Closed-set-based Discovery of Representative Association Rules Revisited

José L Balcázar, Cristina Tîrnăucă

Departamento de Matemáticas, Estadística y Computación
Universidad de Cantabria, Santander, Spain
{joseluis.balcazar, cristina.tirnauca}@unican.es

Abstract. The output of an association rule miner is often huge in practice. This is why several concise lossless representations have been proposed, such as the "essential" or "representative" rules. We revisit the algorithm given by Kryszkiewicz (Int. Symp. Intelligent Data Analysis 2001, Springer-Verlag LNCS 2189, 350–359) for mining representative rules. We show that its output is sometimes incomplete, due to an oversight in its mathematical validation, and we propose an alternative complete generator that works within only slightly larger running times.

1 Introduction

Association rule mining is among the most popular conceptual tools in the field of Data Mining. We are interested in the process of discovering and representing regularities between sets of items in large scale transactional data. Syntactically, the association rule representation has the form of an implication, $X \rightarrow Y$; however, whereas in Logic such an expression is true if and only if Y holds whenever X does, an association rule is a partial implication, in the sense that it is enough if Y holds *most of the times* X does.

To endow association rules with a definite semantics, we need to make precise how this intuition of "most of the times" is formalized. There are many proposals for this formalization. One of the frequently used measures of intensity of this kind of partial implication is its *confidence*: the ratio between the number of transactions in which X and Y are seen together and the number of transactions that contain X. In most application cases, the search space is additionally restricted to association rules that meet a minimal *support* criterion, thus avoiding the generation of rules from items that appear very seldom together in the dataset (formal definitions of support and confidence are given in Section 2.1).

Many association rule miners exists, Apriori (see Agrawal et al. (1996)) being one of the most widely discussed and used. The major problem shared by all mining algorithms is that, in practice, even for reasonable support and confidence thresholds, the output is often huge. Therefore, several concise lossless representations of the whole set of association rules have been proposed. These representations are based on different notions of "redundancy". In one of these, a rule is redundant if it is possible to compute exactly its confidence and support from other information such as the confidences and supports of other *informative* rules (see

Luxenburger (1991); Kryszkiewicz (2002); Hamrouni et al. (2008)); this is a quite demanding property. We settle for a weaker version proposed in several works; informally, in that version, a rule is *redundant* with respect to another one if its confidence and support are always greater, in *any* dataset. To avoid this redundancy, exactly one notion has been identified in several sources, namely the *representative rules* (precise definitions and references are given below).

We focus in this paper on the main results of Kryszkiewicz (2001), where a purportedly faster algorithm to construct representative rules is given, and show by an example that that algorithm is not guaranteed to always output all representative rules, because it is based on a property that does not hold in general; namely, the characterization of the frequent closed sets that admit a decomposition into representative rules misses some such sets. We propose an alternative, complete characterization, leading us to the proposal of a first alternative algorithm that is guaranteed to output all the representative rules: we pre-compute, for each closed set, some parameters that depend on the confidence and support thresholds, and then use the above mentioned new characterization to generate all representative rules.

Compared to the previous, potentially incomplete algorithm in Kryszkiewicz (2001), this algorithm, guaranteed to be complete, has a main drawback: in Kryszkiewicz (2001), the internal local parameters only depend on the support threshold, but in our algorithm these parameters depend also on confidence. Therefore, each time a new confidence threshold is introduced by the user, the algorithm has to redo all computations. Thus, we provide a second algorithm, composed of two parts: the first one is a pre-processing phase, dependent only on support, in which a subdivision of the interval $(0, 1]$ is associated to each closed itemset, and the second part uses this partition to determine, for a given value of the confidence threshold, which are those sets that can generate representative rules.

There are a couple of subtle differences between one of the usual definitions of association rule (the one we employ) and the one in Kryszkiewicz (2001). First, we do allow having rules with empty antecedent (clearly, all of them have confidence equal to the normalized support of the consequent). Moreover, we do not require the inequalities to be strict when imposing a given support and confidence threshold. This is just a small detail that comes handy when the user is interested in obtaining the set of all representative rules of confidence 1. However, we have carefully tuned all our argumentations in such a way that these differences are not relevant; for instance, we have chosen a counterexample that invalidates Property 9 of Kryszkiewicz (2001) independently of which of the two definitions is used.

The article is structured as follows. In Section 2 we introduce the basic notions and notations that will be used throughout the paper and part of the contents of Kryszkiewicz (2001); and we show that the algorithm provided there is not guaranteed to always provide the whole set of representative rules. In Section 3 we define new parameters and discuss their usefulness in generating the set of all representative rules, providing also efficient algorithms for this task. Section 4 contains a comparison of our approach with the one in Kryszkiewicz (2001) on some datasets. Concluding remarks and further research topics are presented in Section 5.

2 Preliminaries

A given set of available items $\mathcal{U}$ is assumed; subsets of it are called itemsets. We will denote itemsets by capital letters from the end of the alphabet, and use juxtaposition to denote union, as in XY. The inclusion sign as in $X \subset Y$ denotes proper subset, whereas improper

inclusion is denoted $X \subseteq Y$. For a given dataset $\mathcal{D}$, consisting of n transactions, each of which is an itemset labeled with a unique transaction identifier, we define the *support* $sup(X)$ of an itemset X as the ratio between the cardinality of the set of transactions that contain X and the total number of transactions n. An itemset X is called *frequent* if its support is greater than or equal to some user-defined threshold $\tau \in (0, 1]$.

