Knowledge Redundancy Approach to Reduce Size in Association Rules

Julio César Díaz Vera
University of Informatics Sciences, Havana, Cuba
E-mail: jcdiaz@uci.cu

Guillermo Manuel Negrín Ortiz
University of Informatics Sciences, Havana, Cuba
E-mail: gmnegrin@uci.cu

Carlos Molina
University of Jaen, Jaen, Spain
E-mail: carlosmo@ujaen.es

Maria Amparo Vila
University of Granada, Granada, Spain
E-mail:vila@decsai.ugr.es

Keywords: association rule mining, redundant rules, knowledge guided post-processing

Received: June 19, 2019

Association Rules Mining is one of the most studied and widely applied fields in Data Mining. However, the discovered models usually result in a very large set of rules; so the analysis capability, from the user point of view, is diminishing. Hence, it is difficult to use the found model in order to assist in the decision-making process. The previous handicap is hightened in the presence of redundant rules in the final set. In this work, a new definition of redundancy in association rules is proposed, based on user prior knowledge. A post-processing method is developed to eliminate this kind of redundancy, using association rules known by the user. Our proposal allows finding more compact models of association rules to ease its use in the decision-making process. The developed experiments have shown reduction levels that exceed 90 percent of all generated rules, using prior knowledge always below ten percent. So, our method improves the efficiency of association rules mining and the exploitation of discovered association rules.

Povzetek: Opisan je sistem za zmanjševanje števila in dolžine pravil s pomočjo analize redundantnosti za metode asociativnega učenja.

1 Introduction

Mining for association rules has been one of the most studied fields in data mining. Its main goal is to find unknown relations among items in a database.

Given a set of items I which contains all the items in the domain and a transactional database D where every transaction is composed by a transaction id (tid) and a set of items, subset of I (itemset).

An association rule is presented as an implication $X \rightarrow Y$ where X is the antecedent and Y is the consequent of the rule. Both X and Y are itemsets and usually, but not necessarily, they check $X \cap Y = \emptyset$ property. Association rules reflect how much the presence of the rule antecedent influences the presence of the rule consequent in the database records.

What generally makes a rule meaningful are two statistical factors: support and confidence. The support of a rule $supp(X \rightarrow Y)$ refers to the portion of the database transaction for which $X \cap Y$ is true while confidence $conf(X \rightarrow Y)$ is a measure of certainty to evaluate the validity of the rule, it is a measure for the portion of record which contains Y from those that contain X. The problem with association rule mining deals with finding all the rules that satisfy a user-given threshold for support and confidence. Most algorithms face the challenge in a two steps procedure

1. Find all the itemsets which support value is equal or greater than the support threshold.

2. Generate all association rules $X \rightarrow (Y - X)$, considering: Y is a frequent itemset, $X \subset Y$, and $conf(X \rightarrow Y)$ is equal or greater than the confidence threshold value.

The discovering of meaningful association rules can help in the decision-making process but the quite large number of rules usually makes it difficult for decision-makers in order to process, interpret and apply them. A significant part of the rules presented to the user are irrelevant because they are obvious, too general, too specific or because they are not relevant for the decision topic. Several methods were proposed in the literature to overcome this handicap such as interest measures development, concise representations of frequent itemsets and redundancy reduction. Section 2 discusses some of the most important works in the field.

This paper proposes a new approach to deal with redundancy, taking into account user previous knowledge about the studied domain. Previous knowledge is used to detect and prune redundant rules. We adapt the concept of redundancy and we propose a procedure to develop the redundancy reduction process in the post-processing stage.

The paper is organized as follows. Section 2 discusses related work. In section 3 we propose an algorithm to find and prune redundant rules. In section 4 the proposed algorithm is used over three datasets one with data about financial investment [1], other with data about the USA census [2] and the other with data about Mushrooms [2]. Section 5 closes the paper with conclusions.

2 Related work

Interestingness is difficult to define quantitatively [3] but most interestingness measures are classified in objective measures and subjective measures. Objective measures are domain-independent, one of them is the interestingness which is expressed in terms of statistic or information theory applied over the database. Several surveys [4, 5, 6] summarize and compare objective measures. The explosion of objective measures has raised a new problem: What are the best metrics to use in a specific situation and a particular application field? Several papers attempt to solve it [8, 9] but it is far from being solved. The correlation between 11 objective rule interestingness measures and real human interest over eight different datasets were computed in [10] and there was not a clear "winner", the correlation values associated with each measure varied considerably across the eight datasets.

Subjective measures were proposed in order to involve explicitly user knowledge in the selection of interesting rules so that the user can make a better selection. According to [11] subjective measures are classified in:

- Unexpectedness: a pattern is interesting if it is surprising to the user.
- Actionability: a pattern is interesting if it can help the user to take some actions.

Actionability started as an abstract notion, with an unclear definition, but nowadays, several researchers are interested in it. The actionability problem is discussed in [12].

Unexpectedness or novelty [13] was proposed in order to solve the pattern triviality problem, assessing the surprise level of the discovered rules. Several techniques have been used to accomplish this aim:

- Templates: Templates are syntactic constraints that allow the user to define a group of rules that are interesting or not to him/her [14, 15]. A template is defined as $A_1 ... A_n \rightarrow A_{n+1}$ where A_i is a class name in a hierarchy or an expression E over a class name. Templates may be inclusive or restrictive. A rule is considered interesting if it matches an inclusive template and uninteresting if it matches a restrictive template. The use of templates is quite restrictive because the matching method requires each rule element to be an instance of the elements in templates, and all template elements must have at least one instance in the rule. Moreover, the template definition makes hard to use it for declaring restrictive templates because it should be composed of elements subsuming all attributes of the rule, being in a subsuming relation with the inclusive template elements.

  The best known form of templates is meta-rules [16, 40] a meta-rule is the relationship between two association rules. The main drawback of this approach is that meta-rules are restricted to having a single rule in their antecedent and consequent, because of this some important information may be lost.

- Belief: Silbershatz and Tuzilin [11] defined user knowledge as a set of convictions, denominated belief. They are used in order to measure the unexpectedness of a pattern. Each belief is defined as a predicate formula expressed in first-order logic with a degree of confidence associated, measuring how much the user trusts in the belief. Two types of belief were defined:

  - Soft belief is that knowledge user accepts to change if new evidence contradicts the previous one. The interestingness of the new pattern is computed by how the new pattern changes the degree of beliefs.
  - Hard belief is that knowledge user will not change whatever new patterns are extracted. They are constraints that cannot be changed with new evidence.

  This approach is still in a development stage, no further advances were published, so it is not functional.

- General Impressions: were presented in [17] and later developed in [18] and [19]. They developed a specification language to express expectations and goals. Three levels of specification were established: General Impressions, Reasonably Precise Concept and Precise Knowledge. Item taxonomies concept was integrated in the specification languages in order to generalize rule selection. The matching process involved a syntactic comparison between antecedent/-consequent elements. Thus, each element in the general impression should find a correspondent in the association rule.

- Logical Contradiction: was developed in [20]. It consists in extracting only those patterns which logically contradict the consequent of the corresponding belief.

An association rule $X \rightarrow Y$ is unexpected with respect to some belief $A \rightarrow B$ if:

- $Y \wedge B \models FALSE$ B and Y are in logical contradiction;
- $X \wedge B$ has an important support in the database. This condition eliminates those rules which could be considered unexpected, but not those concerning the same transaction in the database;
- $A, X \rightarrow B$ exists.

- Preference Model: was proposed in [21]. It is a specific type of user knowledge representing how the basic knowledge of the user, called knowledge rules (K), will be applied over a given scenario or tuples of the database. The user proposes a covering knowledge (Ct) for each tuple (t) - a subset of the knowledge rule set K that the user prefers to apply to the tuple t. The approach validates the transactions which satisfy the extracted rule.

All the previously presented works use some kind of knowledge to reduce the number of useless association rules in the final set. In this way, our approach is similar to them but there are some remarkable differences.

Like in templates our approach uses the syntactical notation of association rules to represent knowledge. Templates use this knowledge to constraint the structure of selected rules, pruning out those rules which do not satisfy the template but produce a lot of association rules with similar information. On the other hand, we use the knowledge to remove those rules with similar information, presenting to the user a set of unexpected rules that can help him to better understand the underlying domain.

The approach followed by Belief tries to find just unknown rules, this is our main goal too but, they use a complex and fixed formal knowledge representation based on first order logic and degrees of belief with no clear way of building and maintaining the belief system. Instead, we use a simpler and natural rule-based form of knowledge, focused on the enhanced capability to increase interactively the knowledge system.

2.1 Rule redundancy reduction

Research community accepts the semantical definition of association rule redundancy given in [22] "an association rule is redundant if it conveys the same information - or less general information - than the information conveyed by another rule of the same usefulness and the same relevance". But several formal definitions have been proposed over time. In table 1, a sample transactional database is presented. Defining a support threshold of 0.15 and a confidence threshold of 0.75, an association rule model with 92 rules is obtained. It is used to show redundancy definitions.

