obtain better quality patterns than the existing ones.

1. Introduction

Association Rule Mining (Agrawal, Imielinski, & Swami, 1993) is an important task for Knowledge Discovery in Data. It has been used for marketing (Wong, Zhou, Yang, & Yeung, 2005; Yunyan & Juan, 2010), crime analysis (Leong, Chan, Ng, & Shiu, 2008), intrusion detection (Ertoz et al., 2004), fraud detection (Sánchez, Vila, Cerda, & Serrano, 2009), disease diagnostic or analysis (Chaves, Ramrez, & Grriz, 2012, 2013; Dua, Singh, & Thompson, 2009; Nahar, Imam, Tickle, & Chen, 2013; Patil, Joshi, & Toshniwal, 2010), etc. Association rule mining consists in finding interesting "if-then" rules between feature value combinations in a dataset. An association rule $X \rightarrow Y$, where X and Y are combinations of feature values (patterns), means that if X appears in an object then Y also appears in the same object. Commonly an association rule is considered interesting if its frequency and confidence are not less than user-specified frequency and confidence thresholds. The frequency of a rule $X \rightarrow Y$ is the frequency of the pattern XY in a dataset Ω; and its confidence is the fraction of objects in the dataset in which if X appears then Y also appears. Association rule mining consists of two fundamental steps: (I) Search of frequent patterns (patterns with frequency not less than a frequency threshold); (II) Construction of association rules from frequent patterns.

The first step (also called frequent pattern mining) is very important by itself because regularities (patterns) in data are discovered, and depending on the application these patterns could represent user profiles, modus operandi, common syndromes, risk factors, etc., in areas such as Marketing, Bioinformatics, Medicine, Network security, and others (Alatas, Akin, & Karci, 2008; Hu, Sung, Xiong, & Fu, 2008; Kalpana & Nadarajan, 2008; LaRosa, Xiong, & Mandelberg, 2008; Lopez, Blanco, Garcia, Cano, & Marin, 2008; Xin & Zhi-Hong, 2010; Li & Deng, 2010). Moreover, frequent patterns play an essential role into some methods of other data mining tasks like classification (Hernández-León, Carrasco-Ochoa, Martínez-Trinidad, & Hernández-Palancar, 2012; Nahar et al., 2013; Nguyen, Vo, Hong, & Thanh, 2012) and clustering (Malik, Kender, Fradkin, & Moerchen, 2010).

What does a frequent pattern mean? It means that the same feature value combination occurs a certain number of times in the dataset. For example, given the dataset described by numerical and not numerical features (mixed data) shown in Table 1, assuming 0.6 as frequency threshold and 0.9 as confidence threshold, the only frequent combination of feature values is (*Married* = *No*) which appears 4 times in the 6 objects of the dataset; and there are no interesting association rules.

The concept of similarity or its opposite, the concept of dissimilarity (not necessarily a distance) is a natural tool commonly used in soft sciences to make decisions (Geology Gómez, Rodríguez, Valladares, & Ruiz-Shulcloper, 1994, Medicine Ortiz-Posadas, Vega-Alvarado, & Toni, 2009, Sociology Ruiz-Shulcloper & Fuentes-Rodríguez, 1981, etc.). If a similarity function different from the equality is employed, a *frequent pattern*, also called *frequent similar pattern* (Rodríguez-González, Martínez-Trinidad, Carrasco-Ochoa, & Ruiz-Shulcloper, 2008), is a combination of feature values of the study objects, such that, the similarity accumulation of its

* Corresponding author. Tel.: +52 222 2613016.
E-mail addresses: ansel@ccc.inaoep.mx (A.Y. Rodríguez-González), fmartine@ccc.inaoep.mx (J.F. Martínez-Trinidad), ariel@ccc.inaoep.mx (J.A. Carrasco-Ochoa), jshulcloper@cenatav.co.cu (J. Ruiz-Shulcloper).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.06.041

Table 1
Example of a mixed dataset.

Ω	*Age*	*Car*	*Married*
O_1	23	*Compact*	*No*
O_2	25	*Big*	*No*
O_3	25	*Medium*	*No*
O_4	29	*Medium*	*No*
O_5	34	*Big*	*Yes*
O_6	38	*Fancy*	*Yes*

similar patterns is not less than an user-specified frequency threshold.

Considering the last frequent similar pattern definition and supposing that: (I) two ages are similar if the absolute value of their difference is at most 5 years; and (II) compact cars are similar to medium cars, medium cars are similar to big cars; big cars are similar to fancy cars; then the frequent similar patterns and the interesting association rules mined from Table 1 as well as their frequency and confidence values would be those shown in Table 2.

As it can be noticed, the use of a similarity function different from the equality (between feature values and object descriptions) produces frequent patterns and interesting association rules which are hidden for algorithms that use the equality as similarity function.

Preliminar results of this paper were presented in Rodríguez-González et al. (2008). In the present work, we focused on association rule mining using similarity functions on mixed data. This process is divided in two steps: (I) frequent similar pattern mining; (II) generation of interesting association rules from frequent similar patterns. For the first step we propose two algorithms: One for similarity functions that hold the *f-downward closure property* and other for similarity functions that do not hold this property. For the second step we propose an adaptation of the *GenRules* Algorithm (Agrawal & Srikant, 1994). The main differences of this paper with the conference paper are that here (I) we formalize and proof the properties in which our frequent similar pattern mining algorithms are based, (II) we evaluate the quality of the mined patterns, and (III) we propose an adaptation of the *GenRules* Algorithm (Agrawal & Srikant, 1994) for computing association rules from frequent similar patterns.

It is important to highlight that in Rodríguez-González, Martínez-Trinidad, Carrasco-Ochoa, and Ruiz-Shulcloper (2011) a frequent similar pattern mining algorithm (called *RP-Miner*) for Boolean similarity functions that does not meet the Downward Closure property was proposed. Even though the experimental results reported in Rodríguez-González et al. (2011) show that *RP-Miner* is more efficient than the algorithm proposed in this paper for similarity functions that does not fulfill the Downward Closure property (*STreeNDC-Miner*) and more effective than the algorithm proposed in this paper for similarity functions that fulfill the Downward Closure property (*STreeDC-Miner*). In Rodríguez-González et al. (2011) it is also shown that in those problems where we know that the similarity function fulfills the Downward Closure property, *STreeDC-Miner* is faster that *RP-Miner*. While in those problems where we know that the similarity function does not fulfill the Downward Closure property *STreeNDC-Miner* finds all the patterns while *RP-Miner* finds only a subset. Therefore the algorithms proposed in this paper constitute an alternative to those cases where *RP-Miner* does not provide good results. Therefore, the results presented in this paper complete the study of algorithms for mining frequent similar patterns on mixed data using Boolean similarity functions different from the equality.

The outline of this paper is as follows. In Section 2 related works are reviewed. Section 3 provides basic concepts. Section 4 describes the proposed similar frequent pattern mining algorithms. In Section 5 we adapt the *GenRules* Algorithm for computing association rules from frequent similar patterns. Finally, in Sections 6 and 7 experimental results and conclusions are respectively exposed.

Table 2
Frequent similar patterns and interesting association rules.

Frequent similar patterns	Frequency
(*Age* = 25)	0.66
(*Age* = 29)	0.66
(*Car* = *Medium*)	0.83
(*Car* = *Big*)	0.83
(*Married* = *No*)	0.66
(*Age* = 25, *Car* = *Medium*)	0.66
(*Age* = 29, *Car* = *Medium*)	0.66
(*Age* = 25, *Married* = *No*)	0.66
(*Car* = *Medium*, *Married* = *No*)	0.66
(*Age* = 25, *Car* = *Medium*, *Married* = *No*)	0.66
Interesting association rules	**Confidence**
(*Age* = 25) → (*Car* = *Medium*)	1
(*Age* = 29) → (*Car* = *Medium*)	1
(*Age* = 25) → (*Married* = *No*)	1
(*Married* = *No*) → (*Age* = 25)	1
(*Car* = *Medium*) → (*Married* = *No*)	0.8
(*Age* = 25) → (*Car* = *Medium*, *Married* = *No*)	1
(*Age* = 25, *Car* = *Medium*) → (*Married* = *No*)	1
(*Age* = 25, *Married* = *No*) → (*Car* = *Medium*)	1
(*Car* = *Medium*, *Married* = *No*) → (*Age* = 25)	1
(*Married* = *No*) → (*Age* = 25, *Car* = *Medium*)	1

2. Related work

*ObjectMiner*Dánger, Ruiz-Shulcloper, and Berlanga (2004) was the first algorithm that used similarity functions for mining frequent patterns. In order to allow pruning the search space of frequent similar patterns, this algorithm was designed for similarity functions that hold: if two objects are not similar with respect to a feature set S then they are not similar with respect to any superset of S. *ObjectMiner* was inspired in the *Apriori* Algorithm (Agrawal & Srikant, 1994). It works following a breadth first search strategy: first, for each feature, all frequent similar values (frequent similar subdescriptions with only one feature), are determined. Combining frequent similar patterns of length 1, candidates to frequent similar patterns of length 2 are obtained. Afterwards, for each pair (P_i,P_j) of frequent similar subdescriptions with $k-1$ features, found in the iteration $k-1$, if P_i and P_j have a common subdescription with $k-2$ features, they are combined in order to create a new candidate subdescription P^* with k features (see Fig. 1). In this step, for each pair (P_i,P_j) of frequent similar subdescriptions with $k-1$ features and a common subdescription with $k-2$ features, the next process is done:

- The combination P^* is obtained from P_i and P_j.
- A set of candidates similar to P^* is obtained intersecting the set of subdescriptions similar to P_i and the set of subdescriptions similar to P_j.
- From the set of candidates, a set of subdescriptions similar to P^* is obtained.
- The frequency of P^* is computed in the dataset.