Given a set $X \subseteq \mathcal{U}$, the *closure* $\overline{X}$ *of* X is the maximal set (with respect to the set inclusion) $Y \subseteq \mathcal{U}$ such that $X \subseteq Y$ and $sup(X) = sup(Y)$. It is easy to see that $\overline{X}$ is uniquely defined. We say that a set $X \subseteq \mathcal{U}$ is *closed* if $\overline{X} = X$.

Closure operators are characterized by the three properties of monotonicity $X \subseteq \overline{X}$, idempotency $\overline{\overline{X}} = \overline{X}$, and extensivity, $\overline{X} \subseteq \overline{Y}$ if $X \subseteq Y$. Intersections of closed sets are closed.

A *minimal generator* is a set X for which all proper subsets have closures different from the closure of X (that is, X is a minimal generator if and only if $sup(Y) > sup(X)$ for all $Y \subset X$). We denote by $F_\tau = \{X \subseteq \mathcal{U} \mid sup(X) \geq \tau\}$ the set of all frequent itemsets.

Also, $FC_\tau = \{X \in F_\tau \mid \overline{X} = X\}$ represents the set of all frequent closed sets, and $FG_\tau = \{X \in F_\tau \mid \forall Y \subset X, sup(Y) > sup(X)\}$ is the set of all frequent minimal generators. Note that FC_τ constitutes a concise lossless representation of frequent itemsets, since knowing the support of all sets in FC_τ is enough to retrieve the support of all sets in F_τ.

Example 1 Let $\mathcal{D}$ be the dataset represented in Table 1 where the universe $\mathcal{U}$ of attributes is $\{a, b, c, d, e, f\}$, and consider $\tau = 0.15$. Clearly, all subsets of $\mathcal{U}$ are frequent, FC_τ = $\{\emptyset, a, b, c, ab, ac, ad, bc, abcde, abcdef\}$ and FG_τ = $\{\emptyset, a, b, c, d, e, f, ab, ac, bc, bd, cd, abc\}$ (we abuse the notation and denote sets by the juxtaposition of their constituent elements).

TAB. 1 – Dataset $\mathcal{D}$

a	b	c	d	e	f
1	1	1	1	1	1
1	1	1	1	1	0
1	1	0	0	0	0
1	0	1	0	0	0
0	1	1	0	0	0
1	0	0	1	0	0

2.1 Association Rules and Representative Rules

Given X in F_τ, two definitions, with longer names, are introduced in Kryszkiewicz (2001):

$mxs_\tau(X) = \max(\{sup(Z) \mid Z \in FC_\tau, Z \supset X\} \cup \{0\})$,

$mns_\tau(X) = \min(\{sup(Y) \mid Y \in FG_\tau, Y \subset X\} \cup \{\infty\})$.

That is, $mxs_\tau(X)$ represents the maximum support of all proper frequent closed supersets of X, and $mns_\tau(X)$ is the minimum support of minimal generators that are proper subsets of X. The extra 0 and ∞ are added in order to make sure that $mxs_\tau(X)$ and $mns_\tau(X)$ are defined even for the cases in which X has no proper supersets that are frequent and closed, or when it does not have proper subsets that are minimal generators. It is easy to check that $mxs_\tau(X) \leq sup(X) \leq mns_\tau(X)$. Moreover, it can be shown that:

Proposition 1 (Kryszkiewicz (2001)) *Let $X \in F_\tau$. Then $X \in FC_\tau$ iff $sup(X) > mxs_\tau(X)$, and $X \in FG_\tau$ iff $sup(X) < mns_\tau(X)$.*

The types of association rules considered in this work are implications of the form $X \to Y$, where $X, Y \subseteq \mathcal{U}$, $Y \neq \emptyset$ and $X \cap Y = \emptyset$. In Kryszkiewicz (2001), rules with $X = \emptyset$ are disallowed, but we do permit them as in practice such rules often play a useful role related to coverings, described below. The *confidence* of $X \to Y$ is $conf(X \to Y) = sup(XY)/sup(X)$, and its *support* is $sup(X \to Y) = sup(XY)$. The problem of mining association rules consists in generating all rules that meet the minimum support and confidence threshold criteria. Let $AR_{\tau,\gamma} = \{X \to Y \mid sup(X \to Y) \geq \tau, conf(X \to Y) \geq \gamma\}$.

Since the whole set of association rules is quite big in real-world applications, a number of formalizations of the notion of *redundancy* among association rules have been introduced (see Aggarwal and Yu (2001); Balcázar (2010); Cristofor and Simovici (2002); Kryszkiewicz (1998b); Luxenburger (1991); Pasquier et al. (2005); Phan-Luong (2001); Zaki (2004), the survey Kryszkiewicz (2002), and section 6 of Ceglar and Roddick (2006)). In one common approach, the *cover* set $C(X \to Y)$ of a rule $X \to Y$ is defined by $C(X \to Y) = \{X' \to Y' \mid X \subseteq X' \text{ and } XY \supseteq X'Y'\}$. Such rules $X' \to Y'$ are redundant with respect to $X \to Y$ in the following sense:

Proposition 2 (Kryszkiewicz (1998b),Aggarwal and Yu (2001)) *Let $r : X \to Y$ and $r' : X' \to Y'$ be association rules. Then $r \in C(r')$ implies $sup(r) \geq sup(r')$ and $conf(r) \geq conf(r')$.*

In fact, this implication is a full characterization, that is, if $X' \to Y'$ has always at least the same confidence and at least the same support as $X \to Y$ then it must belong to the cover set (see Balcázar (2010)). Avoiding such redundancies leads to the set $RR_{\tau,\gamma}$ of *representative association rules*. A rule r in $AR_{\tau,\gamma}$ is said to be *representative*, or sometimes *essential*, if it is not contained in the cover set of any other rule in $AR_{\tau,\gamma}$.

$$RR_{\tau,\gamma} = \{r \in AR_{\tau,\gamma} \mid \forall r' \in AR_{\tau,\gamma}(r \in C(r') \Rightarrow r = r')\}.$$

Under different names, this notion has been proposed and studied in several sources, e.g. Aggarwal and Yu (2001); Kryszkiewicz (1998b); Phan-Luong (2001).

Proposition 3 (Kryszkiewicz (1998a,b)) *The following properties hold:*

- $RR_{\tau,\gamma} = \{X \to Y \in AR_{\tau,\gamma} \mid \neg\exists X' \to Y' \in AR_{\tau,\gamma}, (X = X', XY \subset X'Y')$ *or* $(X \supset X', XY = X'Y')\}$
- *if $X \to Z \backslash X$ with $X \subset Z$ is in $RR_{\tau,\gamma}$ then $Z \in FC_\tau$ and $X \in FG_\tau$.*

Therefore, any algorithm that aims at the discovery of all representative rules should consider only rules of the form $X \to Z \backslash X$ with $X \subset Z$, $Z \in FC_\tau$ and $X \in FG_\tau$. Clearly, not all sets in FC_τ can be decomposed in such a way, and one should look only into those that do.

Example 2 Consider the dataset in Example 1. The set ad is both frequent and closed, but none of the rules $a \to d$, $d \to a$ or $\emptyset \to ad$ are representative given the thresholds $\tau = 0.15$ and $\gamma = 0.33$: $a \to d$ is in the cover set of $a \to bd$, $d \to a$ is in the cover set of $d \to ab$ and $\emptyset \to ad$ is in the cover set of $\emptyset \to abd$. Also, it is easy to check that, at $\gamma = 0.4$, one can obtain representative rules exactly out of the following closed sets: ab, ac, ad, bc, $abcde$, and $abcdef$.

So, if we denote by $RI_{\tau,\gamma}$ the set of all frequent closed itemsets from which at least one representative rule can be generated, one possible approach to representative rule mining is to synthesize first the set $RI_{\tau,\gamma}$, and then, for each element Z in $RI_{\tau,\gamma}$, to find a non-empty subset X such that $X \rightarrow Z \backslash X$ is representative. This is precisely the idea behind Algorithm *GenRR* in Kryszkiewicz (2001). The problem there is that the characterization of the set $RI_{\tau,\gamma}$ given by Proposition 9 of the same paper (on page 355) is incorrect, possibly leaving out some of the sets that can lead to representative rules. Namely, it is stated that $RI_{\tau,\gamma} = \{X \in FC_\tau \mid sup(X) > \gamma * mns_\tau(X) \geq mxs_\tau(X)\}$; right-to-left inclusion indeed holds, but equality does not hold in general, as one can see from the following counterexample.

Example 3 *Consider the itemset $X = abcde$ in Example 1, and assume $\tau = 0.15$ and $\gamma = 0.4$. Let us verify that $abcde \in RI_{\tau,\gamma} \backslash \{X \in FC_\tau \mid sup(X) > \gamma * mns_\tau(X) \geq mxs_\tau(X)\}$. Clearly, the rule $b \rightarrow acde$ is in $AR_{\tau,\gamma}$, having support 2/6 and confidence 0.5. Moreover, by extending the right-hand side or moving the item b to the right-hand side we get only the rules $b \rightarrow acdef$, $\emptyset \rightarrow abcde$ and $\emptyset \rightarrow abcdef$ of confidence 1/4, 2/6 and 1/6, respectively. Hence, we can conclude that $b \rightarrow acde \in RR_{\tau,\gamma}$. On the other hand, $mxs_\tau(X) = 1/6$ and $mns_\tau(X) = 2/6$, so $\gamma * mns_\tau(X) = 0.8/6$ is strictly smaller than $mxs_\tau(X)$. In this case, Algorithm* GenRR *does not work correctly since it does not list the rule $b \rightarrow acde$ as being representative.*

An alternative counterexample is given in the proof of Lemma 1 below.

3 Bounds that Help Characterize Representative Rules

The goal of pruning off sets that do not give representative rules, by keeping only $RI_{\tau,\gamma}$, cannot be reached using the bounds given, as we have seen that this set comprises all X in FC_τ with $sup(X) \geq \gamma * mns_\tau(X) > mxs_\tau(X)$ but may also include other frequent closed sets X that do not satisfy the condition $\gamma * mns_\tau(X) > mxs_\tau(X)$. We consider two alternatives.

