Semi-Supervised Learning to Support the Exploration of Association Rules

Veronica Oliveira de Carvalho[1], Renan de Padua[2],
and Solange Oliveira Rezende[2]

[1] Instituto de Geociências e Ciências Exatas,
UNESP - Univ Estadual Paulista, Rio Claro, Brazil
veronica@rc.unesp.br

[2] Instituto de Ciências Matemáticas e de Computação,
USP - Universidade de São Paulo, São Carlos, Brazil
{padua,solange}@icmc.usp.br

Abstract. In the last years, many approaches for post-processing association rules have been proposed. The automatics are simple to use, but they don't consider users' subjectivity. Unlike, the approaches that consider subjectivity need an explicit description of the users' knowledge and/or interests, requiring a considerable time from the user. Looking at the problem from another perspective, post-processing can be seen as a classification task, in which the user labels some rules as interesting [I] or not interesting [NI], for example, in order to propagate these labels to the other unlabeled rules. This work presents a framework for post-processing association rules that uses semi-supervised learning in which: (a) the user is constantly directed to the [I] patterns of the domain, minimizing his exploration effort by reducing the exploration space, since his knowledge and/or interests are iteratively propagated; (b) the users' subjectivity is considered without using any formalism, making the task simpler.

Keywords: Association Rules, Post-processing, Semi-supervised Learning (SSL).

1 Introduction

Association is a widely used task in data mining that has been applied in many domains due to its simplicity and comprehensibility. Since the task discovers strong correlations that may exist among the data set items, the problem is the number of patterns that are obtained. Even a small data set can generate a sufficient number of rules that can overload the user in the post-processing phase. In fact, finding interesting patterns in this huge exploration space becomes a challenge. It is infeasible for the user to explicitly explore all the obtained patterns in order to identify whether they are relevant or not.

Many post-processing approaches have been proposed to overcome the exposed problem (Section 2). The aim is to provide tools that allow users to find

L. Bellatreche and M.K. Mohania (Eds.): DaWaK 2014, LNCS 8646, pp. 452–464, 2014.

the interesting patterns of the domain so that their effort is minimized – the idea is that users don't need to explore all the rules in order to identify what is relevant or not. Post-processing approaches can be automatic or not, i.e., if it is necessary or not to provide information to achieve the desired answers. Although the automatics are simple to use, as objective evaluation measures, the users' subjectivity is not considered. Thereby, since the user is the person who will in fact validate the results, many approaches consider the user's domain knowledge and/or interests. In these cases, the user explicitly describes, through some formalism (ontologies, schemas, etc.), his knowledge and/or interests. However, providing such descriptions requires a considerable time from the user, which may lead to incomplete and/or incorrect specifications – sometimes known relations are forgotten and a previous knowledge which a user has another user may not have. Additionally, in most of the times, the user doesn't have an idea of what is probably interesting, nor from which relations to start the search, since the mining motivation is to support the user to find what he doesn't know.

Considering the post-processing phase from another perspective, the problem can be seen as a classification task, in which the user must label the rules as interesting [I] or not interesting [NI], for example (in fact, other classes may exist). As mentioned before, it is infeasible to explicitly explore all the obtained patterns in order to split the ones that are [I] from the patterns that are [NI]. However, if the user could label few rules and propagate these labels to the other unlabeled rules, the user would minimize his exploration effort. In this context, the semi-supervised learning (SSL) seems useful, since it is suitable when there are many unlabeled data and few labeled data. Besides, its use is also adequate when labeled data are expensive and/or scarce to obtain: it is expensive to discover the rules' labels, since the user must do the labeling process.

Based on the exposed, it would be relevant to develop a framework in which the user's knowledge and/or interests be implicitly obtained, through an iterative and interactive process, in such a way that this information be automatically propagated through the rule set, in order to direct the user to the [I] patterns of the domain. This work presents a framework for post-processing association rules that uses SSL to direct the users to the [I] patterns of the domain: starting from a subset of rules evaluated (labeled) by the user and suggested by the framework in order to implicitly capture the user's knowledge and/or interests, a SSL algorithm is applied to propagate the obtained labels to the rules which are not evaluated yet. Thereby, this paper contributes with current researches since: (a) the user is constantly directed to the potentially [I] patterns of the domain, which minimize his exploration effort through a reduction in the exploration space, once his knowledge and/or interests are iteratively propagated; (b) the user's subjectivity is considered, although his knowledge and/or interests be implicitly obtained, without using any formalism, making this specification task simpler. To the best of our knowledge, this is the first work that discusses a framework for post-processing association rules based on SSL.