Income	Balance	Sex	Unemployed	Loan
High	High	F	No	Yes
High	High	M	No	Yes
Low	Low	M	No	No
Low	High	F	Yes	Yes
Low	High	M	Yes	Yes
Low	Low	F	Yes	No
High	Low	M	No	Yes
High	Low	F	Yes	Yes
Low	Medium	M	Yes	No
High	Medium	M	No	Yes
Low	Medium	F	Yes	No
Low	Medium	M	No	No

Table 1: Sample transactions

Definition 1. ***Minimal non-redundant association rules[22]:*** *An association rule $R : X \rightarrow Y$ is a minimal non-redundant association rule if there is not an association rule $R_1 : X_1 \rightarrow Y_1$ with:*

- $support(R) = support(R_1)$
- $confidence(R) = confidence(R_1)$
- $X_1 \subseteq X$ *and* $Y \subseteq Y_1$

From data on table 1 we can obtain the rules: $R : \{[balance].[medium]\} \rightarrow \{[income].[low], [loan].[no]\}$ $supp = 0.25$, $conf = 0.75$ and $R_1 : \{[balance].[medium]\} \rightarrow \{[loan].[no]\}$ $supp = 0.25$, $conf = 0.75$. According to definition 1 R is a redundant rule. No new information is provided by its inclusion into the association rules model.

Several works have been developed to prune that kind of redundancy. Mining Closed Associations, uses frequent closed itemsets [23] tries to produce the set of minimal generators for each itemset. The number of closed association rules is linear to the number of closed frequent itemsets. It can be large for sparse and large datasets.

The Generic Basis (GB) and the Informative Basis (IB) [22] used the Galois connections to propose two condensed basics that represent non-redundant rules. The Gen-GB and Gen-RI algorithms were presented to obtain a generic basis and a transitive reduction of the IB. The reduction ratio of IB was improved by [24] maximal closed itemsets. The Informative Generic Basis [25] also uses the Galois connection semantics but taking the support of all frequent itemsets as an entry, so it can calculate the support and confidence of derived rules. The augmented Iceberg Galois lattice was used to construct the Minimal Generic Basis (MGB) [26]. The concept of generator was incorporated into high utility itemsets mining in [27].

The redundancy definition presented in definition 1 requires that a redundant rule and its corresponding non-redundant rule must have identical confidence and identical support. From data on table 1 we can obtain the

rules:
$R : \{[income].[high], [unemployed].[no]\} \rightarrow \{[loan].[yes]\}$ $supp = 0.33, conf = 1.0$, and $R_1 : \{[income].[high]\} \rightarrow \{[loan].[yes]\}$ $supp = 0.41, conf = 1.0$ those rules are non-redundant ones, but the consequent of R can be obtained from R_1 a rule with the same confidence and fewer conditions. So without R the same results are achieved, rule R must be a redundant rule. Xu [28] formalizes this kind of redundancy in definition 2.

Definition 2. ***Redundant rules[28]:*** *Let $X \rightarrow Y$ and $X_1 \rightarrow Y_1$ be two association rules with confidence cf and cf_1, respectively. $X \rightarrow Y$ is said to be a redundant rule to $X_1 \rightarrow Y_1$ if*

- *$X_1 \subseteq X$ and $Y \subseteq Y_1$*

- *$cf \leq cf_1$*

Based on definition 2 the Reliable basis was proposed. It consists of two bases the ReliableApprox used in partial rules, and ReliableExact used in exact rules. Frequent closed itemsets are used to perform the reliable redundancy reduction process. It generates rules with minimal antecedent and maximal consequent. The reliable basis removes a great amount of redundancy without reducing the inference capacity of the remaining rules. Phan [29] uses a more radical approach to define redundancy see definition 3.

Definition 3. ***Representative association rules[29]:*** *Let $X \rightarrow Y$ an association rule. $X \rightarrow Y$ is said to be a representative association rule if there is not other interesting rule $X_1 \rightarrow Y_1$ such that $X_1 \subseteq X$ and $Y \subseteq Y_1$.*

The redundancy definitions presented above do not guarantee the exclusion of all non-interesting patterns of the final model. Example 1 shows a group of rules with no new information to the user, and they are not classified as redundant by the previous definitions.

Example 1. ***A set of redundant rules from data in table 1*** *Let's see a subset of association rules obtained from table 1:*

$R_1 : \{[income].[high]\} \rightarrow \{[loan].[yes]\}$
$R_2 : \{[sex].[female], [unemployed].[no]\} \rightarrow \{[income].[high]\}$
$R_3 : \{[sex].[female], [unemployed].[no]\} \rightarrow \{[income].[high], [loan].[yes]\}$
$R_4 : \{[sex].[female], [unemployed].[no]\} \rightarrow \{[loan].[yes]\}$
$R_5 : \{[income].[high], [loan].[yes]\} \rightarrow \{[unemployed].[no]\}$
$R_6 : \{[income].[high], [loan].[yes], [sex].[male]\} \rightarrow \{[unemployed].[no]\}$
$R_7 : \{[balance].[high], [income].[high], [loan].[yes]\} \rightarrow \{[unemployed].[no]\}$

If we analyze the rules R_1 and R_3 we see that item [loan].[yes] in R_3 consequent provides no new information, because this is known by R_1. So rule R_3 is redundant but this kind of redundancy is not detected by the previous definitions. Analyzing rules R_1, R_2 and R_4 we can check that combining, transitively, of R_1 and R_2 it will produce R_4 so, R_4 is redundant. One more time this kind of redundancy is not detected by previous definitions. In R_5,R_6 and R_7 antecedent the item $[loan].[yes]$ provides no new information because this is known by R_1. It is redundant and must be pruned, but it can not be detected by redundancy definitions.

2.2 Post-processing

Since the year 2000, the interest in post-processing methods in association rules has been increasing. Perhaps the most accurate definition of post-processing tasks were done by Baesens et al. [30] Post-processing consists of different techniques that can be used independently or together: pruning, summarizing, grouping and visualization. We have a special interest in pruning techniques that prune those rules that do not match to the user knowledge. Those techniques are associated with interestingness measures that may not satisfy the downward closure property, so it is impossible to integrate them in Apriori like extraction algorithms.

An element to consider is the nature of Knowledge Discovery in Databases (KDD) as an interactive and iterative user-centered process. Enforcing constraints during the mining runs neglects the character of KDD [31], [32]. A single and possibly expensive mining run is accepted but all subsequent mining questions are supposed to be satisfied with the initial result set.

In this work, a method is developed to obtain non-redundant association rules about user knowledge. It is important to ensure the user capability to refine his/her knowledge in an interactive and iterative way, accepting any of the discovered associations or discarding some previous associations and updating prior knowledge. This approach also makes possible to fulfill the mining question of different users, with different domain knowledge, in a single mining run.

3 A knowledge guided approach

3.1 Knowledge based redundancy

In example 1, a group of redundant rules, which are currently not covered by the definitions of redundancy, are showed. Our interest is to eliminate these forms of redundancy in association rule models. Based on a core set of rules that represent the user belief; a result of his experience working in the subject area. This knowledge is more general than rules obtained in the mining process which only represent a particular dataset with partial information so the quality metric value for this kind of rule is considered maximal. This set of rules will be named prior knowledge. A rule that does not contradict prior knowledge of the user will be considered redundant. We formalize the notion of prior knowledge redundancy in definition 4. User can represent previous knowledge in different ways like semantic networks, ontologies, among others.

Considering that, the expert is interested in association rules discovering, prior knowledge is incorporated to the model using association rules format. For example an expert working with the dataset presented in table 1 knows that customers with high income ($[income].[high]$) pay their loans on time and therefore these must be approved. This knowledge can be represented as the association rule $\{[income].[high]\} \rightarrow \{[loan].[yes]\}$.

Definition 4. ***Knowledge Based Redundancy:*** *Let $\mathcal{S}$ be a set of association rules and $\mathcal{S}_c$ a set of prior known rules, defined over the same domain of $\mathcal{S}$. An association rule $R : X \rightarrow Y \in \mathcal{S}$ is redundant with respect to $\mathcal{S}_c$ if there is a rule $R^{'} : X^{'} \rightarrow Y^{'} \in \mathcal{S}_c$ and fulfills some of the following conditions.*

1. $X^{'} \subseteq X \wedge Y^{'} \cap Y \neq \{\emptyset\}$
   A rule is redundant if there is another rule presented in $\mathcal{S}_c$ that contains more general information.

2. $X^{'} \subseteq X \wedge \exists R^{''} : X^{''} \rightarrow Y^{''} \in \mathcal{S}_c : X^{''} \subseteq Y^{'} \wedge Y \subseteq Y^{''}$
   A rule R is redundant if there is a rule $R^{'}$ in $\mathcal{S}_c$ that contains part or the whole antecedent and there is a third rule $R^{''}$ in $\mathcal{S}_c$ that shares information with $R^{'}$ and its consequent contains R consequent.

3. $X^{'} \subseteq X \wedge Y^{'} \cap X \neq \{\emptyset\}$
   A rule is redundant if its antecedent contains a part or the whole information of a previously known rule.