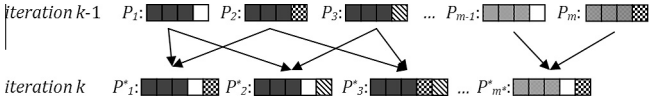

Fig. 1. Combination of subdescriptions with $k-1$ features into subdescriptions with k features.

- If the frequency of P^* is greater than or equal to *minFreq* then P^* is a frequent similar subdescription.

This process finishes when an iteration does not produce any frequent similar subdescription with k features.

The main weakness of *ObjectMiner* is that although descriptions or subdescriptions of objects are usually repeated in the datasets, it does not use this fact in order to reduce the number of operations on subsequent steps. For this reason, the similarity between repetitions of the same subdescription are computed, causing an additional and unnecessary computational effort. Also, storing the set of similar subdescriptions (including its repetitions) for each frequent subdescription affects the performance of *ObjectMiner* when it processes datasets with many objects.

In Dánger et al. (2004) it was shown that the use of similarity functions different from the equality for computing the frequency of feature value combinations allows to find frequent patterns hidden when the equality is used as similarity function. Also, in Dánger et al. (2004), interesting association rules obtained from frequent similar patterns, are shown. However, no algorithm for generating interesting association rules is presented in Dánger et al. (2004).

For generating interesting association rules from frequent patterns in Agrawal and Srikant (1994) the *GenRules* algorithm was proposed. The *GenRules* algorithm consists in generating for each frequent pattern all possible rules by separating the features in the frequent pattern in two disjoint subsets, and verifying if the confidence of the rules are greater than or equal to a specified minimum confidence threshold.

In Agrawal and Srikant (1994) a fast algorithm for generating interesting association rules from frequent patterns is described. This algorithm is based on the following property. For each frequent pattern, if the confidence of the rule $X \to Y$ obtained by separating its features in two disjoint subsets is less than a specified minimum confidence threshold, then the confidence of all rule $X - Z \to YZ$, where Z is a subpattern of X, is also less than the specified minimum threshold. As a consequence of this property, if a rule $X \to Y$ is not interesting then all rules $X - Z \to YZ$, where Z is a subpattern of X, are also not interesting, and it is not necessary verifying their confidence. In the fast algorithm, for each frequent pattern, first all rules with only one feature in the consequent and the remaining features in the antecedent, are generated. Later, for each rule, the features of the antecedent are recursively moved to the consequent and thus new rules are generated, until the confidence of the rule become less than the minimum confidence threshold.

Other algorithms for generating interesting association rules from frequent patterns have been reported in the literature, however they are designed for specific kinds of associations rules or domains (Ayubi, Muyeba, Baraani, & Keane, 2009; Chen & Wei, 2000; Choi & Hyun, 2010; Li-Min, Shu-Jing, & Don-Lin, 2010; Ya-Han & Yen-Liang, 2006).

3. Basic concepts

Let $\Omega = \{O_1, O_2, \ldots, O_n\}$ be a dataset. Each object O is described by a set of features $R = \{r_1, r_2, \ldots, r_m\}$ and represented as a tuple $(v_1, v_2, \ldots, v_m)$ where $v_i \in D_i$ (D_i is the domain of the feature r_i) $(1 \leqslant i \leqslant m)$. A *subdescription* of an object O for a subset of features $S \subseteq R$ denoted as $I_S(O)$, is the description of O in terms of the features in S; $O[r]$ denotes the value of O in the feature $r \in R$; and $f_S(O,O')$ denotes the similarity between O and O' using their subdescriptions $I_S(O)$ and $I_S(O')$ respectively (Martínez-Trinidad, Ruiz-Shulcloper, & Lazo-Cortés, 2000). Given two subdescriptions $I_S(O), I_S(O')$, with $O, O' \in \Omega, f_S(O,O') = 1$ means that O is similar to O' with respect to S and $f_S(O,O') = 0$ means that O is not similar to O' with respect to S. Two examples of similarity functions that depend on a feature set S are:

$$f_S(O,O') = \begin{cases} 1 & \text{if } \forall r \in S, C_r(O[r], O'[r]) = 1 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$f_S(O,O') = \begin{cases} 1 & \text{if } \frac{|\{r \in S | C_r(O[r], O'[r]) = 1\}|}{|S|} \geqslant \alpha \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where $C_r : D_r \times D_r \to \{0,1\}$ is a comparison function between values of the feature r, and $\alpha \in [0,1]$. Two examples of comparison functions are:

$$C_r(x,y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$$C_r(x,y) = \begin{cases} 1 & \text{if } |x - y| \leqslant \varepsilon \text{ where } \varepsilon \in \mathbb{R}^+ \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Let $I_S(O)$ be a subdescription, $O \in \Omega, S \subseteq R$, $S \neq \emptyset$ and f be a similarity function that depends on a feature set S; then the *frequency* of $I_S(O)$ in Ω for f is defined as:

$$f_S - freq(O) = \frac{|\{O' \in \Omega \ | f_S(O,O') = 1\}|}{|\Omega|} \quad (5)$$

We say that $I_S(O)$ is a *f-frequent subdescription* in Ω, also called *frequent similar pattern*, if its frequency is not less than a frequency threshold *minFreq*.

In this context, an association rule is an expression $X \to Y$ where $X = I_{S_1}(O)$ and $Y = I_{S_2}(O)$, such that $O \in \Omega$, $S_1, S_2 \subset R, S_1 \neq \emptyset, S_2 \neq \emptyset$ and $S_1 \cap S_2 = \emptyset$. Let $X \to Y$ be an association rule, $X = I_{S_1}(O)$ and $Y = I_{S_2}(O)$ and let f be a similarity function that depends on a feature set S; the confidence of $X \to Y$ in Ω for f is defined as:

$$f_S - conf(I_{S_1}(O) \to I_{S_2}(O)) = \frac{f_{S_1 \cup S_2} - freq(O)}{f_{S_1} - freq(O)} \quad (6)$$

We say that $X \to Y$, where $X = I_{S_1}(O)$ and $Y = I_{S_2}(O), S_1 \neq \emptyset, S_2 \neq \emptyset, S_1 \cap S_2 = \emptyset$, is an interesting association rule if its confidence is not less than a confidence threshold *minConf*, and $I_{S_1 \cup S_2}(O)$ is a frequent similar pattern.

Given a dataset of objects Ω, described by a set of features R, a similarity function f that depends on a feature set S, and a frequency threshold *minFreq*, the frequent similar pattern mining problem consists in finding all frequent similar patterns in Ω. If a confidence threshold *minConf* is also given, the association rule mining problem using a similarity function consists in finding all interesting association rules from the frequent similar patterns in Ω.

4. Frequent similar pattern mining

The downward closure property has been used in frequent itemset mining for pruning the search space (Agrawal & Srikant, 1994). This property ensures that all supersets of a non-frequent itemset are also non-frequent itemsets. An analogous downward closure property, for mining frequent similar patterns, can be expressed as follows: all superdescriptions of a non-f-frequent subdescription are also non-f-frequent subdescriptions. We call f-downward closure property to this property.

Given a dataset of objects Ω and a similarity function f that depends on a feature set S:

Definition 1 (*Monotonic similarity function*). f is non increasing monotonic iff $\forall O, O', S_1, S_2$; $O, O' \in \Omega, \emptyset \neq S_1 \subseteq S_2 \subseteq R [f_{S_1}(O,O') = 0] \Rightarrow [f_{S_2}(O,O') = 0]$.

The similarity function (1) is an example of non increasing monotonic similarity function.

Property 1 (Monotony of the frequency). *f fulfills the monotony of the frequency iff* $\forall O,S_1,S_2;\ O \in \Omega[\emptyset \neq S_1 \subseteq S_2 \subseteq R] \Rightarrow [f_{S_1} - freq(O) \geqslant f_{S_2} - freq(O)]$.

Property 2 (*f*-downward closure). *f fulfills the f-downward closure iff* $\forall O,S_1,S_2;\ O \in \Omega;\ \emptyset \neq S_1 \subseteq S_2 \subseteq R\ \ [f_{S_1} - freq(O) < minFreq] \Rightarrow [f_{S_2} - freq(O) < minFreq]$.

However, the *f*-downward closure property, unlike the downward closure property for frequent itemset mining, is not always true, because its fulfillment depends on the monotony of the frequency, which also depends on the monotony of the similarity function. These dependencies can be expressed as:

Proposition 1. *If f is a non increasing monotonic similarity function, then f fulfills the monotony of the frequency.*

Proof. If f is a non increasing monotonic similarity function then $\forall O,O',S_1,S_2;\ \ O,O' \in \Omega\ \ \emptyset \neq S_1 \subseteq S_2 \subseteq R\ \ [f_{S_1}(O,O') = 0] \Rightarrow [f_{S_2}(O,O') = 0]$. Therefore, $\forall O,O',S_1,S_2;\ O,O' \in \Omega$:

$$[\emptyset \neq S_1 \subseteq S_2 \subseteq R] \Rightarrow [[f_{S_2}(O,O') = 1] \Rightarrow [f_{S_1}(O,O') = 1]] \tag{7}$$

then, $\forall O,S_1,S_2;\ O \in \Omega$:

$$[\emptyset \neq S_1 \subseteq S_2 \subseteq R] \Rightarrow [\{O' \in \Omega\ |f_{S_1}(O,O') = 1\} \supseteq \{O' \in \Omega\ |f_{S_2}(O,O') = 1\}] \tag{8}$$

and

$$[\emptyset \neq S_1 \subseteq S_2 \subseteq R] \Rightarrow \left[\frac{|\{O' \in \Omega\,|f_{S_1}(O,O') = 1\}|}{|\Omega|} \geqslant \frac{|\{O' \in \Omega\,|f_{S_2}(O,O') = 1\}|}{|\Omega|}\right] \tag{9}$$