3.1 Closed Sets Instead of Minimal Generators

For closed X, $mns_\tau(X)$ is almost the same thing as the minimal support among all proper subsets of X, or again among all proper closed subsets of X; all these notions coincide when X is its own minimal generator, otherwise they only differ due to the minimal generators of X. Therefore it makes sense to try and exclude the minimal generators of X from consideration. This way, we get another parameter,

$$bmns_\tau(X) = \min(\{sup(Y) \mid Y \in FC_\tau, Y \subset X\} \cup \{\infty\}).$$

The value of $bmns_\tau$ is never smaller than mns_τ as we shall shortly see. Thus, there will be more sets that meet the condition $\gamma * bmns_\tau(X) > mxs_\tau(X)$.

Proposition 4 *The following properties hold.*

- $bmns_\tau(X) = \min(\{sup(Y) \mid Y \in FG_\tau, \overline{Y} \subset X\} \cup \{\infty\})$,
- $mns_\tau(X) \leq bmns_\tau(X)$,
- *if* $X \in FC_\tau \cap FG_\tau$ *then* $mns_\tau(X) = bmns_\tau(X)$,

Proof. We omit the proof of the first two claims because they are straightforward. So, let X be a frequent closed set that is also a minimal generator. If $X = \emptyset$, then $mns_\tau(X) = bmns_\tau(X) = \infty$. Otherwise, let $Y \in FG_\tau$ be such that $Y \subset X$ and $mns_\tau(X) = sup(Y)$. Clearly, $\overline{Y} \in FC_\tau$ and $\overline{Y} \subseteq \overline{X} = X$. Since $X \in FG_\tau$ and $Y \subset X$, $sup(Y) > sup(X)$ and hence $sup(\overline{Y}) > sup(X)$, and therefore $\overline{Y} \subset X$. We get $sup(\overline{Y}) \geq bmns_\tau(X)$ and $mns_\tau(X) \geq bmns_\tau(X)$. Combining it with the fact that $mns_\tau(X) \leq bmns_\tau(X)$ always holds, we conclude that $mns_\tau(X) = bmns_\tau(X)$. ■

Unfortunately, the new parameter can still leave out some sets in $RI_{\tau,\gamma}$.

Lemma 1 $RI_{\tau,\gamma} \not\subseteq \{X \in FC_\tau \mid sup(X) > \gamma * bmns_\tau(X) \geq mxs_\tau(X)\}$.

Proof. Let $\mathcal{U} = \{a, b, c\}$ and $\mathcal{D}$ be the dataset containing the following 13 transactions: $t_1 = \cdots = t_8 = abc, t_9 = ab, t_{10} = t_{11} = t_{12} = a, t_{13} = b$; assume $\tau = 0.07$ and $\gamma = 0.7$. One can check that, although $ab \in RI_{\tau,\gamma}$ (since $a \rightarrow b \in RR_{\tau,\gamma}$), both $bmns_\tau(ab) = 10/13$ and $mns_\tau(ab) = 10/13$; but $\gamma * mns_\tau(ab) = \gamma * bmns_\tau(ab) = 7/13 < 8/13 = mxs_\tau(ab)$. ■

The next construction shows that by using $bmns_\tau$ instead of mns_τ we can even leave out some sets in $RI_{\tau,\gamma}$ that would not have been left out otherwise.

Lemma 2 $RI_{\tau,\gamma} \cap \{X \in FC_\tau \mid sup(X) > \gamma * mns_\tau(X) \geq mxs_\tau(X)\} \not\subseteq \{X \in FC_\tau \mid sup(X) > \gamma * bmns_\tau(X) \geq mxs_\tau(X)\}$.

Proof. Let $\mathcal{U} = \{a, b, c, d, e\}$ and $\mathcal{D}$ be a dataset containing 35 transactions: $t_1 = t_2 = abcde, t_3 = t_4 = t_5 = abcd, t_6 \cdots = t_{20} = a$ and $t_{21} = \cdots t_{35} = b$. Pick $\tau = 0.05$ and $\gamma = 0.75$. Note that $ab \rightarrow cd \in RR_{\tau,\gamma}$, and therefore $abcd \in RI_{\tau,\gamma}$. Now, $mns_\tau(abcd) = 5/35$, $bmns_\tau(abcd) = 20/35$, $sup(abcd) = 5/35$ and $mxs_\tau(abcd) = 2/35$. Although $\gamma * mns_\tau(abcd) = 3.5/35 = 0.1$ belongs to the interval $[2/35, 5/35)$, $\gamma * bmns_\tau(abcd) = 15/35$ does not. ■

3.2 Minimal Generators of Bounded Support

In order to give a complete characterization for the set $RI_{\tau,\gamma}$, let us first introduce the following notation: for a set X in FC_τ, let $mxgs_{\tau,\gamma}(X)$ be the maximal support of those minimal generators that are included in X and are not more frequent than $sup(X)/\gamma$:

$$mxgs_{\tau,\gamma}(X) = \max(\{sup(Y) \mid Y \in FG_\tau, Y \subset X, \gamma * sup(Y) \leq sup(X)\} \cup \{0\}).$$

Note that $mxgs_{\tau,\gamma}(X)$ is either 0, or it is greater than or equal to $sup(X)$. We prove two propositions that explain how we can use this value in order to compute the set $RI_{\tau,\gamma}$ and how to find, given $X \in RI_{\tau,\gamma}$, a subset $X_0 \subset X$ such that $X_0 \rightarrow X \backslash X_0 \in RR_{\tau,\gamma}$.