4. $X^{'} \subseteq Y \wedge Y^{'} \cap Y \neq \{\emptyset\}$
   A rule is redundant if its consequent contains a part or the whole information of a previously known rule.

Reviewing rules in example 1 with definition 4 we have: $\mathcal{S}_c = \{\{[income].[high]\} \rightarrow \{[loan].[yes]\}$, $\{[sex].[female], [unemployed,].[no]\} \rightarrow \{[income].[high]\}\}$

Rule R_3 : $\{[sex].[female], [unemployed].[no]\} \rightarrow \{[income].[high], [loan].[yes]\}$ fulfills condition 1 in definition 4 because:

1. $[sex].[female], [unemployed].[no] \subseteq [sex].[female], [unemployed].[no]$

2. $[income].[high] \subseteq [income].[high], [loan].[yes]$

Rule R_3 : $\{[sex].[female], [unemployed].[no]\} \rightarrow \{[income].[high], [loan].[yes]\}$ fulfills condition 4 in definition 4 because:

1. $[income].[high] \subseteq [income].[high], [loan].[yes]$

2. $[loan].[yes] \subseteq [income].[high], [loan].[yes]$

Rule R_4 : $\{[sex].[female], [unemployed].[no]\} \rightarrow \{[loan].[yes]\}$ fulfills condition 2 in definition 4 because:

1. $[sex].[female], [unemployed].[no] \subseteq [sex].[female], [unemployed].[no]$

2. $[income].[high] \subseteq [income].[high]$

3. $[loan].[yes] \subseteq [loan].[yes]$

Rule R_5 : $\{[income].[high], [loan].[yes]\} \rightarrow \{[unemployed].[no]\}$ fulfills condition 3 in definition 4 because:

1. $[income].[high] \subseteq [income].[high], [loan].[yes]$

2. $[loan].[yes] \subseteq [income].[high], [loan].[yes]$

Rule R_6 : $\{[income].[high], [loan].[yes], [sex].[male]\} \rightarrow \{[unemployed].[no]\}$ fulfills condition 3 in definition 4 because:

1. $[income].[high] \subseteq [income].[high], [loan].[yes], [sex].[male]$

2. $[loan].[yes] \subseteq [income].[high], [loan].[yes], [sex].[male]$

Rule R_7 : $\{[balance].[high], [income].[high], [loan].[yes]\} \rightarrow \{[unemployed].[no]\}$ fulfills condition 3 in definition 4 because:

1. $[income].[high] \subseteq [balance].[high], [income].[high], [loan].[yes]$

2. $[loan].[yes] \subseteq [balance].[high], [income].[high], [loan].[yes]$

Armstrong's axioms [33] are a set of inference rules. They allow to obtain the minimum set of functional dependencies that are maintained in a database. The rest of functional dependencies can be derived from this set. They are part of clear mechanisms designed to find smaller subsets of a larger set of functional dependencies called "covers" that are equivalent to the "bases" in Closure Spaces and Data Mining.

Armstrong's axioms can not be used as an inference mechanism in association rules [34] because it is impossible to obtain the values of support and confidence in the derived rules:

- Reflexivity (if $B \subset A$ then $A \rightarrow B$) holds because $conf(A \rightarrow B) = \frac{supp(A \cap B)}{supp(A)} = \frac{supp(A)}{supp(A)} = 1$

- Transitivity if $A \rightarrow B$ and $B \rightarrow C$ both hold with confidence $\geq threshold$ we can not know the value for $conf(AD \rightarrow C)$ so the Transitivity does not hold.

- Augmentation (if $A \rightarrow B$ then $AC \rightarrow B$) does not hold. Enlarging the antecedent of a rule may give a rule with much smaller confidence, even zero: think of a case where most of the times X appears it comes with Z, but it only comes with Y when Z is not present; then the confidence of $X \to Z$ may be high whereas the confidence of $XY \to Z$ may be null.

Our intention is to use Armstrong's axioms in order to assess if a rule has Prior Knowledge Redundancy over a set of rules S_c from previous knowledge. So they must verify the condition presented in definition 4.

Condition $X^{'} \subseteq X \wedge Y^{'} \cap Y \neq \{\emptyset\}$ represents the classical definition of redundancy like in definition 1, definition 2 and definition 3. This condition is fulfilled if a single attribute in Y is redundant. Armstrong's axioms can be used to perform this operation. Let $R_1 : X \to Y$ and $R_2 : X^{'} \to Y^{'}$ be association rules. Suppose $Y^{'} \cap Y = Y_1$. Then by the reflexivity axiom on R_2 consequent $R_3 : Y \to Y_1$ and by reflexivity on R_1 consequent $R_4 : Y^{'} \to Y_1$. By transitivity between R_1 and R_3 we have $R_5 : X \to Y_1$, applying transitivity between R_2 and R_4 we have $R_6 : X^{'} \to Y_1$. $X^{'} \subseteq X$ by statement condition, applying augmentation in R_6 until $X^{'} = X$, $R_7 : X \to Y_1$. Therefore Armstrong's axioms check the condition. For example, the rule $R : \{[income].[high], [sex].[male]\} \to \{[loan].[yes], [unemployed].[no]\}$ is part of the association model generated from the dataset in table 1. This rule can be classified as redundant by condition 1 of definition 4 with respect to prior knowledge. $S_c = \{R_{s1} : [income].[high] \to [loan].[yes]$, $R_{s2} : [sex].[female], [unemployed].[no] \to [income].[high]\}$. By the application of Reflexivity, we have that $R_1 : [loan].[yes] \to [loan].[yes]$ by Augmentation of $[unemployed].[no]$ on R_1 we have $R_2 : [loan].[yes], [unemployed].[no] \to [loan].[yes]$ and by Transitivity between R and R_2 we have $R_3 : [income].[high], [sex].[male] \to [loan].[yes]$, the same procedure must be followed to $[unemployed].[no]$. Now by Augmentation of $[sex].[male]$ in rule $[income].[high] \to [loan].[yes] \in S_c$ we have $R_4 : [income].[high], [sex].[male] \to [loan].[yes]$ $R_4 = R_3$ so item $[loan].[yes]$ is redundant in R and therefore R is also redundant.

Condition $X^{'} \subseteq X \wedge \exists R^{''} : X^{''} \to Y^{''} \in \mathcal{S}_c : X^{''} \subseteq Y^{'} \wedge Y \subseteq Y^{''}$ represents the notion of transitivity a common term in human thinking. This condition is fulfilled if a single attribute in Y is redundant. Let $R_1 : X \to Y$, $R_2 : X^{'} \to Y^{'}$ and $R_3 : X^{''} \to Y^{''}$ be rules. Suppose $Y^{''} \cap Y = Y_1$. Then by the reflexivity axiom on R_1 consequent $R_4 : Y \to Y_1$ by transitivity between R_1 and R_4 we have $R_5 : X \to Y_1$. By statement condition $X^{''} \subseteq Y^{'}$ so by reflexivity on R_2 consequent we have $R_6 : Y^{'} \to X^{''}$. By transitivity between R_2 and R_6 we have $R_7 : X^{'} \to X^{''}$ now by transitivity between R_2 and R_7 we have $R_8 : X^{'} \to Y^{''}$. Applying augmentation in R_8 until we have $R_9 : X \to Y^{''}$. By reflexivity in R_9 consequent $R_{10} : Y \to Y_1$ and by transitivity between R_9 and R_{10} we have $R_{11} : X \to Y_1$. Therefore Armstrong's axioms check the condition. For example, taking into account rule $R : \{[sex].[female], [unemployed].[no]\} \to \{[loan].[yes]\}$ and prior knowledge $S_c = \{R_{s1} : [income].[high] \to [loan].[yes]$, $R_{s2} : [sex].[female], [unemployed].[no] \to [income].[high]\}$. R is classified as redundant according to condition 2 in definition 4. R is a single consequent rule so no separation is needed. By the application of Transitivity between $[income].[high] \to [loan].[yes]$ and $[sex].[female], [unemployed].[no] \to [loan].[yes]$ both in S_c the rule $R_1 : [sex].[female], [unemployed].[no] \to [loan].[yes]$ is obtained $R = R_1$ so R is a redundant rule.

Condition $X^{'} \subseteq X \wedge Y^{'} \cap X \neq \{\emptyset\}$ represents the case when any item in the antecedent of a rule is a redundant one. Let $R_1 : X \to Y$ and $R_2 : X^{'} \to Y^{'}$ be rules. Suppose $Y' \cap X = X_1$. Then by augmentation of X_1 in R_2 we have $R_3 : X^{'}X_1 \to X_1Y^{'}$ and by transitivity between R_3 and R_1 $R_4 : X \to Y$. Therefore Armstrong's axioms fulfill the condition. For example, with $R : \{[income].[high], [loan].[yes]\} \to \{[unemployed].[no]\}$ and $S_c = \{R_{s1} : [income].[high] \to [loan].[yes]$, $R_{s2} : [sex].[female], [unemployed].[no] \to [income].[high]\}$ R is classified as redundant by condition 3 in definition 4. Applying Reflexivity of $[income].[high]$ in $[income].[high] \to [loan].[yes]$ rule $R_1 : [income].[high] \to [income].[high], [loan].[yes]$ is obtained by Transitivity between R_1 and R we have $R_2 : [income].[high] \to [income].[high]$ R_2 is simpler than R with the same information so R is a redundant rule. However, by Augmentation of $[loan].[yes]$ in R_2 we have $R_3 : [income].[high], [loan].[yes] \to [unemployed].[no]$ $R = R_3$.