Thus, $\forall O,S_1,S_2;O \in \Omega[\emptyset \neq S_1 \subseteq S_2 \subseteq R] \Rightarrow [f_{S_1} - freq(O) \geqslant f_{S_2} - freq(O)]$ □

Proposition 2. *If f fulfills the monotony of the frequency, then f fulfills the f-downward closure.*

Proof. If f fulfills the Monotony of the frequency then $\forall O,S_1,\ S_2;\ O \in \Omega[\emptyset \neq S_1 \subseteq S_2 \subseteq R] \Rightarrow [f_{S_1} - freq(O) \geqslant f_{S_2} - freq(O)]$. Therefore, $\forall O,S_1,S_2;\ O \in \Omega$:

$$[\emptyset \neq S_1 \subseteq S_2 \subseteq R] \Rightarrow [[minFreq > f_{S_1} - freq(O)] \Longleftrightarrow [minFreq > f_{S_2} - freq(O)]] \tag{10}$$

Thus, $\forall O,S_1,S_2;\ \ O \in \Omega;\ \ \emptyset \neq S_1 \subseteq S_2 \subseteq R[f_{S_1} - freq(O) < minFreq] \Rightarrow [f_{S_2} - freq(O) < minFreq]$. □

Proposition 3. *If f is a non increasing monotonic similarity function, then f satisfies the f-downward closure.*

Proof. Based on Propositions 1 and 2 the proof is immediate. □

In frequent itemset mining, all supersets of a non-frequent itemset are pruned. However, in frequent similar pattern mining although the superdescriptions of the non-frequent similar patters are non-frequent similar patters, some superdescriptions of the non-frequent similar patters can be similar to the frequent similar patterns. Consequently, these superdescriptions of the non-frequent similar patters contribute to the frequency of the frequent similar patterns. Therefore these superdescriptions can not be pruned.

Then non-prunable and prunable patterns are defined.

Definition 2 (*f-non-prunable pattern*). Given a similarity function f, that depends on a feature set S, a subdescription $I_S(O)$ is a *f-non-prunable pattern* if $I_S(O)$ is a f-frequent subdescription or it contributes to the frequency of a f-frequent subdescription $I_S(O')$ (i.e, $f_S(O',O) = 1$); $I_S(O') \neq I_S(O)$.

In contraposition, a subdescription $I_S(O)$ is a *f-prunable pattern* if $f_S - freq(O) < minFreq$ and $\forall O';\ O' \in \Omega; I_S(O') \neq I_S(O)\ [f_S - freq(O') \geqslant minFreq] \Rightarrow [f_S(O',O) = 0]$.

Proposition 4. *If f is a non increasing monotonic similarity function and a subdescription $I_S(O)$ is a f-prunable pattern, then all superdescriptions $I_{S'}(O)$ are f-prunable patterns.*

Proof. if $I_S(O)$ is a f-prunable pattern, then

$$f_S - freq(O) < minFreq \tag{11}$$

and

$$\begin{aligned} &\forall O'; O' \in \Omega; I_S(O') \neq I_S(O) \\ &[f_S - freq(O') \geqslant minFreq] \Rightarrow [f_S(O',O) = 0] \end{aligned} \tag{12}$$

By the hypothesis and the Proposition 3, f fulfills the f-downward closure. So, based in (11) we have

$$\forall S'; S \subseteq S' \subseteq R[f_S - freq(O) < minFreq] \Rightarrow [f_{S'} - freq(O) < minFreq] \tag{13}$$

From the non increasing monotony of the similarity function f and (12):

$$\begin{aligned} &\forall O', S'; O' \in \Omega; S \subseteq S' \subseteq R; I_S(O') \neq I_S(O) \\ &[f_S - freq(O') \geqslant minFreq] \Rightarrow [f_{S'}(O',O) = f_S(O',O) = 0] \end{aligned} \tag{14}$$

In addition, it is obvious that:

$$\begin{aligned} &\forall O', S'; O' \in \Omega; S \subseteq S' \subseteq R \\ &[I_S(O') \neq I_S(O)] \Rightarrow [I_{S'}(O') \neq I_{S'}(O)] \end{aligned} \tag{15}$$

Therefore, from (14) and (15):

$$\begin{aligned} &\forall O'; O' \in \Omega; I_{S'}(O') \neq I_{S'}(O) \\ &[f_{S'} - freq(O') \geqslant minFreq] \Rightarrow [f_{S'}(O',O) = 0] \end{aligned} \tag{16}$$

Finally, from (13) and (16) we obtain that $\forall S';S \subseteq S' \subseteq R$:

$$f_{S'} - freq(O) < minFreq$$

and

$$\begin{aligned} &\forall O'; O' \in \Omega; I_{S'}(O') \neq I_{S'}(O) \\ &[f_{S'} - freq(O') \geqslant minFreq] \Rightarrow [f_{S'}(O',O) = 0] \end{aligned}$$

That is, all superdesdriptions of a f-prunable pattern $I_S(O)$ are f-prunable patterns. □

Notice that, f-prunable patterns neither are frequent similar patterns, nor contribute to the frequency of any frequent similar pattern. Additionally, from the Proposition 4, all superdesdriptions of a f-prunable pattern are not frequent similar patterns. Therefore, all f-prunable patterns can be pruned without losing frequent similar patterns.

The universe of all similarity functions can be divided in two subsets; the subset of similarity functions that hold the f-downward closure property, and the subset of similarity functions that do not hold it. In the following subsections we propose an algorithm for mining frequent similar patterns for each one of these subsets.

4.1. STreeDC-Miner algorithm

In this section, we introduce the *STreeDC-Miner* algorithm for mining frequent similar patterns when the similarity function holds the f-downward closure property.

The *f*-downward closure property ensures that superdescriptions of non-frequent subdescriptions are non-frequent subdescriptions, and we use this property to prune the feature combination space during the search of frequent similar patterns.

Let $\prec$ be a linear order in R. We can obtain all possible expansion of $\emptyset$, by means of consecutive direct expansions, where a *direct expansion* of $S \subseteq R$ is defined as $\hat{S} = S \cup \{r\}, r \in R - S, \forall r' \in S, r' \prec r$ such that the number of *f*-frequent subdescriptions with respect to S is greater than zero or $S = \emptyset$. For each expansion $\hat{S}$, the *f*-frequent subdescriptions regarding $\hat{S}$ can be obtained. The proposed algorithm, named *STreeDC-Miner*, follows this expansion process for discovering all frequent similar patterns.

In order to search the *f*-frequent subdescriptions from each expansion $\hat{S}$, we build for each $\hat{S}$ a structure called $STree_{\hat{S}}$. This structure is a tree where each path from the root to a leaf represents a subdescription P. Each leaf contains: *P.objs*, the list of objects having a subdescription equal to P; *P.similars*, the list of subdescriptions which are similar to P; and $P.c_{\approx}$, the number of occurrences of subdescriptions in the dataset, which are similar but not equal to P. In Fig. 2(c) an example of an *STree* is shown.

We distinguish two cases for building an $STree_{\hat{S}}$ structure:

- $|\hat{S}| = 1$ then all the objects in Ω taking into account only the feature in $\hat{S}$, are added to $STree_{\hat{S}}$. After that, the similarities between all subdescriptions in $STree_{\hat{S}}$ are computed, and for each subdescription $\hat{P}$, the list $\hat{P}.similars$ is updated (Algorithm 1, lines 4–10). Let q be the number of subdescriptions in $STree_{\hat{S}}$ then the number of similarity function evaluations is $\frac{q*(q-1)}{2}$. If $q < |\Omega|$(it means some subdescriptions in $STree_{\hat{S}}$ are repeated in Ω) *STreeDC-Miner* avoids unnecessary similarity function evaluations compared with the number of similarity function evaluations between all subdescriptions respect to $\hat{S}$ in Ω $\left(\frac{|\Omega|*(|\Omega|-1)}{2}\right)$. For example, in a dataset with 10 000 objects, if there are only 100 different subdescriptions respect to a feature subset, we would compute 9 900 similarity values, instead of 99 990 000.
- If $|\hat{S}| > 1$ then, since $\hat{S} = S \cup \{r\}$ we will build $STree_{\hat{S}}$ from $STree_S$ as follows; for each *f*-non-prunable pattern P in $STree_S$, all objects in *P.objs* are added to $STree_{\hat{S}}$, but including the values of r. After that, only the similarity between all subdescriptions

Algorithm 1: STreeDC-Miner($STree_{\hat{S}}$,$\hat{\hat{S}}$,Ω, f, $minFreq$)

Input: $STree_S$ - Data Structure, $\hat{S}$ - Feature set, Ω - Dataset, f - Similarity Function, $minFreq$ - Minimum Frequency Threshold
Output: F - Frequent Similar Pattern Set

```
1  if Ŝ ≠ ∅ then
2      STree_Ŝ ← empty STree structure
3      if |Ŝ| = 1 then
4          foreach O ∈ Ω do
5              if ¬STree_Ŝ.contain(I_Ŝ(O)) then
6                  STree_Ŝ.add(O)
7              STree_Ŝ.I_Ŝ(O).objs ← STree_Ŝ.I_Ŝ(O).objs ∪ {O}
8          foreach P̂, P̂′ ∈ STree_Ŝ such that P̂′ ≠ P̂ do
9              if f_Ŝ(P̂, P̂′) = 1 then
10                 P̂′.similars ← P̂′.similars ∪ {P̂}
11     else
12         foreach P ∈ STree_S do
13             foreach O ∈ P.objs do
14                 if ¬STree_Ŝ.contain(I_Ŝ(O)) then
15                     STree_Ŝ.add(O)
16                 STree_Ŝ.I_Ŝ(O).objs ← STree_Ŝ.I_Ŝ(O).objs ∪ {O}
17         foreach P̂, P̂′ ∈ STree_Ŝ such that I_S(P̂) ∈ I_S(P̂′).similars do
18             if f_Ŝ(P̂, P̂′) = 1 then
19                 P̂′.similars ← P̂′.similars ∪ {P̂}
20     foreach P̂, P̂′ ∈ STree_Ŝ such that I_Ŝ(P̂′) ∈ I_Ŝ(P̂).similars do
21         P̂′.c̃ ← P̂′.c̃ + |P̂.objs|
22     F ← {P ∈ STree_Ŝ | P.c̃ + |P.objs| ≥ minFreq}
23     STree_Ŝ.removefPrunablePatterns()
24 if Ŝ = ∅ or F ≠ ∅ then
25     foreach direct expansion Ŝ̂ of Ŝ do
26         F ← F ∪ STreeDC-Miner(STree_Ŝ,Ŝ̂,Ω, f, minFreq)
```

in $STree_{\check{S}}$, such that their similarities respect to S are different from zero are computed (this is an implication of the f-downward closure property that reduces the number of similarity function evaluations), and for each subdescription $\hat{P}$ in $STree_{\check{S}}$, the list $\hat{P}.similars$ is updated (Algorithm 1, lines 12–19).