Proposition 5 $RI_{\tau,\gamma} = \{X \in FC_\tau \mid \gamma * mxgs_{\tau,\gamma}(X) > mxs_\tau(X)\}$.

Proof. Let X be an arbitrary set in $RI_{\tau,\gamma}$ and take X_0 in FG_τ such that $X_0 \rightarrow X \backslash X_0 \in RR_{\tau,\gamma}$ and $X_0 \subset X$. We have, on one hand, $conf(X_0 \rightarrow X \backslash X_0) \geq \gamma$, and on the other hand, $conf(X_0 \rightarrow Z \backslash X_0) < \gamma$ for all $Z \in FC_\tau$ with $Z \supset X$. That is, $sup(X) \geq \gamma * sup(X_0) > sup(Z)$ for all $Z \in FC_\tau$ with $Z \supset X$. From the first inequality, we deduce that X_0 meets all the conditions in order to be considered for the computation of $mxgs_{\tau,\gamma}(X)$, and therefore, $mxgs_{\tau,\gamma}(X) \geq sup(X_0)$. From the second, we get $\gamma * sup(X_0) > mxs_\tau(X)$. We conclude that $\gamma * mxgs_{\tau,\gamma}(X) > mxs_\tau(X)$.

Algorithm 1 RR Generator

Input: support threshold τ, confidence threshold γ
$F_\tau = \{X \subseteq \mathcal{U} \mid sup(X) \geq \tau\}$
$FC_\tau = \{X \in F_\tau \mid \overline{X} = X\}$
$FG_\tau = \{X \in F_\tau \mid \forall Y \subset X, sup(Y) > sup(X)\}$
for all $X \in FG_\tau$ **do**
  $mns_\tau(X) = \min(\{sup(Y) \mid Y \in FG_\tau, Y \subset X\} \cup \{\infty\})$
end for
$RI_{\tau,\gamma} = \emptyset$
for all $X \in FC_\tau \backslash \{\emptyset\}$ **do**
  $mxs_\tau(X) = \max(\{sup(Z) \mid Z \in FC_\tau, Z \supset X\} \cup \{0\})$
  $mxgs_{\tau,\gamma}(X) = \max(\{sup(Y) \mid Y \in FG_\tau, Y \subset X, \gamma * sup(Y) \leq sup(X)\} \cup \{0\})$
  if $\gamma * mxgs_{\tau,\gamma}(X) > mxs_\tau(X)$ **then**
    add X to $RI_{\tau,\gamma}$
  end if
end for
for all $X \in RI_{\tau,\gamma}$ **do**
  $c_1 = mxs_\tau(X)/\gamma$
  $c_2 = sup(X)/\gamma$
  Ant = $\{X_0 \in FG_\tau \mid X_0 \subset X, c_1 < sup(X_0) \leq c_2 < mns_\tau(X_0)\}$
  for all $X_0 \in$ Ant **do**
    output $X_0 \rightarrow X \backslash X_0$
  end for
end for

Conversely, let $X \in FC_\tau$ be such that $\gamma * mxgs_{\tau,\gamma}(X) > mxs_\tau(X)$. Clearly, $mxgs_{\tau,\gamma}(X)$ cannot be 0 (since $mxs_\tau(X) \geq 0$), so $\{Y \in FG_\tau \mid Y \subset X, \gamma * sup(Y) \leq sup(X)\}$ is not empty. Take $X_0 \in FG_\tau$ to be a set of maximal support that satisfies $X_0 \subset X$ and $\gamma * sup(X_0) \leq sup(X)$. Therefore, $mxgs_{\tau,\gamma}(X) = sup(X_0)$. Since $sup(X_0 \rightarrow X \backslash X_0) = sup(X) \geq \tau$ and $conf(X_0 \rightarrow X \backslash X_0) = \frac{sup(X)}{sup(X_0)} \geq \gamma$ we deduce that $X_0 \rightarrow X \backslash X_0 \in AR_{\tau,\gamma}$. Note that for any $Z \supset X$, $conf(X_0 \rightarrow Z \backslash X_0) = \frac{sup(Z)}{sup(X_0)} \leq \frac{mxs_\tau(X)}{sup(X_0)} = \frac{mxs_\tau(X)}{mxgs_{\tau,\gamma}(X)} < \gamma$. Moreover, for any $X_0' \subset X_0$, $sup(X_0') > sup(X_0)$ (since $X_0 \in FG_\tau$) and $\gamma * sup(X_0') > sup(X)$ (due to the choice we have made for X_0). This is why $conf(X_0' \rightarrow X \backslash X_0') = \frac{sup(X)}{sup(X_0')} < \gamma$. We conclude that $X_0 \rightarrow X \backslash X_0 \in RR_{\tau,\gamma}$ and $X \in RI_{\tau,\gamma}$. ■

Proposition 6 *Let $X \in RI_{\tau,\gamma}$, $c_1 = mxs_\tau(X)/\gamma$, $c_2 = sup(X)/\gamma$ and $X_0 \subset X$. Then $X_0 \rightarrow X \backslash X_0 \in RR_{\tau,\gamma}$ if and only if $c_1 < sup(X_0) \leq c_2 < mns_\tau(X_0)$.*

Proof. Consider $X \in RI_{\tau,\gamma}$ and $X_0 \subset X$. Clearly, $X_0 \rightarrow X \backslash X_0 \in RR_{\tau,\gamma}$ if and only if the rule $X_0 \rightarrow X \backslash X_0$ is in $AR_{\tau,\gamma}$ and does not belong to the cover set of any other rule in $AR_{\tau,\gamma}$. That is equivalent to: $sup(X) \geq \tau$, $\frac{sup(X)}{sup(X_0)} \geq \gamma$, $\frac{sup(X)}{sup(X_0')} < \gamma$ for all $X_0' \subset X$ and $\frac{sup(Z)}{sup(X_0)} < \gamma$ for all $Z \supset X$ that satisfy $sup(Z) \geq \tau$.