Condition $X^{'} \subseteq Y \wedge Y^{'} \cap Y \neq \{\emptyset\}$ represents the case when any item in the consequent of R is redundant with respect to other item in consequent. This condition is fulfilled if a single attribute in Y is redundant. Let $R_1 : X \to Y$ and $R_2 : X^{'} \to Y^{'}$ be rules. Suppose $Y \cap Y^{'} = Y_1$. Then by the reflexivity axiom on R_2 consequent $R_3 : Y^{'} \to Y_1$ by transitivity between R_2 and R_3 we have $R_4 : X^{'} \to Y_1$. By statement condition we have $X \subseteq Y$ so by transitivity between R_1 and R_4 we have $R_5 : X \to Y_1$. Therefore Armstrong's axioms fulfill the condition. For example, $R : \{[balance].[high], [unemployed].[no]\} \to \{[income].[high], [loan].[yes]\}$ and $S_c = \{R_{s1} : [income].[high] \to [loan].[yes]$, $R_{s2}[sex].[female], [unemployed].[no] \to [income].[high]\}$. R is redundant according to condition 4 in definition 4. Applying Reflexivity, Augmentation and Transitivity we obtain $R_1 : [balance].[high], [unemployed].[no] \to [income].[high]$ and $R_2 : [balance].[high], [unemployed].[no] \to [loan].[yes]$ now by Transitivity between R_1 and $[income].[high] \to [loan].[yes] \in S_c$ we have $R_3 : [balance].[high], [unemployed].[no] \to [loan].[yes]$. $R_2 = R_3$ so R is a redundant rule.

We do not use Armstrong's Axioms as an inference mechanism so, we do not worry if it is not able to ensure the support and confidence threshold in the inferred rules.

3.2 Algorithm to eliminate prior knowledge redundancy in association rules

In this section we present an algorithm to determine if a rule contains redundant items, see Fig. 1. The closure algorithm presented in [35] is used to compute X^{+}.

```
Require: Set of previous knowledge rules $S_c$
    A rule $R_i$ in form $X \rightarrow Y$
Ensure: Boolean value to indicate if the rule is redundant
$i = 0$
$n = |Y|$
while $i < n$ do
    if $Y[i] \in X^{+}_{S_c \cup X \rightarrow (Y - \{Y[i]\})}$ then
        return true
    end if
    $i = i + 1$
end while
$i = 0$
$n = |X|$
while $i < n$ do
    if $X[i] \in (X - X[i])^{+}_{S_c \cup (X - X[i]) \rightarrow Y}$ then
        return true
    end if
    $i = i + 1$
end while
return false
```

Algorithm 1: Prior Knowledge Redundancy detection

To determine the redundancy of a rule $X \rightarrow Y$ we have to prove if any item A in the rule's antecedent is redundant or if an item W in the consequent is redundant. The item A is redundant if the consequent can be derived from the prior knowledge without A. The first part of algorithm 1 performs this task for all items $A \in X$ by calculating the closure of the new antecedent $X - \{A\}$ over the previous knowledge rules joined to the studied rule focus, and comparing results with the closure of the same antecedent over the set of previous rules joined to a new rule, where the item A is not a part of the antecedent. If both results are equal, then the item A is redundant and the entire rule is also redundant. To test if item W is redundant we have to apply a similar procedure, the second part of algorithm 1 performs this task.

Example 2. ***Prior Knowledge Redundancy detection:*** *We use the following Prior Knowledge* $S_c = \{R_{s1} : [income].[high] \rightarrow [loan].[yes]$, $R_{s2} : [sex].[female], [unemployed].[no] \rightarrow [income].[high]\}$ *and the rules* $R_1 : \{[balance].[high], [unemployed].[no]\} \rightarrow \{[income].[high], [loan].[yes]\}$ *and* $R_2 : \{[income].[high], [loan].[yes]\} \rightarrow \{[unemployed].[no]\}$ *to show the performance of algorithm 1. For* R_1 *we have:*

The first step is to compute $F = S_c \cup R_i$ *for* R_1 $F = \{R_{f1} : [income].[high] \rightarrow [loan].[yes]$, $R_{f2} : [sex].[female], [unemployed].[no] \rightarrow [income].[high]$, $R_{f3} : [balance].[high], [unemployed].[no] \rightarrow [income].[high], [loan].[yes]\}$.

Second, checks the redundancy in the antecedent, computing closure of $[balance].[high]$ *over* F. *This is* $[balance].[high]^{+}_{F} = [balance].[high]$ *and comparing with closure of* $[balance].[high]$ *over* G *where* $G = ((F - \{R_1\}) \cup ([balance].[high]) \rightarrow [income].[high], [loan].[yes])$, $[balance].[high]^{+}_{G} = [balance].[high], [income].[high], [loan].[yes]$. *They are different so* $[unemployed].[no]$ *is not redundant. The item* $[balance].[high]$ *is also non-redundant.*

And last, checks the redundancy in the consequent.
$F^{'} = \{(F - R_1 \cup ([balance].[high], [unemployed].[no] \rightarrow [income].[high])\}$
$[balance].[high], [unemployed].[no]^{+}_{F} = [balance].[high], [unemployed].[no], [income].[high], [loan].[yes]$,
$[balance].[high], [unemployed].[no]^{+}_{F'} = [balance].[high], [unemployed].[no], [income].[high], [loan].[yes]$. *They are the same so the item* $[loan].[yes]$ *and the rule* R_1 *are redundant.*

For R_2 *we have:*

- $F^{'} = (F - R_1) \cup [income].[high] \rightarrow [unemployed].[no]$.
  $F = \{(F - R_1) : [income].[high] \rightarrow [loan].[yes]$, $R_{f2}[sex].[female], [unemployed].[no] \rightarrow [income].[high]$, $R_{f3}[income].[high], [loan].[yes] \rightarrow [unemployed].[no]\}$.
- $[income].[high]^{+}_{F} = [income].[high], [loan].[yes], [unemployed].[no]$, $[income].[high]^{+}_{F'} = [income].[high], [loan].[yes], [unemployed].[no]$. *They are the same so the rule is redundant.*

3.2.1 Correctness

We first prove that closure algorithm [35] can be used to detect redundancy according to definition 4. Closure algorithm applies Armstrong's axioms to find all items implied by a given itemset.

Theorem 1. *Let* $\mathcal{S}_c$ *be a set of prior known rules and* $R : X \rightarrow Y$ *an association rule. If there is a rule* $R^{'} : X^{'} \rightarrow Y^{'} \in \mathcal{S}_c$ *and* $X^{'} \subseteq X \wedge Y^{'} \cap Y \neq \{\emptyset\}$ *then* $Y^{'} \cap Y \in X^{+}_{\mathcal{S}_c}$

Proof. Assume $X^{'} \subseteq X \wedge Y^{'} \cap Y \neq \{\emptyset\}$. Then $X^{'} \in X^{+}_{\mathcal{S}_c}$ by assumption $X^{'} \subseteq X$ and reflexivity axiom. So

$Y^{'} \in X^{+}_{\mathcal{S}_c}$ by transitivity between $X \to X^{'}$ and $X^{'} \to Y^{'}$. Therefore $Y^{'} \cap Y \in X^{+}_{\mathcal{S}_c}$ by definition of set intersection. □

Theorem 2. *Let $\mathcal{S}_c$ be a set of prior known rules and $R : X \to Y$ one association rule. If there is a rule $R^{'} : X^{'} \to Y^{'} \in \mathcal{S}_c$ and $X^{'} \subseteq X \wedge \exists R^{''} : X^{''} \to Y^{''} \in \mathcal{S}_c : X^{''} \subseteq Y^{'} \wedge Y \subseteq Y^{''}$ then $Y \in X^{+}_{\mathcal{S}_c}$.*

Proof. Assume $X^{'} \subseteq X \wedge \exists R^{''} : X^{''} \to Y^{''} \in \mathcal{S}_c : X^{''} \subseteq Y^{'} \wedge Y \subseteq Y^{''}$. Then $X^{'} \in X^{+}_{\mathcal{S}_c}$ by assumption $X^{'} \subseteq X$ and reflexivity axiom. $Y^{'} \in X^{+}_{\mathcal{S}_c}$ by transitivity between $X \to X^{'}$ and $X^{'} \to Y^{'}$. $X^{''} \in X^{+}_{\mathcal{S}_c}$ by assumption $X^{''} \subseteq Y^{'}$ and subset definition. So $Y^{''} \in X^{+}_{\mathcal{S}_c}$ by transitivity between $X \to X^{''}$ and $X^{''} \to Y^{''}$. Therefore $Y \in X^{+}_{\mathcal{S}_c}$ by assumption $Y \subseteq Y^{''}$ and subset definition. □