Finally, in both cases, for each subdescription $\hat{P}$ in $STree_{x\check{S}}$, $\hat{P}.\tilde{c}$ is computed, the f-frequent similar patterns are obtained and the f-Prunable Patterns are removed from $STree_{\check{S}}$ (Algorithm 1, lines 20–23).

4.2. STreeNDC-Miner algorithm

Since *STreeDC-Miner* finds all frequent similar patterns only if the similarity function fulfills the f-downward closure property, in this section, we introduce the *STreeNDC-Miner* algorithm for mining all frequent similar patterns when the similarity function does not hold the f-downward closure property.

If f does not hold the f-downward closure property, then this property can not be used for pruning the feature combination space during the search of frequent similar patterns. Thus, an alternative is to search the f-frequent subdescriptions for all $S \subseteq R$, $S \neq \emptyset$, which could be very computational expensive. However, the computational effort for searching all frequent similar patterns could be reduced using a top down strategy and the *STree* data structure.

In order to search the frequent similar patterns in Ω, we will obtain all possible reductions of R, by means of consecutive direct reductions, where a *direct reduction* of $S \subseteq R$, $|S| > 1$ is defined as $\check{S} = S - \{r\}, r \in S$ and $\forall\ r' \in (R - S), r' \prec r$. For each reduction $\check{S}$, the f_S-frequent subdescriptions regarding $\check{S}$ are obtained. The proposed algorithm, called *STreeNDC-Miner*, follows this reduction process for discovering all frequent similar patterns.

To obtain all f-frequent subdescriptions for each subset of features $\check{S}$, *STree* structure is used. Thus, unnecessary similarity function evaluations are avoided. In the *STree* structure used by *STreeNDC-Miner* for each leaf the list of objects having a subdescription equal to P, is substituted by $(\bar{c})$ the number of objects having a subdescription equal to P.

In *STreeNDC-Miner*, like in *STreeDC-Miner*, we distinguish two cases for building an $STree_{\check{S}}$ structure:

- $\check{S} = R$. All objects in Ω are added to $STree_{\check{S}}$ (Algorithm 2, lines 3–6).
- $\check{S} \subset R, \check{S} = S - \{r\}$. All subdescriptions in $STree_S$, are added to $STree_{\check{S}}$ discarding the values of r, as well as the number of their repetitions in Ω (Algorithm 2, lines 8–12).

Finally, in both cases, the similarities between all subdescriptions in $STree_{\check{S}}$ are computed, for each subdescription P in $STree_{\check{S}}$, the list $P.similars$ is updated and the f-frequent similar patterns are obtained (Algorithm 2, lines 13–18).

5. Generating interesting association rules

The use of similarity functions different from the equality for computing the frequency of the subdescriptions allows to find interesting association rules hidden when the equality is used as similarity function. Additionally, when the equality is used instead a similarity function different from the equality, false association rules could be generated. A false association rule is a rule that using the genuine similarity function to compute its frequency and its confidence, results non interesting. These affirmations are formalized and proof on the following proposition.

Algorithm 2: STreeNDC-Miner($STree_S$,$\check{S}$,Ω, f, $minFreq$)

Input: $STree_S$ - Data Structure, $\check{S}$ - Feature set, Ω - Dataset, f - Similarity Function, $minFreq$ - Minimum Frequency Threshold
Output: F - Frequent Similar Pattern Set

1 $STree_{\check{S}} \leftarrow$ empty $STree$ structure
2 **if** $\check{S} = R$ **then**
3 **foreach** $O \in \Omega$ **do**
4 **if** $\neg STree_{\check{S}}.contain(I_{\check{S}}(O))$ **then**
5 $STree_{\check{S}}.add(O)$
6 $STree_{\check{S}}.I_{\check{S}}(O).\bar{\bar{c}} \leftarrow STree_{\check{S}}.I_{\check{S}}(O).\bar{\bar{c}} + 1$
7 **else**
8 **foreach** $P \in STree_S$ **do**
9 **foreach** $O \in P.objs$ **do**
10 **if** $\neg STree_{\check{S}}.contain(I_{\check{S}}(O))$ **then**
11 $STree_{\check{S}}.add(O)$
12 $STree_{\check{S}}.I_{\check{S}}(O).\bar{\bar{c}} \leftarrow STree_{\check{S}}.I_{\check{S}}(O).\bar{\bar{c}} + STree_S.I_S(O).\bar{\bar{c}}$
13 **foreach** $P, P' \in STree_{\check{S}}$ *such that* $P' \neq P$ **do**
14 **if** $f_{\check{S}}(P, P') = 1$ **then**
15 $P'.similars \leftarrow P'.similars \cup \{P\}$
16 **foreach** $P, P' \in STree_{\check{S}}$ *such that* $P' \in P.similars$ **do**
17 $P'.\tilde{c} \leftarrow P'.\tilde{c} + P.\bar{\bar{c}}$
18 $F \leftarrow \{P \in STree_{\check{S}} \mid P.\tilde{c} + P.\bar{\bar{c}} \geq minFreq\}$
19 **foreach** *direct reduction* $\check{\check{S}}$ *of* $\check{S}$ **do**
20 $F \leftarrow F \cup$ STreeNDC-Miner($STree_{\check{S}}$,$\check{\check{S}}$,Ω, f, $minFreq$)

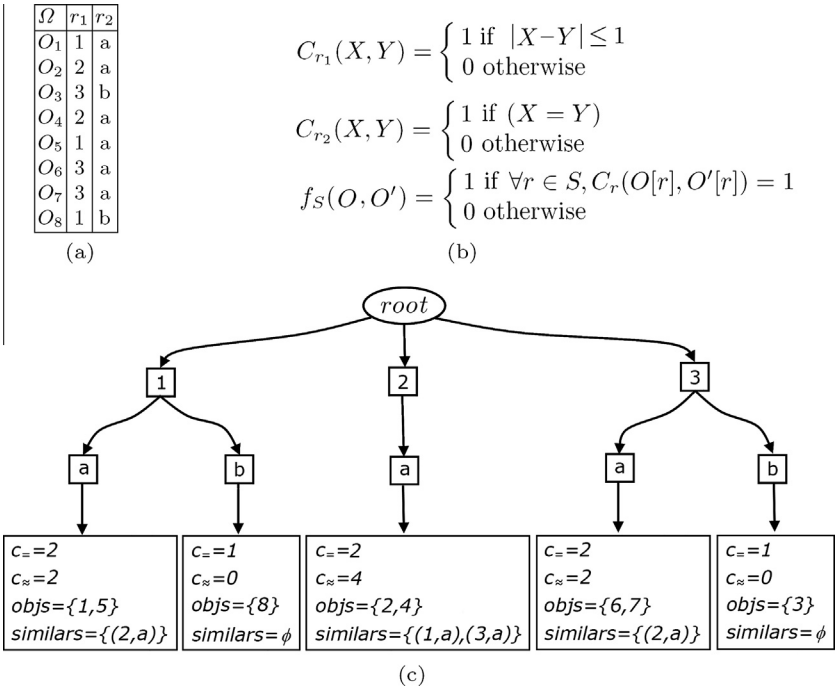

Fig. 2. Example of $STree_{\{r_1,r_2\}}$. (a) Dataset Ω. (b) Similarity function and comparison functions. (c) $STree_{\{r_1,r_2\}}$ structure.

Proposition 5. *Let f be the genuine similarity function (different from de equality) and $\bar{\bar{f}}$ the equality similarity function, if association rules are mined using $\bar{\bar{f}}$ instead f, then some association rules could be lost and some false association rules could be generated.*

Proof. Let f be the genuine similarity function (different from de equality) and $\bar{\bar{f}}$ the equality similarity function, then:

$\forall$; $O \in \Omega$; $S \subseteq R$

$$\{O' \in \Omega | \{\bar{\bar{f}}_S(O,O') = 1\} \subseteq \{O' \in \Omega | f_S(O,O') = 1\} \tag{17}$$

Because using the equality similarity function, each subdescription $I_S(O)$ is only similar to subdescriptions identical to it; but using the genuine similarity function different from de equality, each subdescription $I_S(O)$ also be similar to other subdescriptions.