Now, it is easy to see that:

- $sup(X) \geq \tau$ always holds because $X \in FC_\tau$,
- $\frac{sup(X)}{sup(X_0)} \geq \gamma \Leftrightarrow sup(X_0) \leq c_2$,
- $\forall X'_0 \subset X_0 : \frac{sup(X)}{sup(X'_0)} < \gamma \Leftrightarrow \frac{sup(X)}{mns_\tau(X_0)} < \gamma \Leftrightarrow c_2 < mns_\tau(X_0)$,
- $\forall Z \supset X : \left(Z \in F_\tau \Rightarrow \frac{sup(Z)}{sup(X_0)} < \gamma\right) \Leftrightarrow \frac{mxs_\tau(X)}{sup(X_0)} < \gamma \Leftrightarrow c_1 < sup(X_0)$,

which concludes the proof. ■

Example 4 *Considering again Example 1, simple arithmetic suffices to check that Proposition 5 identifies exactly the closed sets from which representative rules follow as per Example 2; likewise, Proposition 6 can be illustrated with the representative rule* $b \rightarrow acde$ *of Example 3, which is obtained from* $abcde$ *(for which indeed* $0.4 * 5/6 > 1/6$ *as per Proposition 5) using* $c_1 = 2.5/6$ *and* $c_2 = 5/6$*, as* $c_1 < 4/6 \leq c_2 < 6/6$.

The correctness of Algorithm 1 trivially follows from Proposition 5 and Proposition 6.

3.3 An Algorithm for Different Confidence Thresholds

The disadvantage of Algorithm 1, compared to the one in Kryszkiewicz (2001), is that, for a given X in FC_τ, $mxgs_{\tau,\gamma}(X)$ depends on the confidence threshold, and hence it cannot be reused once γ has changed, whereas both $mxs_\tau(X)$ and $mns_\tau(X)$ can be computed only once for a given value of τ and then used for different confidence values. On the other hand, this one is guaranteed not to lose representative rules, whereas the one in Kryszkiewicz (2001) risks giving incomplete output, as in our counterexample above.

Algorithm 2 RR Generator - preprocessing phase

Input: support threshold τ
$F_\tau = \{X \subseteq \mathcal{U} \mid sup(X) \geq \tau\}$
$FC_\tau = \{X \in F_\tau \mid \overline{X} = X\}$
$FG_\tau = \{X \in F_\tau \mid \forall Y \subset X, sup(Y) > sup(X)\}$
for all $X \in FG_\tau$ **do**
  $mns_\tau(X) = \min(\{sup(Y) \mid Y \in FG_\tau, Y \subset X\} \cup \{\infty\})$
end for
for all $X \in FC_\tau \backslash \{\emptyset\}$ **do**
  $mxs_\tau(X) = \max(\{sup(Z) \mid Z \in FC_\tau, Z \supset X\} \cup \{0\})$
  $n[X] = |\{Y \in FG_\tau \mid Y \subset X\}|$
  let $\{Y_1, \ldots, Y_{n[X]}\}$ be the set $\{Y \in FG_\tau \mid Y \subset X\}$ in descending order of support
  for all $i \in \{1, \ldots, n[X]\}$ **do**
    $y_i[X] = sup(Y_i)$
    $p_i[X] = sup(X)/y_i[X]$
  end for
  $p_0[X] = 0$
end for

Instead of computing $mxgs_{\tau,\gamma}(X)$ for each and every γ, one can find the individual points of the interval $(0, 1]$ where $mxgs_{\tau,\gamma}(X)$ changes its value. Indeed, given $X \in FC_\tau \backslash \{\emptyset\}$, let $\{Y_1, \ldots, Y_{n[X]}\}$ be the set $\{Y \in FG_\tau \mid Y \subset X\}$ in descending order of support. It is easy to see that

$$mxgs_{\tau,\gamma}(X) = \begin{cases} sup(Y_1), & \text{if } \gamma \leq \frac{sup(X)}{sup(Y_1)} \\ sup(Y_{i+1}), & \text{if } \gamma \in \left(\frac{sup(X)}{sup(Y_i)}, \frac{sup(X)}{sup(Y_{i+1})}\right], i \in \{1, \ldots, n[X]-1\} \\ 0, & \text{if } \gamma > \frac{sup(X)}{sup(Y_{n[X]})}. \end{cases}$$

Now, each time a new value of the confidence threshold γ is given, one can decide whether a frequent closed set X is in $RI_{\tau,\gamma}$ by simply retrieving the interval $(p_i[X], p_{i+1}[X]]$ with $i \in \{0, \ldots, n[X]-1\}$ to which γ belongs (recall that in this case $mxgs_{\tau,\gamma}(X) = y_{i+1}[X]$) and then checking whether the inequality $\gamma * y_{i+1}[X] > mxs_\tau(X)$ holds. Note that if no such i exists (that is, whenever γ has a value strictly greater than $p_{n[X]}[X]$), $mxgs_{\tau,\gamma}(X)$ takes the value 0, which makes $\gamma * mxgs_{\tau,\gamma}(X)$ smaller than or equal to $mxs_\tau(X)$.