Theorem 3. *Let $\mathcal{S}_c$ be a set of prior known rules and $R : X \to Y$ one association rule. If there is a rule $R^{'} : X^{'} \to Y^{'} \in \mathcal{S}_c$ and $X^{'} \subseteq X \wedge Y^{'} \cap X \neq \{\emptyset\}$ then $Y^{'} \cap X \in (X - (Y^{'} \cap X))^{+}_{\mathcal{S}_c}$.*

Proof. Assume $X^{'} \subseteq X \wedge Y^{'} \cap X \neq \{\emptyset\}$. Then $X^{'} \in (X - (Y^{'} \cap X))^{+}_{\mathcal{S}_c}$ by assumption $X^{'} \subseteq X$ and reflexivity axiom. $Y^{'} \in (X - (Y^{'} \cap X))^{+}_{\mathcal{S}_c}$ by transitivity between $X \to X^{'}$ and $X^{'} \to Y^{'}$. Therefore $Y^{'} \cap X \in (X - (Y^{'} \cap X))^{+}_{\mathcal{S}_c}$ by definition of set intersection. □

Theorem 4. *Let $\mathcal{S}_c$ be a set of prior known rules and $R : X \to Y$ one association rule. If there is a rule $R^{'} : X^{'} \to Y^{'} \in \mathcal{S}_c$ and $X^{'} \subseteq Y \wedge Y^{'} \cap Y \neq \{\emptyset\}$ then $Y^{'} \cap Y \in X^{+}_{\mathcal{S}_c \cup X \to (Y - (Y^{'} \cap Y))}$.*

Proof. Assume $X^{'} \subseteq Y \wedge Y^{'} \cap Y \neq \{\emptyset\}$. Then $X^{'} \in X^{+}_{\mathcal{S}_c \cup X \to (Y - (Y^{'} \cap Y))}$ by assumption $X^{'} \subseteq Y$ and association rule property $X \cap Y = \emptyset$. $Y^{'} \in X^{+}_{\mathcal{S}_c \cup X \to (Y - (Y^{'} \cap Y))}$ by transitivity between $X \to X^{'}$ and $X^{'} \to Y^{'}$. Therefore $Y^{'} \cap Y \in X^{+}_{\mathcal{S}_c \cup X \to (Y - (Y^{'} \cap Y))}$ by definition of set intersection. □

Theorem 5. *If $(\exists A_i \in X \wedge A_i \in (X - A_1)^{+}_{\mathcal{S}_c \cup (X - A_i) \to Y}) \vee (\exists W_i \in Y \wedge W_i \in X^{+}_{\mathcal{S}_c \cup X \to Y - W_i})$ then rule $X \to Y$ has prior knowledge redundancy over $\mathcal{S}_c$.*

Proof. Direct from theorem 1, theorem 2, theorem 3 and theorem 4. □

Hoare triple was introduced by C. A. R. Hoare [38] as $\{P\}C\{Q\}$, for specifying what a program does. In such a Hoare triple:

- C is a program.
- P and Q are assertions, conditions on the program variables used in C. They will be written using standard mathematical notation together with logical operators. We can use functions and predicates to express high-level properties based on a domain theory [39] covering specifics of the application area.

We say $\{P\}C\{Q\}$ is true, if whenever C is executed in a state satisfying $\{P\}$ and if the execution of C finishes, then the state in which C execution finishes satisfies Q. If there is a loop in C, loop invariants must be used to prove correctness. If loop invariants are proved to be true after each loop iteration then the postcondiction must be proven true.

In algorithm 1 lines one through eight and lines nine through sixteen perform basically the same operation, one over the rule antecedent and the other over the rule consequent. So we analize them only one time. Line four checks if $Y[i]$ is subset of the closure. So closure algorithm must be computed, this algorithm has been proved as correct[35]. The search of $Y[i]$ within closure can be done by a well known linear search algorithm, we assume it is correct.

Preconditions:

- $\mathcal{S}_c$ is a set of previous knowledge rules.
- $X \to Y$ is an association rule with $X = X_1, .., X_n$ and $Y = Y_1, .., Y_m$

Postcondition: If $(\exists A_i \in X \wedge A_i \in (X - A_1)^{+}_{\mathcal{S}_c \cup (X - A_i) \to Y}) \vee (\exists W_i \in Y \wedge W_i \in X^{+}_{\mathcal{S}_c \cup X \to Y - W_i})$ the return value is $true$.

Loop invariants: If the loop is executed j or more times, then after j executions

- $i = j$
- $0 \leq i \leq n$
- $Y[h] \notin X^{+}_{\mathcal{S}_c \cup X \to (Y - \{Y[i]\})}$ for $0 \leq h < i$

Proving the loop invariant: (by induction on j) **Base Case**: $j = 0$

- before first execution of loop $i = 0$
- loop invariant holds, $i = 0 \Rightarrow (0 \leq h < 0)$. No such h value.

Inductive hypothesis: assume that, if the loop iterates j times then the loop invariant holds $i_{old} = j$. Proving that if the loop iterates $j + 1$ times, then the loop invariant holds for $i_{new} = j + 1$. If true for iteration $i_{old} = j$ then $Y[h] \notin X^{+}_{\mathcal{S}_c \cup X \to (Y - \{Y[i]\})}$ for $0 \leq h < i_{old}$.

- if loop iterates then $Y[i_{old}] \notin X^{+}_{\mathcal{S}_c \cup X \to (Y - \{Y[i_{old}]\})}$ and $i_{new} = i_{old} + 1$.
- thus $Y[h] \notin X^{+}_{\mathcal{S}_c \cup X \to (Y - Y[h])}$ for $0 \leq h < i_{new}$.
- because loop iterated for $i_{old} = j$ we have $i_{old} < n$ and $i_{new} \leq n$

Thus, the loop invariant holds for $j+1$.

When the loop test fails, the loop invariant holds and either $i \geq n$ or $Y[i] \in X^{+}_{S_c \cup X \to (Y - Y[i])}$

- **Case 1** ($j \geq n$): loop invariant implies that $Y[h] \notin X^{+}_{S_c \cup X \to (Y - Y[h])}$ for $0 \leq h < n$, so no element in cosequent is a redundant one.

- **Case 2** ($j < n$): loop invariant implies that $Y[i] \in X^{+}_{S_c \cup X \to (Y - Y[i])}$ and $true$ is returned

Conclusions: Poscondition is satisfied in either case, so the algorithm is correct.

3.2.2 Complexity analysis

Time complexity of an algorithm is a function $T(n)$ limiting the maximum number of steps in the algorithm for an input size n. $T(n)$ depends on what is counted as one computation step, the random access machine (RAM) model is the most extended one. RAM is a model for a simple digital computer with random access memory. For the sake of simplicity $T(n)$ is approximated by a simplest function, it is written $T(n) = O(f(n))$ if there are constants $c \geq 0$ and $n_1 \geq 0$ such that: $T(n) \leq cf(n)$ for all $n \geq n_1$.

For algorithm in Fig 1 we considered a as the number of different attribute symbols in S_c and p the number of previous knowledge rules presented in S_c. The complexity order to compute the closure is $O(n)$ see [35]. The execution time of the first **while** loop (the consequent of the rule) takes $a * p$ since the number of rules in F is p, and we compute the closure with a cost of $O(p)$. The execution time of the second **while** loop (the antecedent of the rule) takes the same value of $a * p$ because it performs the same operation and in the same way the complexity of the steps is $O(ap)$. To compute the complexity of the entire algorithm, the complexity of the first and second **while** loops must be added so it is $O(ap) + O(ap) = 2O(ap)$ but the constant 2 can be ignored and the final value for complexity of the algorithm is $O(ap)$.

Association rules extraction algorithms have much higher complexity [36] than the reduction approach presented here. This difference led us to propose a reduction mechanism in which rule extraction algorithm is executed once and then, in the post processing stage, the reduction algorithm is fired to prune the redundant rules, rather than applying prior knowledge as restriction within the extraction algorithms, which would force to execute it for each different user and even for each change on a user's prior knowledge. The computational cost for the constraint approach is very high. However, our approach, in post processing stage, allows us to run a simpler routine when the user changes or the user prior knowledge is updated. The temporal cost of this approach did not exceed 5 seconds in any of the applied tests.

4 Experimental results

4.1 Methodology

In order to verify the effectiveness of our approach we performed experiments with four datasets. The first one with data about USA census[2], the second one with data about stock market investments [1], the third one with data about hypothetical samples of mushroom[2] and the last one with data about breast cancer[2]. Prior knowledge consists of 6 rules for each dataset. We use Pruning Ratio metric $PR = (PrunedRules/TotalRules) \times 100$ to evaluate our results.

Table 2 shows the result of the experiments. Each row corresponds to an experiment following the next steps:

1. Find the complete set of rules using as support threshold the value in column 2 and confidence threshold the value in column 3. The number of rules is showed in column 4.

2. Apply the steps presented in algorithm 1. The number of pruned rules are presented in column 5 of Table 2.

3. After applying the algorithm to the dataset, the final number of rules is presented in column 6 of Table 2 while column 7 contains the pruning ratio. The execution time is presented in column 8.