Therefore, $\forall O \in \Omega$; $S \subseteq R$

$$\frac{|\{O' \in \Omega | \{\bar{\bar{f}}_S(O,O') = 1\}|}{|\Omega|} \leqslant \frac{|\{O' \in \Omega | f_S(O,O') = 1\}|}{|\Omega|} \tag{18}$$

then,

$$\bar{\bar{f}}_S - freq(O) \leqslant f_S - freq(O) \tag{19}$$

In this point, for each $O \in \Omega$,S,$S' \subseteq R$ one of the following cases could happen [1]:

1. $\bar{\bar{f}}_S - freq(O) < minFreq \leqslant f_S - freq(O)$ or $\bar{\bar{f}}_{S'} - freq(O) < minFreq \leqslant f_{S'} - freq(O)$.
   An association rule is interesting, only if the antecedent subdescription and the consequent subdescription are frequent similar patterns. Therefore, when the equality similarity function is used, the association rule $I_S(O) \rightarrow I_{S'}(O)$ is not generated because its antecedent subdescription or its consequent subdescription is a non-frequent similar pattern. However, when the genuine similarity function different from de equality is used, the association rule $I_S(O) \rightarrow I_{S'}(O)$ could be generated if its confidence is greater than or equal to *minConf*, because its antecedent subdescription and its consequent subdescription are frequent similar patterns.
2. $minFreq \leqslant \bar{\bar{f}}_S - freq(O) < f_S - freq(O)$ and $minFreq \leqslant \bar{\bar{f}}_{S'} - freq(O) < f_{S'} - freq(O)$. In this case both subdescriptions (antecedent subdescription and consequent subdescription) are similar frequent patterns. Therefore, the association rule $I_S(O) \rightarrow I_{S'}(O)$ is a candidate to be an interesting association rule, and the minimum confidence threshold *minConf* defines if the association rule is or not interesting.

   The confidence of the association rule from the definition of confidence is: for $\bar{\bar{f}}$

$$\bar{\bar{f}}_S - conf(I_S(O) \rightarrow I_{S'}(O)) = \frac{\bar{\bar{f}}_{\{S \cup S'\}} - freq(O)}{\bar{\bar{f}}_S - freq(O)} \tag{20}$$

and for f

$$f_S - conf(I_S(O) \rightarrow I_{S'}(O)) = \frac{f_{\{S \cup S'\}} - freq(O)}{f_S - freq(O)} \tag{21}$$

Due to (19), the numerators of $\bar{\bar{f}}_S$-*conf* and f_S-*conf* are related by the inequation $\bar{\bar{f}}_{\{S \cup S'\}} - freq(O) \leqslant f_{\{S \cup S'\}} - freq(O)$ and the denominators are related by the inequation $\bar{\bar{f}}_S - freq(O) \leqslant f_S - freq(O)$. However it does not exist any order relation between $\bar{\bar{f}}_S$-*conf* and f_S-*conf* and then the following cases can be considered:

Table 3
Description of datasets.

Dataset	Objects	Numerical Features	Non Numerical Features
Car Evaluation (Car)	1728	2	5
Contraceptive Method Choice (CMC)	1473	2	8
Census (Census)	32561	6	9
Poker Hand (PH)	1000000	0	11

[1] Other cases could also happen, but the presented cases are enough to proof the Proposition 5.

- $\bar{\bar{f}}_S\text{-}conf(I_S(O) \rightarrow I_{S'}(O)) < minConf \leqslant f_S\text{-}conf(I_S(O) \rightarrow I_{S'}(O))$. In this case, the association rule $I_S(O) \rightarrow I_{S'}(O)$ is not generated.
- $f_S\text{-}conf(I_S(O) \rightarrow I_{S'}(O)) < minConf \leqslant \bar{\bar{f}}_S\text{-}conf(I_S(O) \rightarrow I_{S'}(O))$. In this case, the false association rule $I_S(O) \rightarrow I_{S'}(O)$ is generated. The rule $I_S(O) \rightarrow I_{S'}(O)$ is a false association rule because its confidence using $\bar{\bar{f}}_S$ is greater than or equal to *minConf*, but its confidence using f_S is lesser than *minConf*.

□

In order to generate interesting association rules from frequent similar patterns, we propose an adaptation of the algorithm proposed in Agrawal and Srikant (1994) for mining interesting rules on binary data. The adaptation (Algorithm 3) consists in generating, for each frequent similar pattern, all possible interesting association rules that can be obtained by separating its features in two disjoint subsets, and verifying the interesting association rule conditions defined in Section 3.

Algorithm 3: FSP-GenRules

Input: F: Frequent Similar Pattern Set; f: Similarity Function; $minConf$: Minimum Confidence Threshold;
Output: RA: Interesting Association Rule Set

```
1 forall the frequent similar patterns I_S(O) ∈ F do
2     forall the Š ⊂ S such that Š ≠ ∅, I_Š(O) ∈ F, I_{S−Š}(O) ∈ F do
3         if I_S(O).freq / I_Š(O).freq ≥ minConf then
4             RA ← RA ∪ {(I_Š(O) → I_{S−Š}(O))}
5 return RA
```

6. Experimental results

In this section, we compare *STreeDC-Miner + FSP-GenRules* (*STDC + GR*) and *STreeNDC + FSP-GenRules* (*STNDC + GR*) algorithms against the *ObjectMiner + FSP-GenRules* (*ObjMiner + GR*) algorithm (provided by its authors) (Dánger et al., 2004). We conducted three experiments. In the first experiment (Section 6.1), we evaluate the performance of the proposed algorithms using a similarity function that satisfies the *f*-downward closure property. The comparison of the algorithms is in terms of the time needed to mine the frequent similar patterns and interesting association rules. In the second experiment (Section 6.2), we evaluate the proposed algorithms using a similarity function that does not satisfy the *f*-downward closure property. In this experiment, we also evaluate the performance of the proposed algorithms, in terms of the number of frequent similar patterns and interesting association rules mined by them. In the third experiment (Section 6.3), we compare the quality of the set of frequent similar patterns obtained by each algorithm.

Table 3 gives a description of the datasets[2] used in experiments 1 and 2. The experiments were done on an PC with a Intel Core 2 Duo processor at 2.0Ghz and 4 Gb of RAM and Ubuntu 10.04. The algorithms were implemented in *java*.

6.1. Experiment 1

For this experiment, we used the similarity function (1), which satisfies the *f*-downward closure property. For *Car Evaluation*, we used the comparison function (4) with $\varepsilon = 2$ for the features *Doors* and *Persons*, respectively; and for the remaining features we used the comparison function (3). For *Contraceptive Method Choice*, we used the comparison function (4) with $\varepsilon = 5$ for the feature *Age* and for the remaining features we used the comparison function (3). For *Census*, we used the comparison function (4) with $\varepsilon = 5$ for the feature *Age*. Also, we used the comparison function (4) with $\varepsilon = 1000$ for the features *Capital gain* and *Capital loss*. For the remaining features we used the comparison function (3). For *Poker Hand*, we used the comparison function (3) for all features.

The frequent similar pattern mining problem consists in finding all frequent similar patterns. Therefore, the effectiveness of an algorithm for mining frequent similar patterns can be measured as the number of frequent similar patterns mined by it, whereas the efficiency can be measured as the time spent for mining the frequent similar patterns. Analogously, the effectiveness of an algorithm for mining interesting association rules can be measured as the number of interesting association rules mined by it, whereas the efficiency can be measured as the time spent for mining the interesting association rules.

Since the similarity function used in this experiment satisfies the *f*-downward closure property, the compared algorithms obtain the same frequent similar patterns and interesting association rules. By this reason, the effectiveness of the algorithms is the same and therefore it was not evaluated. Thus, we only evaluated the efficiency of the algorithms.

In Fig. 3, the execution time for mining frequent similar patterns and interesting association rules in each dataset, for several values of *minFreq* from 0.02 to 0.20 and *minConf* = 0.8, are shown. The results of *STNDC* and *STNDC + GR* were not plotted in Fig. 3(c), (d), (g) and (h) because the *STNDC* algorithm needed more than 8 h for each one of these datasets, which is much longer than the time needed by the other algorithms.

STDC achieved better performance for mining frequent similar patterns than the other algorithms for all datasets, see Fig. 3. As a consequence *STDC + GR* also achieved better performance for mining interesting association rules than the other algorithms for all datasets. In this experiment, the time needed by *FSP-GenRules* to generate the interesting association rules from frequent similar patterns was much shorter in comparison to the time needed for mining the frequent similar patterns. The performance of *STDC* and *STDC + GR* was much better for small values of *minFreq* with respect to the other algorithms.

The performance for *STDC* was much better than the other algorithms for small values of *minFreq*.

6.2. Experiment 2

In this experiment, the similarity function (2) with $\alpha = 0.7$, which does not satisfy the *f*-downward closure property, and the same comparison functions used in the previous experiment were used.