These ideas are implemented in Algorithms 2 and 3.

Algorithm 3 RR Generator - second phase

```
Input: support threshold τ, confidence threshold γ
RI_{τ,γ} = ∅
for all X ∈ FC_τ\{∅} do
    if ∃i ∈ {0,...,n[X]−1} such that γ ∈ (p_i[X], p_{i+1}[X]] then
        if γ * y_{i+1}[X] > mxs_τ(X) then
            add X to RI_{τ,γ}
        end if
    end if
end for
for all X ∈ RI_{τ,γ} do
    c_1 = mxs_τ(X)/γ
    c_2 = sup(X)/γ
    Ant = {X_0 ∈ FG_τ | X_0 ⊂ X, c_1 < sup(X_0) ≤ c_2 < mns_τ(X_0)}
    for all X_0 ∈ Ant do
        output X_0 → X\X_0
    end for
end for
```

4 Empirical Comparison

We have seen that one can find toy examples of datasets in which the output of the algorithm in Kryszkiewicz (2001) is incomplete. We have tested the algorithm on two real-world datasets: a typical market basket dataset, taken from the data mining workbench Clementine (2005), and the training set part of the UCI Adult US census dataset; see Asuncion and Newman (2007).

We have implemented three different algorithms: one for the incomplete heuristic given in Kryszkiewicz (2001), one for the first heuristic proposed by us in which mns_τ is replaced

by $bmns_\tau$ (also incomplete), and one that generates the complete set of representative rules as described by Algorithm 1. In order to get comparable results, all of them allow rules with empty antecedent and use the same definition of frequent sets and association rules as given in our Preliminaries.

The first dataset under study consists of 1000 transactions over 15 attributes, 11 of them reflecting the type of product that a customer could have purchased (fruitveg, freshmeat, dairy, cannedveg, cannedmeat, frozenmeal, beer, wine, softdrink, fish, confectionery) and 4 others given by the gender and the home ownership status of the client (male, female, homeowner, donotownhome).

Table 2 shows the number of representative rules obtained for different support and confidence thresholds (the third column), as well as the cardinality of the output set when $bmns_\tau$ or mns_τ is used (the fourth and fifth column, respectively). We can see that although for higher support thresholds the output of the algorithms is, most of the times, identical (recall that the output of the algorithm in Kryszkiewicz (2001) is always a subset of the whole set of representative rules), lowering both thresholds shows bigger differences. For comparison, the rightmost column provides the number of rules in the standard sense of Agrawal et al. (1996).

TAB. 2 – Market Basket Dataset (number of rules)

τ	γ	RR	RR with $bmns_\tau$	RR with mns_τ	Standard
0.05	0.7	41	33	33	67
	0.8	17	16	16	36
	0.9	15	15	15	15
0.10	0.7	12	10	10	21
	0.8	5	5	5	12
	0.9	4	4	4	4
0.15	0.7	6	6	6	16
	0.8	2	2	2	2
	0.9	0	0	0	0

As an example, in the case the thresholds for support and confidence are 0.10 and 0.70, respectively, there are a total of 12 representative rules, among which two are lost when using mns or $bmns$ (listed in bold):

[c:0.70,s:0.14] male frozenmeal ⇒ beer, **[c:0.72,s:0.15] male frozenmeal ⇒ cannedveg**,
[c:0.86,s:0.12] confectionery wine ⇒ female, [c:0.70,s:0.14] male cannedveg ⇒ beer frozenmeal,
[c:0.82,s:0.14] beer frozenmeal ⇒ male cannedveg, [c:0.84,s:0.14] beer cannedveg ⇒ male frozenmeal,
[c:0.71,s:0.14] male beer ⇒ cannedveg frozenmeal, [c:0.81,s:0.14] cannedveg frozenmeal ⇒ male beer,
[c:0.73,s:0.10] male fish ⇒ donotownhome, [c:0.89,s:0.12] fish fruitveg ⇒ donotownhome,
[c:0.70,s:0.10] donotownhome beer ⇒ male, [c:0.70,s:0.11] donotownhome frozenmeal ⇒ male

Dataset ADULT is a transactional version of the training set part of the UCI census dataset Adult US, see Asuncion and Newman (2007); it consists of 32561 transactions over 269 items. Note that in this case there are significant differences between the output of the algorithm in Kryszkiewicz (2001) and the set of all representative rules (Table 3). For example, for support and confidence thresholds of 0.05 and 0.8, respectively, more than half of the rules are lost.