4.2 Results and discussion

Pruning Ratio changes according to support in Census and Stocks datasets, first increasing while the support increases, but when the support is greater than 0.07 for the Census dataset and greater than 0.5 for Stocks dataset, the Pruning Ratio decreases while the support increases. The behavior in Mushroom dataset is the opposite, the Pruning Ratio decreases while support increases until the support reaches the 0.5 value then the Pruning Ratio increases while the support value increases.

This behavior shows a relation between support and previous knowledge patterns. If the support is increased, then a number of rules do not meet the support threshold and they are discarded. Hence the discarded rules have no major impact on the rules derived from previous knowledge, Pruning Ratio will be increased, but as the support increases it starts to reduce the rules derived from previous knowledge, so the Pruning Ratio will be decreased.

In Fig 1, Fig 2 and Fig 3 the mean value of Pruning Ratio is shown for several support values in Census, Stocks and Mush datasets respectively using combination of all six rules in S_c.

Dataset	Support	Confidence	Rules	Pruned Rules	Final Rules	Pruning Ratio	Time
Census	0.01	0.4	3408	942	2466	27	0.589
Census	0.03	0.4	835	242	593	28	0.079
Census	0.05	0.4	458	158	300	32	0.043
Census	0.07	0.4	229	79	150	34	0.021
Census	0.09	0.4	163	51	112	31	0.015
Census	0.11	0.4	114	23	91	20	0.010
Stocks	0.2	0.4	11010	5592	5418	50	2.170
Stocks	0.3	0.4	3314	2225	1089	67	0.536
Stocks	0.4	0.4	1230	904	326	73	0.116
Stocks	0.5	0.4	349	294	55	84	0.039
Stocks	0.6	0.4	212	64	148	30	0.020
Mushroom	0.3	0.5	78998	29154	49844	36	11.245
Mushroom	0.4	0.5	5767	1225	4542	21	0.852
Mushroom	0.5	0.5	1148	200	948	17	0.098
Mushroom	0.6	0.5	266	88	178	33	0.025
Mushroom	0.7	0.5	180	83	97	46	0.017
Breast	0.01	0.4	210500	98582	111918	47	27.732
Breast	0.1	0.4	28808	13695	15113	47	4.190
Breast	0.2	0.4	6092	2982	3110	49	0.859
Breast	0.3	0.4	5284	2398	2886	45	0.798
Breast	0.4	0.4	1246	449	797	36	0.118

Table 2: Experiment's result

4.3 Traditional vs. knowledge based reduction

The approach developed in this paper differs from those published until now. Previous woks are concerned with the structural relationship between association rules and mechanisms to reduce redundancy using inference rules and maximal itemsets. We use the user experience to prune rules that do not bring new knowledge to the user, simplifying decision making. Both approaches are not comparable in essence, but we carried out experiments to compare KBR's pruning ratio with previous works.

Fig 4 shows the pruning ratio of some relevant works in redundancy reduction, over a Mushroom dataset with a support value of 0.3. We used Mushroom dataset because we can access to author experiments and it is sufficient to test our case. The values for pruning ratio are taken from the author's papers: MinMax, Reliable, GB, CHARM, CRS and MetaRules.[40]

Reliable has the best Pruning Ratio, see Fig 4, so we compare it with our approach at different support values, see Table 3.

Reliable Pruning Ratio is the best of KBR_{6rules}, KBR_{9rules} and $KBR_{12rules}$. Nevertheless, $KBR_{15rules}$ reaches better Pruning Ratio than Reliable for all supports except 0.4, see Fig. 6. A previous knowledge of 15 rules is equivalent to 0.018% of the whole rule set, for a support value of 0.3, and 7.9% for a support value of 0.7.

With very few rules in KBR is possible to exceed the Pruning Ratio of previous works. Of course there is a narrow relationship between the Pruning Ratio and the repercussion of the previous knowledge rules over the whole set of rules. The Pruning Ratio of knowledge rules increases in the same way that they are able to describe the domain under study. The better KBR results are, the better the user will know the domain under study. Our approach has the possibility to determine when a model can not be improved like in the case of $KBR_{15rules}$ for a support value of 0.7 where the Pruning Ratio is 100%.

4.4 Knowledge vs knowledge based reduction

In section 2 we surveyed some works that used knowledge to reduce the number of association rules presented to the final user. The main goal of those papers is to obtain a set of association rules that satisfies some constraint provided by users, using different forms of knowledge representation. They are able to reduce the association rules set cardinality but generate a lot of rules that represent the same knowledge. Strictly speaking we can not compare our proposal with those ones because of the difference between goals, but we want to test the association rules model cardinality reduction capability of our approach with template, the best known form of knowledge approach.

We compare the pruning ratio of our approach with the template implementation proposed in [41] that up-perform the implementation proposed in [16] across five dataset from [2].

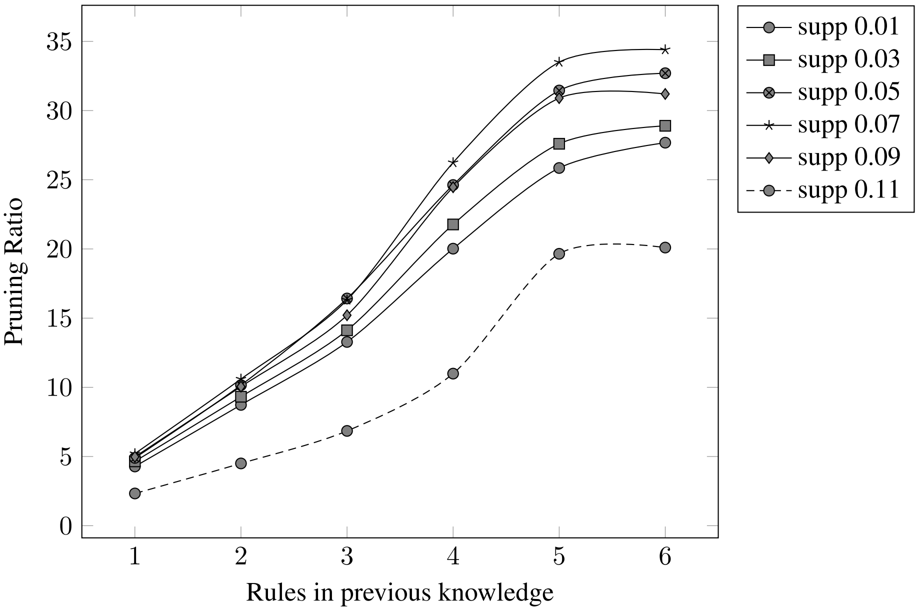

Figure 1: Rules pruned in census dataset

Support	Reliable	KBR_{6rules}	KBR_{9rules}	$KBR_{12rules}$	$KBR_{15rules}$
0.3	95	36	76	80	96
0.4	90	21	37	47	84
0.5	89	17	30	44	93
0.6	74	33	40	62	97
0.7	78	46	46	75	100
Average	85	32,5	45,8	61,5	94

Table 3: Pruning Ratio

- Mushroom data (mush)
- Johns Hopkins University Ionosphere data (ion)
- Statlog Project Heart Disease data (hea)
- Thyroid Disease data (thy)
- Attitudes Toward Workplace Smoking Restrictions data (smo)

The continuous attributes in the data sets used were discretized using a 4-bin equal-frequency discretization. Support and Confidence were set to the same values used in [16]. In table 4 we present the result of our pruning approach (KBR) and compare it with the previous work (MetaRules) [41].

Each row in table 4 represents an experiment where column Dataset contains the dataset id, column TotalRules shows the total number of rules produced by extraction algorithms, MetaRules presents the remaining rules after the application of the aplgorithm proposed in [41] while column KBR contains the average of remaining rules of ten runs of knowledge based redundancy elimination algorithm using a random knowledge of ten rules for each execution. The remaining rules in our approach are lower than the number of rules in metarules approach for all datasets.

Dataset	TotalRules	MetaRules	KBR
mush	1374	138	120.2
ion	1215	452	402.6
hea	371	246	176.7
thy	1442	502	431.6
smo	797	300	283.3

Table 4: Remaining rules

5 Conclusion

The fundamental idea in this work is linked to the main definition of data mining: analysis of large amount of data to extract interesting patterns, previously unknown and the consideration that an association rule that correspond to prior knowledge is a redundant one[37]. Our approach prunes those rules, presenting a simpler model to the final user.