[2] http://archive.ics.uci.edu/ml/datasets.html

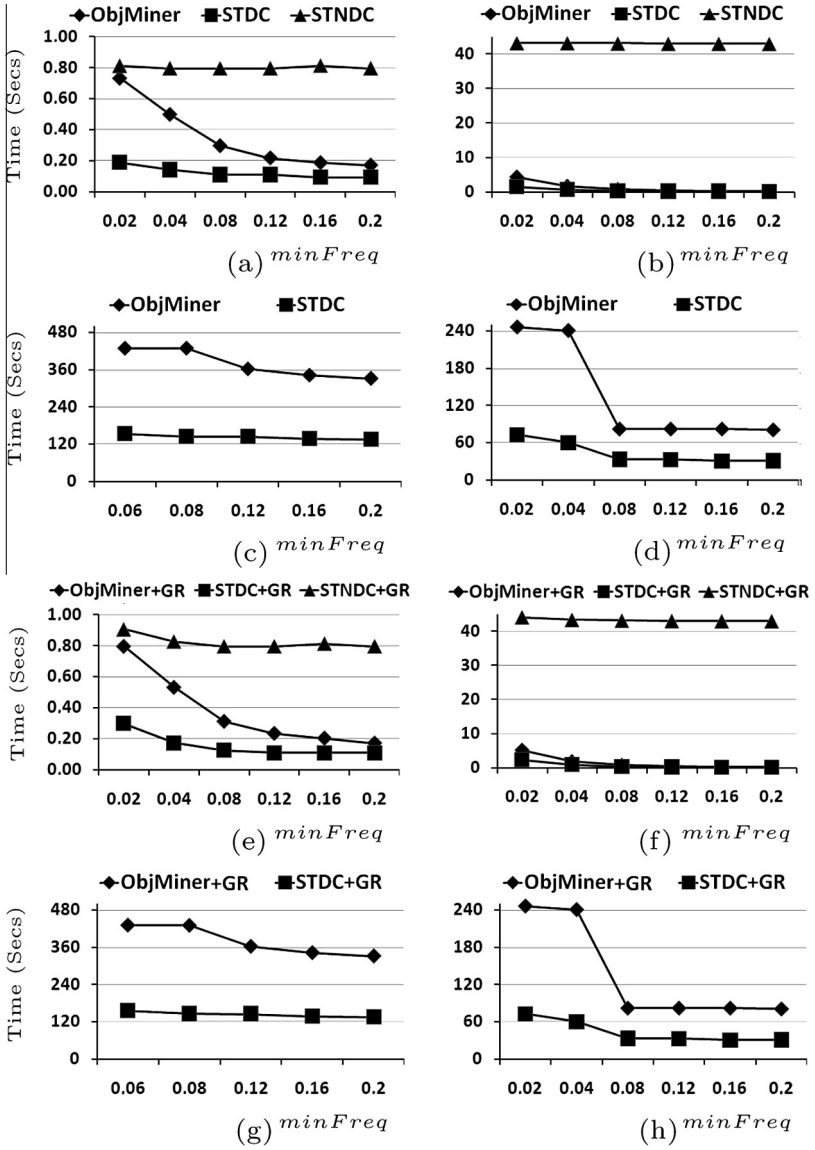

Fig. 3. Experiment results of frequent similar pattern mining and interesting association rules mining using a similarity function that fulfills the *f*-downward closure property. (a), (e) *Car Evaluation*. (b), (f) *Contraceptive Method Choice*. (c), (g) *Census*. (d), (h) *Poker Hand*.

In Fig. 4, the runtime, the number of frequent similar patterns found, as well as the ratio between the number of frequent similar patterns and the runtime; for several values of *minFrec* from 0.02 to 0.20, are shown. Analogously, in Fig. 5 the runtime, the number of interesting association rules, and the ratio between the number of interesting association rules and the runtime, for the same values of *minFrec* and *minConf* = 0.8, are shown.

As in the first experiment, the results of *STNDC* and *STNDC + GR* are not plotted for *Census* and *Poker Hand* datasets.

It is worthwhile to underline that *STNDC* finds all frequent similar patterns, but *ObjMiner* and *STDC*, which assume that the similarity function fulfills the *f*-downward closure property, could not find all frequent similar patterns for this experiment. Additionally, as it was expected, for each *minFreq* value, the set of frequent similar patterns found by *ObjMiner* was a subset of the frequent similar patterns set found by *STDC*, for all datasets. As a consequence, the set of interesting association rules obtained by *ObjMiner* was a subset of the set of interesting association rules obtained by *STDC* and the set of interesting association rule obtained by *STDC* was also a subset of the set of interesting association rules obtained by *STNDC*.

It can be noticed that *ObjMiner* lost up to 92.98% frequent similar patterns regarding all the existing frequent similar patterns and 92.50% w.r.t. the frequent similar patterns obtained by *STDC* for *Contraceptive Method Choice* (Fig. 4(f)) and it lost up to 98.80% w.r.t. the frequent similar patterns obtained by *STDC* for *Census* (Fig. 4(g)).

As a direct consequence *ObjMiner* also lost many interesting association rules. *ObjMiner* lost up to 99.79% interesting association rules regarding all the existing interesting association rules and 99.56% w.r.t. the interesting association rules obtained by *STDC* for *Contraceptive Method Choice* (Fig. 5(f)) and it lost up to 99.89% w.r.t. the interesting association rules obtained by *STDC* for *Census* (Fig. 5(g)).

Another relevant point is that *STDC* and *STDC + GR* in most of the cases had a better throughput, in terms of the ratio between *f*-frequent similar patterns and runtime, only surpassed in some few cases by *STNDC* and *STNDC + GR* respectively.

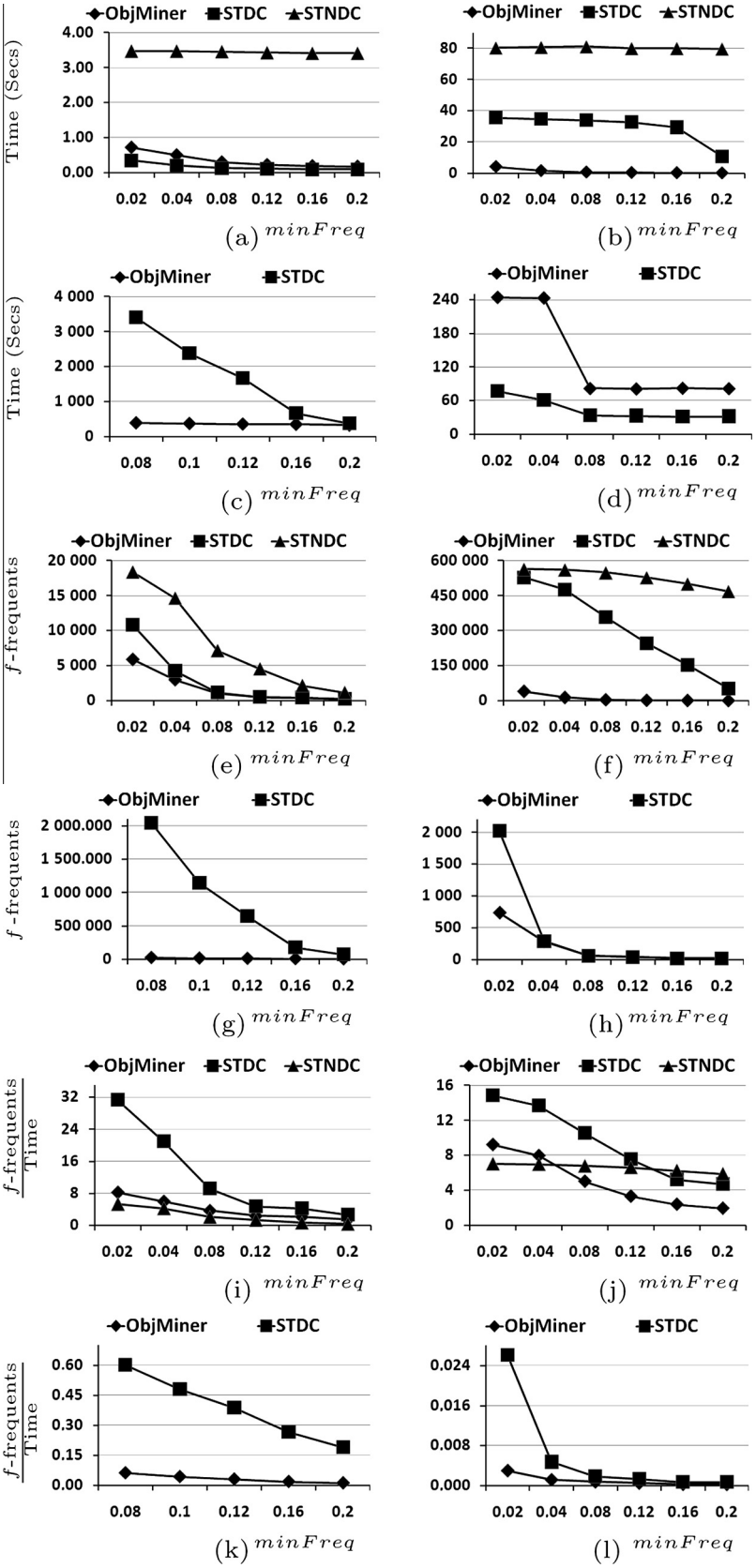

Fig. 4. Experiment results of frequent similar pattern mining using a similarity function that does not fulfill the *f*-downward closure property. (a), (e), (i) *Car Evaluation*. (b), (f), (j) *Contraceptive Method Choice*. (c), (g), (k) *Census*. (d), (h), (l) *Poker Hand*.