We have run the experiments on an Intel Core 2CPU 6300 @ 1.86GHz machine with 2 GB of RAM running under Microsoft Windows XP Professional. The running time of all three

TAB. 3 – Adult US census dataset (number of rules)

τ	γ	RR	RR with $bmns_\tau$	RR with mns_τ	Standard
0.05	0.6	872	383	383	3443
	0.7	781	425	425	2926
	0.8	851	640	640	2426
0.10	0.6	326	124	124	1284
	0.7	274	162	162	1083
	0.8	345	270	270	923

algorithms were between 15 and 47 milliseconds in the case of the market basked dataset and between 62 and 1203 milliseconds for the Adult dataset. The algorithm that correctly outputs all representative rules is slightly slower than the other two but, in our tests, the difference was rather irrelevant since the time needed to print the results on screen (a device slower than the CPU) still dominates the process.

It must be noted that the quantity of representative rules may decrease at lower confidence or support thresholds. This phenomenon has been observed and explained before, see Balcázar (2010), and is caused by powerful rules of a given confidence, say 0.8, that are filtered out at higher thresholds, leaving therefore many other rules as representative, but that force all of these out of the representative rules as they become redundant when the confidence threshold gets below 0.8 and lets the powerful rule in.

5 Perspectives

As future research topics, we wish to extend the characterization given in Proposition 5 of all closed itemsets that can be decomposed into representative rules to the stronger notion of redundancy introduced in Balcázar (2010), namely the closure-based redundancy. Additionally, a puzzling fact that we plan to study further is that, in many of the real-world datasets we have run our algorithms on, our first alternative from Subsection 3.1, also incomplete, gives the same quantity of representative rules as the original incomplete algorithm; this may indicate that further understanding of the sets of rules obtained by these incomplete algorithms might be useful.

References

Aggarwal, C. C. and P. S. Yu (2001). A new approach to online generation of association rules. *IEEE Transactions on Knowledge and Data Engineering 13*(4), 527–540.

Agrawal, R., H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo (1996). Fast discovery of association rules. In *Advances in Knowledge Discovery and Data Mining*, pp. 307–328. AAAI/MIT Press.

Asuncion, A. and D. Newman (2007). UCI machine learning repository.

Balcázar, J. L. (2010). Redundancy, deduction schemes, and minimum-size bases for association rules. *Logical Methods in Computer Science 6*(2:3), 1–33.

Ceglar, A. and J. F. Roddick (2006). Association mining. *ACM Comput. Surv. 38*(2).

Clementine (2005). Clementine 10.0 desktop user guide.

Cristofor, L. and D. A. Simovici (2002). Generating an informative cover for association rules. In *Proc. of the 2002 IEEE International Conference on Data Mining (ICDM)*, pp. 597–600. IEEE Computer Society.

Hamrouni, T., S. Ben Yahia, and E. Mephu Nguifo (2008). Succinct minimal generators: Theoretical foundations and applications. *Int. J. Found. Comput. Sci. 19*(2), 271–296.

Kryszkiewicz, M. (1998a). Fast discovery of representative association rules. In L. Polkowski and A. Skowron (Eds.), *Proc. of the 1st International Conference on Rough Sets and Current Trends in Computing (RSCTC)*, Volume 1424 of *Lecture Notes in Artificial Intelligence*, pp. 214–221. Springer-Verlag.

Kryszkiewicz, M. (1998b). Representative association rules. In X. Wu, K. Ramamohanarao, and K. B. Korb (Eds.), *Proc. of the 2nd Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, Volume 1394 of *Lecture Notes in Artificial Intelligence*, pp. 198–209. Springer-Verlag.

Kryszkiewicz, M. (2001). Closed set based discovery of representative association rules. In F. Hoffmann, D. J. Hand, N. M. Adams, D. H. Fisher, and G. Guimarães (Eds.), *Proc. of the 4th International Symposium on Intelligent Data Analysis (IDA)*, Volume 2189 of *Lecture Notes in Computer Science*, pp. 350–359. Springer-Verlag.

Kryszkiewicz, M. (2002). Concise representations of association rules. In D. J. Hand, N. M. Adams, and R. J. Bolton (Eds.), *Proc. of the ESF Exploratory Workshop on Pattern Detection and Discovery*, Volume 2447 of *Lecture Notes in Computer Science*, pp. 92–109. Springer-Verlag.

Luxenburger, M. (1991). Implications partielles dans un contexte. *Mathématiques et Sciences Humaines 29*, 35–55.

Pasquier, N., R. Taouil, Y. Bastide, G. Stumme, and L. Lakhal (2005). Generating a condensed representation for association rules. *J. Intell. Inf. Syst. 24*(1), 29–60.

Phan-Luong, V. (2001). The representative basis for association rules. In N. Cercone, T. Y. Lin, and X. Wu (Eds.), *ICDM*, pp. 639–640. IEEE Computer Society.

Zaki, M. J. (2004). Mining non-redundant association rules. *Data Min. Knowl. Discov. 9*(3), 223–248.

Résumé

La sortie d'un mineur de règles d'association est souvent énorme dans la pratique. C'est pourquoi plusieurs représentations concises sans perte ont été proposées, telles que les règles "essentielles" ou "représentatives". Nous reviendrons sur l'algorithme donné par Kryszkiewicz (Int. Symp. Intelligent Data Analysis 2001, Springer-Verlag LNCS 2189, 350–359) pour l'extraction des règles représentatives. Nous montrons que sa production est parfois incomplète, à cause d'une manque à la preuve mathématique de validité de cet algorithme, et nous proposons un générateur de remplacement complet avec presque les mêmes temps d'éxécution.