The main contribution in this work is the definition of redundancy of association rules with respect to prior knowledge, and the definition of a mechanism to eliminate this kind of redundancy from the final model of association

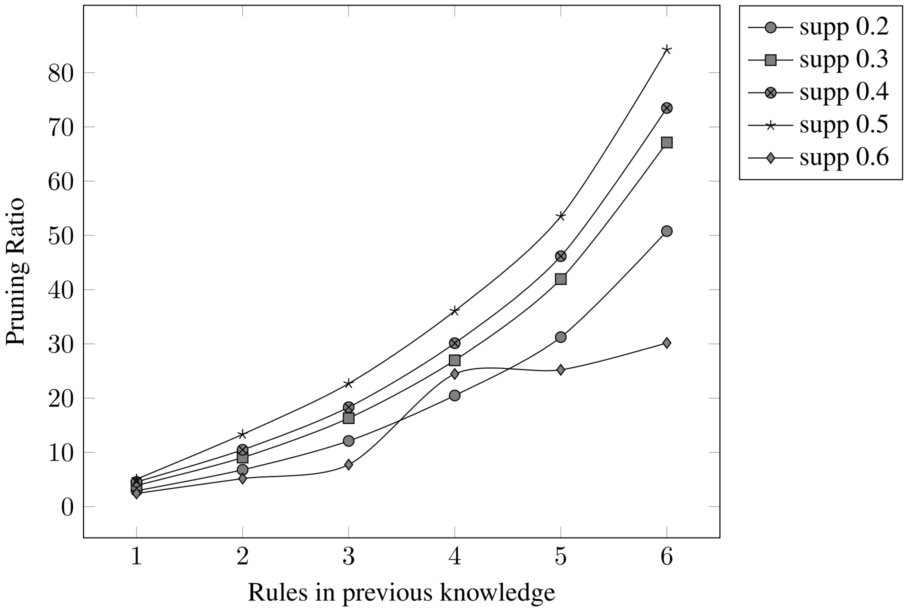

Figure 2: Rules pruned in stocks dataset

rules presented to the end user. The redundancy elimination is performed in two procedures, the first one to detect and prune redundant element in rules antecedent and consequent, and the second one to detect if all information provided by a rule is redundant with respect to prior knowledge and then to prune it.

The results of this study confirm it is possible to use prior knowledge of experts to reduce the volume of association rules. Models of association rules with fewer rules can be interpreted more clearly by specialists so they can generate advantages in decision making process. The experimental results show that prior knowledge of less than 10% can reach a reduction ratio above 90%.

Acknowledgement

This research is partially funded by the research project TIC1582 by “Consejeria de Economia, Innovacion, Ciencia y Empleo from Junta de Andalucia” (Spain).

References

[1] J. Núñez, (2007), “Empleo de Fuzzy OLAP para Obtener Reglas que Caractericen Estrategias de Inversión”.

[2] D. J. Newman, (2007), “UCI Repository of Machine Learning Databases”,University of California, School of Information and Computer Science, Irvine, CA.

[3] Sisodia, Dilip Singh and Singhal, Riya and Khandal, Vijay, (2018), “Comparative performance of interestingness measures to identify redundant and non-informative rules from web usage data”, International Journal of Technology. https://doi.org/10.14716/ijtech.v9i1.1510

[4] Ali Yousif Hasan, (2019), “Evaluation and Validation of the Interest of the Rules Association in Data-Mining”, International Journal of Computer Science and Mobile Computing, Vol.8 Issue.3, pp. 230-239.

[5] N. Bhargava, M. Shukla, (2016), “Survey of Interestingness Measures for Association Rules Mining: Data Mining, Data Science for Business Perspective”, International Journal of Computer Science and Information Technology (IJCSITS), Vol.6, No.2, Mar-April 2016, pp. 74-80.

[6] Sudarsanam, Nandan and Kumar, Nishanth and Sharma, Abhishek and Ravindran, Balaraman, (2019), “Rate of change analysis for interestingness measures”, Knowledge and Information Systems. https://doi.org/10.1007/s10115-019-01352-3

[7] J. Blanchard, F. Guillet, P. Kuntz, (2009), “Semantics-based classification of rule interestingness measures in Post-mining of association rules: techniques for effective knowledge extraction”, IGI Global, pp. 56-79. https://doi.org/10.4018/978-1-60566-404-0.ch004

[8] V. de Carvalho, V. Oliveira, R. de Padua, S. Oliveira, (2016), “Solving the Problem of Selecting Suitable Objective Measures by Clustering Association Rules Through the Measures Themselves”, SOFSEM 2016: Theory and Practice of Computer Science. Springer Berlin Heidelberg. pp. 505-517. https://doi.org/10.1007/978-3-662-49192-8_41

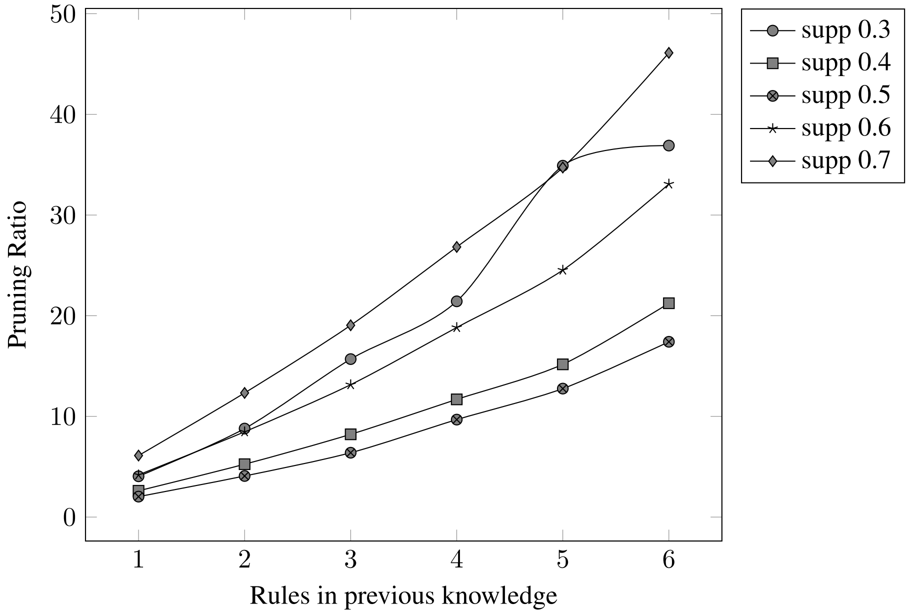

Figure 3: Rules pruned in Mushroom dataset

[9] V. Oliveira, D. Duarte, M. Violante, W. dos Santos, R. de Padua, S. Oliveira, (2017), “Ranking Association Rules by Clustering Through Interestingness”, in Mexican International Conference on Artificial Intelligence, pp 336-351. Annals of Data Science 1.1 (2014): pp. 25-39.

[10] D. R. Carvalho, A. A. Freitas, N. Ebecken, (2005), “Evaluating the correlation between objective rule interestingness measures and real human interest”, Knowledge Discovery in Databases: PKDD 2005, Springer, pp. 453-461. https://doi.org/10.1007/11564126_45

[11] A. Silberschatz, A. Tuzhilin, (1996), “What makes patterns interesting in knowledge discovery systems”, IEEE Trans. Knowledge Data Eng, vol. 8, no. 6, pp. 970-974. https://doi.org/10.1109/69.553165

[12] R. Batra, M. A. Rehman, (2019), “Actionable Knowledge Dsicovery for Increasing Enterprise Profit, Using Domain Driven Data Mining.”, IEEE Acces vol.7, pp. 182924-182936. https://doi.org/10.1109/access.2019.2959841

[13] R. Sehti, B. Shekar, (2019), “Subjective interestingness in Association Rule Mining: A Theoretical Analysis”, Digital Business, Springer Charm, pp. 375-389. https://doi.org/10.1007/978-3-319-93940-7_15

[14] L. Greeshma, G. Pradeepini, (2016), “Unique Constraint Frequent Item Set Mining”, Advanced Computing (IACC), 2016 IEEE 6th International Conference on pp. 68-72. IEEE. https://doi.org/10.1109/iacc.2016.23

[15] A. Kaur, V. Aggarwal, S. K. Shankar, (2016), “An efficient algorithm for generating association rules by using constrained itemsets mining”, Recent Trends in Electronics, Information Communication Technology (RTEICT), IEEE International Conference on (pp. 99-102). IEEE. 2016. https://doi.org/10.1109/rteict.2016.7807791

[16] Berrado, G. C. Runger, (2007), “Using metarules to organize and group discovered association rules”, Data Mining and Knowledge Discovery, vol. 14, no. 3, pp. 409-431. https://doi.org/10.1007/s10618-006-0062-6

[17] W. Liu, W. Hsu, S. Chen, (1997), “Using General Impressions to Analyze Discovered Classification Rules”, KDD, pp. 31-36

[18] W. Liu, W. Hsu, K. Wang, S. Chen, (1999), “Visually aided exploration of interesting association rules”, Methodologies for Knowledge Discovery and Data Mining, Springer, pp. 380-389. https://doi.org/10.1007/3-540-48912-6_52

[19] B. Liu, W. Hsu, S. Chen, Y. Ma, (2000), “Analyzing the subjective interestingness of association rules”, Intell. Syst. Their Appl. IEEE, vol. 15, no. 5, pp. 47-55. https://doi.org/10.1109/5254.889106

[20] B. Padmanabhan, A. Tuzhilin, (2000), “Small is beautiful: discovering the minimal set of unexpected patterns”, Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and