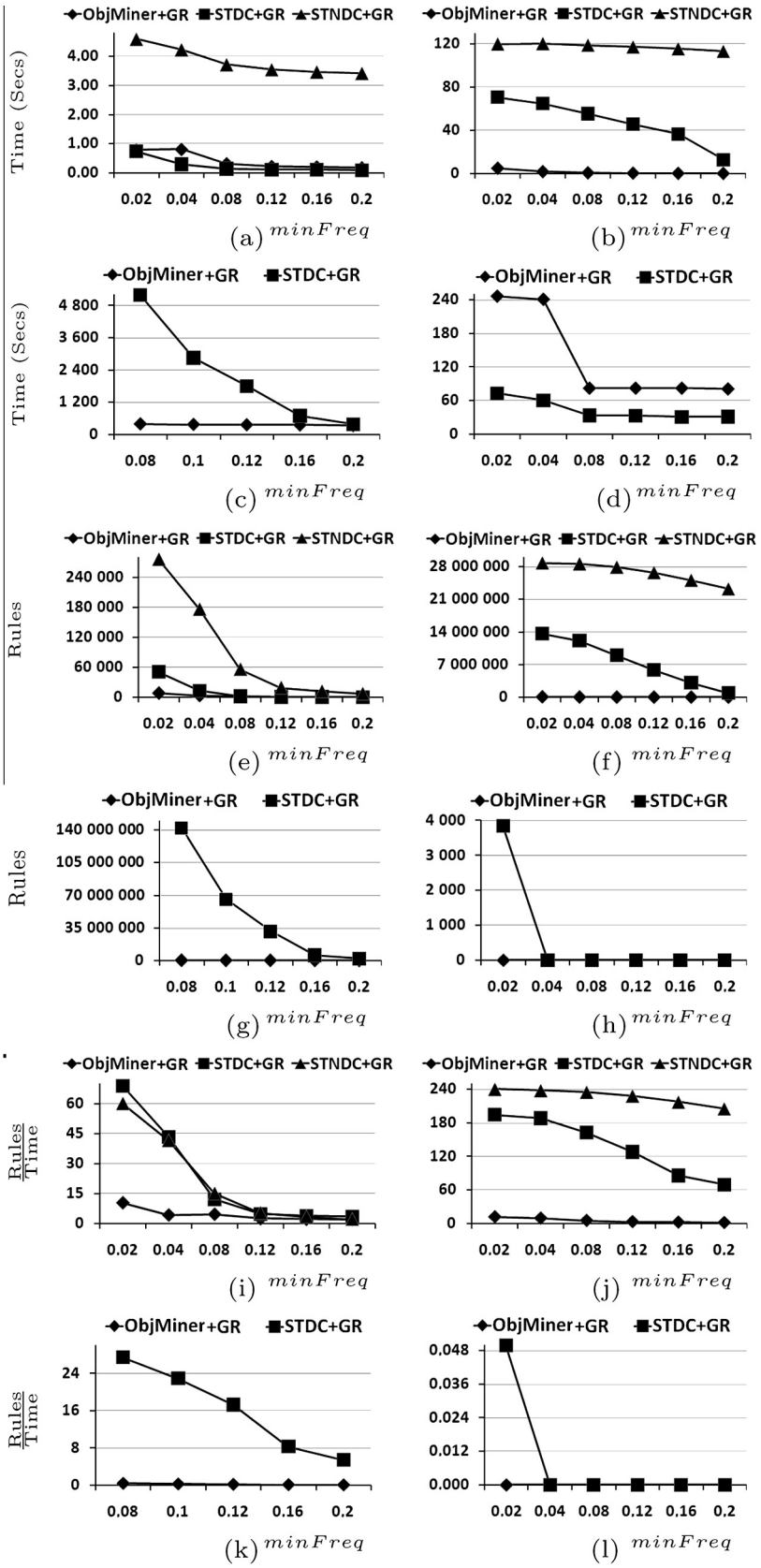

Fig. 5. Experiment results of interesting association rule mining using a similarity function that does not fulfill the *f*-downward closure property. (a), (e), (i) *Car Evaluation*. (b), (f), (j) *Contraceptive Method Choice*. (c), (g), (k) *Census*. (d), (h), (l) *Poker Hand*.

Table 4
Accuracy of the sets of frequent similar patterns for *Car Evaluation*, *Contraceptive Method Choice*, *Iris*, *Diabetes*, *Poker Hand* and *Census*.

Dataset	*minFreq*	*ObjMiner*	*STDC*	*STNDC*	*EQ-STDC*
	0.10	64.86	65.96	51.01	65.43
	0.11	64.86	65.96	51.01	65.43
	0.12	64.86	65.96	51.01	65.43
	0.13	60.52	61.01	38.91	60.14
	0.14	60.52	61.01	38.91	60.14
Car Evaluation	0.15	60.52	61.01	38.91	60.14
	0.16	55.59	56.08	31.31	55.65
	0.17	55.59	56.08	31.31	55.65
	0.18	55.59	56.08	31.31	55.65
	0.19	55.39	55.77	28.61	55.39
	0.20	55.39	55.77	28.61	55.39
Max accuracies		64.86	65.96	51.01	65.43
	0.10	41.12	40.93	38.41	35.15
	0.11	41.36	41.29	37.19	34.79
	0.12	41.05	40.99	37.27	33.76
	0.13	41.02	40.69	35.70	33.23
	0.14	40.96	40.73	35.03	30.96
CMC	0.15	40.67	40.67	34.57	30.32
	0.16	40.76	40.34	34.08	29.65
	0.17	39.74	39.14	33.33	28.34
	0.18	40.36	39.58	33.02	28.63
	0.19	40.48	39.04	32.14	30.28
	0.20	41.03	39.41	32.14	30.23
Max accuracies		41.36	41.29	38.41	35.15
	0.10	92.93	93.33	93.33	66.52
	0.11	92.93	93.33	93.33	66.22
	0.12	92.93	93.33	93.33	65.74
	0.13	91.60	91.87	92.00	65.22
	0.14	91.60	91.87	92.00	65.19
Iris	0.15	91.60	91.87	92.00	64.55
	0.16	91.60	91.87	92.00	64.56
	0.17	89.87	90.00	90.40	63.91
	0.18	89.87	90.00	90.40	63.65
	0.19	89.87	90.00	90.40	63.86
	0.20	89.87	90.00	90.40	63.83
Max accuracies		92.93	93.33	93.33	66.52
	0.10	66.88	67.04	63.42	30.82
	0.11	66.66	66.58	63.28	28.80
	0.12	65.56	65.60	62.86	26.88
	0.13	66.12	65.90	63.20	26.04
	0.14	66.00	65.96	62.90	26.12
Diabetes	0.15	64.08	63.88	62.10	24.34
	0.16	64.26	64.30	62.12	23.28
	0.17	64.46	64.48	62.28	22.26
	0.18	65.78	65.86	62.72	15.48
	0.19	66.68	66.64	63.66	14.30
	0.20	66.80	66.84	64.32	12.70
Max accuracies		66.88	67.04	64.32	30.82
	0.10	9.40	9.40	–	9.40
	0.11	8.18	8.18	–	8.18
	0.12	7.35	7.35	–	7.35
	0.13	8.91	8.91	–	8.91
	0.14	9.97	9.97	–	9.97
Poker Hand	0.15	12.79	12.79	–	12.79
	0.16	13.66	13.66	–	13.66
	0.17	14.16	14.16	–	14.16
	0.18	13.96	13.96	–	13.96
	0.19	13.96	13.96	–	13.96
	0.20	12.18	12.18	–	12.18
Max accuracies		14.16	14.16	–	14.16
	0.10	70,20	72,13	–	70,80
	0.11	70,07	71,33	–	71,40
	0.12	70,07	71,33	–	71,40
	0.13	69,60	70,80	–	70,87
	0.14	68,47	70,07	–	70,53
Census	0.15	66,87	69,73	–	69,80
	0.16	66,87	69,73	–	69,80
	0.17	66,67	68,07	–	69,60
	0.18	65,00	67,07	–	69,47
	0.19	63,60	66,40	–	70,87
	0.20	63,60	66,40	–	70,87
Max accuracies		70,20	72,13	–	71,40

6.3. Experiment 3

As it was shown in the previous experiments, the proposed algorithms obtain different amounts of frequent similar patterns and interesting association rules when the similarity function does not fulfill the *f*-downward closure property. Nevertheless, obtaining a big set of frequent similar patterns or interesting association rules does not necessarily imply that this set is good. For this reason, it is desirable to evaluate the quality of the sets of frequent similar patterns and interesting association rules obtained by each algorithm. There is not a standard measure for evaluating the quality of a set of interesting association rules. However, it is feasible to evaluate the quality of the set of frequent similar patterns obtained by an algorithm as the accuracy that a supervised classifier reaches when it uses this set of patterns to classify unseen objects. In this experiment we follow this idea for comparing the quality of the set of frequent similar patterns obtained by each algorithm.

For this experiment, we used a simple classifier based on frequent similar patterns, which in the training phase obtains the set of frequent similar patterns from each class and removes all the frequent similar patterns that appear in more than one class, in order to keep only patterns that represent objects from a single class. In the classification phase, each object of the testing dataset is classified in the class where there are more frequent similar patterns, which are similar to its subdescriptions. For each dataset and for each value of *minFreq* we randomly select 50% of the dataset for training and the remaining objects for testing. This process was repeated 10 times and the average of the 10 results is reported. Since *STreeNDC-Miner* was too slow for *Poker Hand* and *Census* datasets, the results for these datasets are not reported. Additionally for the *Car Evaluation* and *Contraceptive Method Choice* datasets, we included the *Iris* and *Diabetes* datasets which have 150 and 768 objects and 4 and 8 numerical features, respectively.

For each dataset and for all the algorithms used for mining frequent similar patterns, the classification was evaluated for different values of the *minFreq* threshold, from 0.10 to 0.20. We use the same similarity function used in the previous experiment with $\alpha = 0.7$, which does not satisfy the *f*-downward closure property. In Table 4, we show the results.

The last column in Table 4 contains the result of the *STreeDC-Miner* algorithm using the equality similarity function. The set of frequent patterns obtained by *STreeDC-Miner* using the equality function is the same set of frequent patterns obtained by classical frequent pattern mining algorithms. Notice that, for applying a classical frequent pattern mining algorithm the original dataset must be transformed into a Binary dataset where each Boolean feature represents a feature value into the original dataset. From this column, it can be seen that using the equality function, like in the classical approach, the classification accuracy obtained by the set of frequent patterns is lower than the classification accuracy obtained by the set of frequent similar patterns obtained using the similarity function (2) which is different from the equality. These results are due to (I) the set of frequent patterns obtained using the equality could be too small (the frequency of the subdescriptions using the equality tends to be lesser than the frequency of the subdescriptions using a similarity function different from the equality), then it could be insufficient to cover new objects; (II) the frequent patterns obtained using the equality could be too specific (new objects that contain patterns very similar to the frequent patterns, but not identical to them, would be not covered by the frequent patterns).

In this experiment, the classification accuracy obtained using the frequent similar patterns obtained by *STDC* was the best, followed by the accuracy obtained using the frequent similar patterns obtained by *ObjMiner*, *STNDC* and *EQ-STDC* in this order. Notice that, the set of frequent similar patterns obtained by *STDC* is a superset of the fre-

quent similar patterns obtained by *ObjMiner*. For this reason, we can affirm that the frequent similar patterns lost by *ObjectMiner* negatively affect the classification accuracy i.e. those additional patterns obtained by *STreeDC-Miner* contribute to get better accuracies.

STDC and *ObjMiner* obtained better classification accuracies than using the frequent similar patterns obtained by *STNDC*. This fact is due to the additional frequent similar patterns obtained by *STNDC* are more specific (larger) than the frequent similar patterns obtained by *STDC* and *ObjMiner* (remember that *STDC* and *ObjMiner* prune large frequent similar patterns that are superdescription of non-frequent similar patterns, while *STNDC* computes frequent similar patterns from those patterns containing all the features to those smaller reducing the set of features). As a consequence, the classifier could be overfit and its generalization capability could be reduced.

7. Conclusions

In this paper, we focused on the problem of mining frequent patterns and association rules using similarities. We introduce several properties and proved several propositions that allow pruning the search space of frequent similar patterns. Based on these properties and propositions an efficient data structure to store all necessary information about object subdescriptions and their similarities was introduced. Also, a novel and efficient algorithm for mining frequent similar patterns for similarity functions that fulfill the *f*-downward closure property and another algorithm suitable for any similarity function, were proposed. Additionally an adaptation of the *GenRules* algorithm to generate interesting association rules from frequent similar patterns was proposed and it was shown that some rules can be lost and other false rules can arise when the equality is used instead of a similarity function different from the equality.

The experimental results have shown that the proposed algorithms have better behavior, in time and number of frequent similar patterns and interesting association rules, than *ObjectMiner*. Additionally, the quality of the set of frequent similar patterns computed by each algorithm was measured by means of a classifier. From this experiment we conclude that, our algorithm *STreeDC-Miner* obtains better frequent similar patterns than those patterns obtained by the other algorithms. Also, it was shown that those frequent similar patterns obtained using similarities different from the equality are better than those obtained using the equality.

In those problems where we know that the similarity function fullfills the Downward Closure property, *STreeDC-Miner* is faster that *RP-Miner*. While in those problems where we know that the similarity function does not fulfill the Downward Closure property *STreeNDC-Miner* finds all the patterns while *RP-Miner* finds only a subset. Therefore the algorithms proposed in this paper constitute an alternative to those cases where *RP-Miner* does not provide good results. Therefore, the results presented in this paper complete the study of algorithms for mining frequent similar patterns on mixed data using Boolean similarity functions different from the equality.

As future work, reducing the set of frequent similar patterns and association rules without losing information will be an interesting and imperative research direction. Another interesting research direction is mining frequent similar patterns and association rules for non Boolean similarity functions.

Acknowledgements

This work is partly supported by the National Council of Science and Technology of Mexico under the project CB2008-106443 and Grant No. 32086.

References

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules in large databases. Research Report RJ 9839, IBM Almaden Research Center, San Jose, California.

Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. In *1993 ACM SIGMOD international conference on management of data* (pp. 207–216). Washington, USA.

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. In *20th International Conference on Very Large Data Bases, Santiago de Chile, Chile* (pp. 487–499). Morgan Kaufmann.

Alatas, B., Akin, E., & Karci, A. (2008). Modenar: multi-objective differential evolution algorithm for mining numeric association rules. *Applied Soft Computing, 8*, 646–656.

Ayubi, S., Muyeba, M. K., Baraani, A., & Keane, J. (2009). An algorithm to mine general association rules from tabular data. *Information Sciences, 179*, 3520–3539.

Chaves, R., Ramrez, J., Grriz, J. M., et al. (2012). Association rule-based feature selection method for Alzheimers disease diagnosis. *Expert Systems with Applications, 39*, 11766–11774.

Chaves, R., Ramrez, J., & Grriz, J. M. (2013). Integrating discretization and association rule-based classification for Alzheimers disease diagnosis. *Expert Systems with Applications, 40*, 1571–1578.

Chen, G., & Wei, Q. (2000). Fuzzy association rules and the extended mining algorithms. *Information Sciences, 147*, 201–228.

Choi, D.W., & Hyun, Y.J., (2010). Transitive association rule discovery by considering strategic importance. In *2010 10th IEEE international conference on computer and information technology* (pp. 1654–1659). Bradford UK.

Dánger, R., Ruiz-Shulcloper, J., & Berlanga, R. (2004). Objectminer: a new approach for mining complex objects. In *ICEIS-2004* (pp. 42–47). Oporto, Portugal.

Dua, S., Singh, H., & Thompson, H. W. (2009). Associative classification of mammograms using weighted rules. *Expert Systems with Applications, 36*, 9250–9259.

Ertoz, L., Eilertson, E., Lazarevic, A., Tan, P. N., Kumar, V., Srivastava, J., et al. (2004). *The MINDS: Minnesota Intrusion Detection System. Next Generation Data Mining*. MIT Press.

Gómez, J., Rodríguez, O., Valladares, S., Ruiz-Shulcloper, J., et al. (1994). Applying Mathematical Modeling. *Geophysical International, 33*, 447–467.

Hernández-León, Raudel, Carrasco-Ochoa, Jesús A., Martínez-Trinidad, José Fco., & Hernández-Palancar, José (2012). Classification based on specific rules and inexact coverage. *Expert Systems with Applications, 39*, 11203–11211.

Hu, T., Sung, S. Y., Xiong, H., & Fu, Q. (2008). Discovery of maximum length frequent itemsets. *Information Sciences, 178*, 69–87.

Kalpana, B., & Nadarajan, R. (2008). Incorporating heuristics for efficient search space pruning in frequent itemset mining strategies. *Current Science, 94*, 97–101.

LaRosa, C., Xiong, L., & Mandelberg, K. (2008). Frequent pattern mining for kernel trace data. In *2008 ACM Symposium on Applied Computing* (pp. 880–885). Fortaleza, Ceara, Brazil: ACM.

Leong, K., Chan, S., Ng, V., & Shiu, S. (2008). Introduction of STEM: Space–Time-Event Model for crime pattern analysis. *Asian Journal of Information Technology, 7*, 516–523.

Li, X., & Deng, Z. (2010). Mining frequent patterns from network flows for monitoring network. *Expert Systems with Applications, 37*, 8850–8860.

Li-Min, T., Shu-Jing, L., & Don-Lin, Y. (2010). Efficient mining of generalized negative association rules. In *2010 IEEE international conference on granular computing*. (pp. 471–476). Silicon Valley, USA.

Lopez, F. J., Blanco, A., Garcia, F., Cano, C., & Marin, A. (2008). Fuzzy association rules for biological data analysis: a case study on yeast. *BMC Bioinformatics, 9*, 107.

Malik, H. H., Kender, J. R., Fradkin, D., & Moerchen, F. (2010). Hierarchical document clustering using local patterns. *Data Mining and Knowledge Discovery, 21*, 153–185.

Martínez-Trinidad, J. F., Ruiz-Shulcloper, J., & Lazo-Cortés, M. S. (2000). Structuralization of universes. *Fuzzy Sets and Systems, 112*, 485–500.

Nahar, J., Imam, T., Tickle, K. S., & Chen, Y. P. (2013). Association rule mining to detect factors which contribute to heart disease in males and females. *Expert Systems with Applications, 40*, 1086–1093.

Nguyen, Loan T. T., Vo, Bay, Hong, Tzung-Pei, & Thanh, Hoang Chi (2012). Classification based on association rules: a lattice-based approach. *Expert Systems with Applications, 39*, 11357–11366.

Ortiz-Posadas, M. R., Vega-Alvarado, L., & Toni, B. (2009). A mathematical function to evaluate surgical complexity of cleft lip and palate. *Computer Methods and Programs Biomedicine, 94*, 232–238.

Patil, B.M., Joshi, R.C., & Toshniwal, D. (2010). Association rule for classification of type-2 diabetic patients. In *2010 second international conference on machine learning and computing*. (pp. 330–334). Bangalore, India.

Rodríguez-González, A. Y., Martínez-Trinidad, J. F., Carrasco-Ochoa, J. A., & Ruiz-Shulcloper, J. (2008). Mining frequent similar patterns on mixed data. In J. Ruiz-Shulcloper & W. Kropatsch (Eds.), *Progress in Pattern Recognition. Image Analysis and Applications, LNCS* (vol. 5197, pp. 136–144). Berlin, Germany: Springer-Verlag.

Rodríguez-González, A. Y., Martínez-Trinidad, J. F., Carrasco-Ochoa, J. A., & Ruiz-Shulcloper, J. (2011). RP-Miner: a relaxed prune algorithm for frequent similar pattern mining. *Knowledge and Information Systems, 27*, 451–471.

Ruiz-Shulcloper, J., & Fuentes-Rodríguez, A. (1981). A cybernetic model to analyze juvenile delinquency. *Revista Ciencias Matemticas, 2*, 123–153.

Sánchez, D., Vila, M. A., Cerda, L., & Serrano, J. M. (2009). Association rules applied to credit card fraud detection. *Expert Systems with Applications, 36*, 3630–3640.

Wong, K. W., Zhou, S., Yang, Q., & Yeung, J. M. (2005). Mining customer value: from association rules to direct marketing. *Data Mining and Knowledge Discovery, 11*, 57–79.

Xin, L., & Zhi-Hong, D. (2010). Mining frequent patterns from network flows for monitoring network. *Expert Systems with Applications, 37*(12).

Ya-Han, H., & Yen-Liang, C. (2006). Mining association rules with multiple minimum supports: a new mining algorithm and a support tuning mechanism. *Decision Support Systems, 42*, 1–24.

Yunyan, L., & Juan, C. (2010). Application of association rules mining in marketing decision-making based on rough set. In *2010 International conference on e-business and e-government* (pp. 3749–3752). Guangzhou, China.