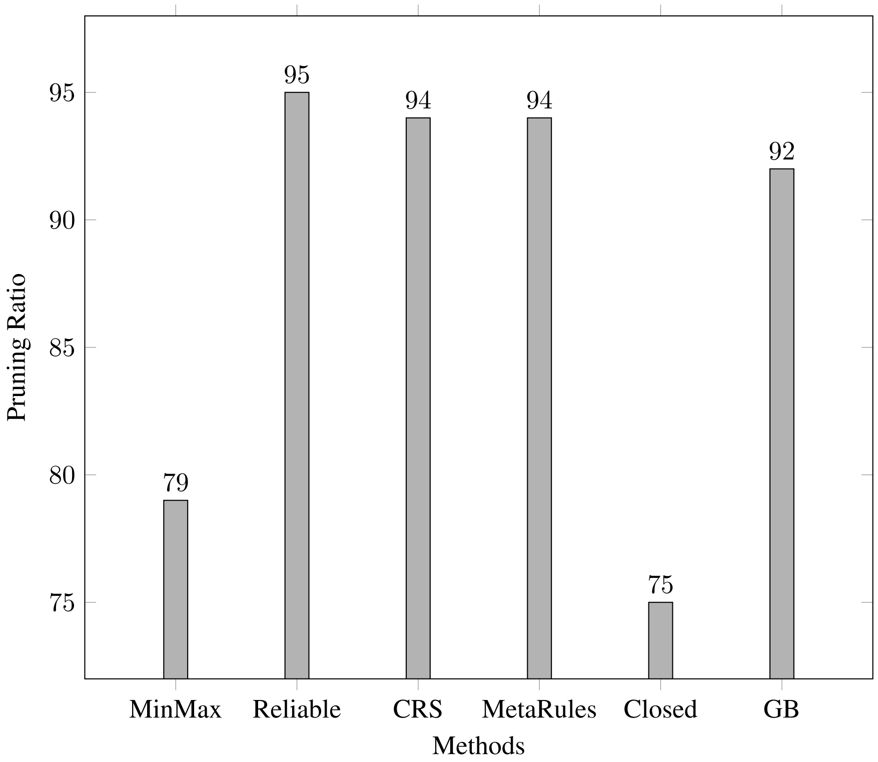

Figure 4: Pruning Ratio for different approaches

data mining, pp. 54-63. https://doi.org/10.1145/347090.347103

[21] K. Wang, Y. Jiang, L. V. Lakshmanan, (2003), "Mining unexpected rules by pushing user dynamics", Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 246-255. https://doi.org/10.1145/956750.956780

[22] Y. Bastide, N. Pasquier, R. Taouil, G. Stumme, and L. Lakhal, (2000), "Mining minimal nonredundant association rules using frequent closed itemsets", Proc. International Conference on Computational Logic (CL 2000), pp. 972-986. https://doi.org/10.1007/3-540-44957-4_65

[23] M. Quadrana, A. Bifet, R. Gavalda, (2015), "An efficient closed frequent itemset miner for the MOA stream mining system", AI Communications 28.1: pp. 143-158. https://doi.org/10.3233/aic-140615

[24] L. Greeshma, G. Pradeepini, (2016), "Mining Maximal Efficient Closed Itemsets Without Any Redundancy", Information Systems Design and Intelligent Applications. Springer India. pp. 339-347. https://doi.org/10.1007/978-81-322-2755-7_36

[25] G. Gasmi, S. B. Yahia, E. M. Nguifo, Y. Slimani, (2005), "A New Informative Generic Base of Association Rules", Advances in Knowledge Discovery and Data Mining, pp. 81-90, Springer Berlin Heidelberg. https://doi.org/10.1007/11430919_11

[26] C. L. Cherif, W. Bellegua, S. Ben Yahia, G. Guesmi, (2005), "VIE-MGB: A Visual Interactive Exploration of Minimal Generic Basis of Association Rules", Proc. International Conferences on Concept Lattices and Applications (CLA 2005), pp.179-196.

[27] P. Fournier-Viger, Wu C.-W., V. S. Tseng, (2014), "Novel Concise Representations of High Utility Itemsets using Generator Patterns", Proc. 10th International Conference on Advanced Data Mining and Applications, Springer LNAI. https://doi.org/10.1007/978-3-319-14717-8_3

[28] Y. Xu, Y. Li, G. Shaw, (2011), "Reliable representations for association rules", Data and Knowledge Engineering, vol. 70, no. 6, pp. 555-575. https://doi.org/10.1016/j.datak.2011.02.003

[29] Phan-Luong, (2001), "The representative basis for association rules", Proc. IEEE. International Conference on Data Mining (ICDM 2001), pp. 639-640. https://doi.org/10.1109/icdm.2001.989588

[30] B. Baesens, S. Viaene, and J. Vanthienen, (2000), "Post-processing of association rules", DTEW Res. Rep. 0020, pp. 118.

[31] J. Hipp, U. Gntzer, (2002), "Is pushing constraints deeply into the mining algorithms really

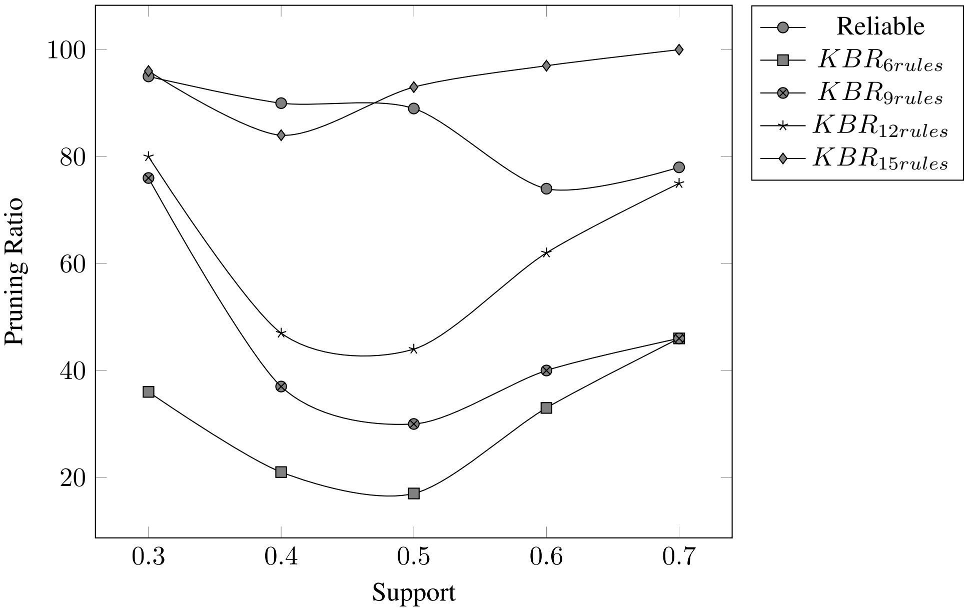

Figure 5: Reliable vs KBR pruning ratio

what we want?: an alternative approach for association rule mining", ACM SIGKDD Explorations 4(1), pp.50-55. https://doi.org/10.1145/568574.568582

[32] R. J. Bayardo, (2005), "The Hows, Whys, and Whens of Constraints and Itemset and Rule Discovery", Constraint-Based Mining and Inductive Databases LNCS3848, Springer: pp.1-13. https://doi.org/10.1007/11615576_1

[33] W. Armstrong, (1974), "Dependency structures of database relationships", IFIP Congress, pp. 580-583.

[34] Tirnuc, Cristina and Balcázar, José L. and Gómez-Pérez, Domingo, (2020), "Closed-SetBased Discovery of Representative Association Rules", International Journal of Foundations of Computer Science, vol. 31, no.1, pp. 143-156. https://doi.org/10.1142/s0129054120400109

[35] D. Maier, (1983), "Theory of Relational Database".

[36] W. A. Kosters, W. Pijls, V. Popova, (2003), "Complexity analysis of depth first and fp-growth implementations of apriori", Machine Learning and Data Mining in Pattern Recognition, Springer, pp. 284-292. https://doi.org/10.1007/3-540-45065-3_25

[37] H. Toivonen, M. Klemettinen, P. Ronkainen, K. Htnen, H. Mannila, (1995), "Pruning and grouping discovered association rules", MLnet Wkshp. on Statistics, Machine Learning, and Discovery in Databases.

[38] C. A. R. Hoare, (1972), "An axiomatic basis for computer programming", Communications of the ACM, 12, pp. 334-341.

[39] C. A. Furia, B. Meyer, S. Velder, (2014), "Loop invariants: Analysis, classification, and examples", ACM Computing Surveys (CSUR), vol. 46, no 3, p. 34. https://doi.org/10.1145/2506375

[40] Y. Xu, Y. Li, G. Shaw, (2011), "Reliable representations for association rules". Data & Knowledge Engineering, 70(6), 555-575. Elsevier. https://doi.org/10.1016/j.datak.2011.02.003

[41] Djenouri, Y., Belhadi, A., Fournier-Viger, P., Lin, J. C. W. (2018). Discovering strong meta association rules using bees swarm optimization. In Pacific-Asia Conference on Knowledge Discovery and Data Mining (2018, June) (pp. 195-206). Springer, Cham. https://doi.org/10.1007/978-3-030-04503-6_21