This paper is organized as follows. Section 2 describes related researches as basic concepts. Section 3 presents the proposed framework. Section 4 describes some experiments that were carried out to analyze the framework. Section 5 discusses the results obtained in the experiments. Finally, conclusion is given in Section 6.

2 Background

In this section a brief introduction of the concepts related to the paper are presented, as well as the related works.

Association Rules Post-processing Approaches. The aim of the post-processing approaches is to provide tools that allow users to find the interesting patterns of the domain in order to minimize their effort. In the [*Filtering by Constraints*] approaches the user explores the rules through constraints imposed on the mined patterns (examples in [1,2]). To specify these constraints some formalism, as templates and/or schemas, are used. In the [*Evaluation Measures*] approaches the rules are evaluated according to their relevance (examples in [3]). These measures are usually classified as objective or subjective: the objectives depend on the pattern structure; the subjectives depend on the user's interests and/or needs. In the [*Summarization*] approaches the aim is to condense the discovered rules in general concepts to provide an overview of the extracted patterns (examples in [4,5]). The abstraction can be done, for example, through generalization processes through ontologies. In the [*Grouping*] approaches the aim is to provide groups of similar rules to organize the patterns (examples in [6,7]). Frequently, grouping is done through clustering algorithms. In the [*Pruning*] approaches the aim is to find what is interesting by removing what is not interesting (examples in [1,2,7]). Finally, in the [*Hybrid*] approaches two or more of the previously approaches are combined (examples in [6,1,2,7]). In this case, they are related with an interactive process, i.e., in which the user is needed – these interactions have been done through the codification of the user's knowledge. Thus, to the best of our knowledge, this is the first work that presents a framework for post-processing based on SSL. Besides, the framework contains some elements of the above approaches and, so, it can be categorized in some of them: "Filtering by Constraints", "Grouping", "Pruning" and "Hybrid".

Semi-supervised Learning [8]. SSL is between supervised and unsupervised learning and, so, it learns from both labeled and unlabeled data. SSL is useful when labeled data are scarce and/or expensive – it is difficult to obtain, in some tasks, a reasonable number of labeled data, since human annotators, special devices, etc. can be necessary. Thereby, the goal is to find a function f, both from labeled and unlabeled data, that will better map the domain in relation to a function f found only over a few number of labeled data. Distinct SSL methods exist, as self-training, co-training and graph-based models, in which each method considers a different assumption about the existing relation between the unlabeled data distribution and the target label. Thus, the performance of the SSL depends on the correctness of the assumptions made by these methods. However, selecting the best assumption for a given application is an open question in the field. One of the simplest SSL methods is self-training: in this case, the learning process uses its own predictions to teach itself. As advantages its simplicity and the fact of being a wrapper method can be cited. As a disadvantage the propagation of errors can be cited: an initial error through f can reinforce its own errors, leading to incorrect classifications. Finally, the self-training assumption considers that its own predictions, at least the more confident, tend to be correct.

3 Proposed Framework for Post-processing Association Rules

The proposed framework for post-processing association rules, seen in Figure 1, is presented in this section. Initially, the association rule set R, seen as the training set, contains all the non evaluated knowledge of the domain. In other words, the training set only contains unlabeled data, i.e., unlabeled rules. Therefore, in the beginning, a subset S of association rules (AR) is automatically selected to be classified, i.e., labeled, by the user (Step [A]). The user labels the rules in S, presented to him, based on some predefined classes (Step [B]). After this, the training set R contains both labeled and unlabeled rules. At this point, a SSL algorithm is applied in order to propagate the user's knowledge and/or interests (labels), implicitly obtained, for all the other rules (Step C). Finally, a stopping criterion is evaluated (Step [D]). If the stopping criterion is met, the rules, which are already classified, are shown to the user; otherwise (Step [E]), some rules are again specified as unlabeled, in order to re-start the process iteratively. This step allows some of the rules' labels to change during the process due to the user's knowledge and/or interests that are obtained through the iterations.

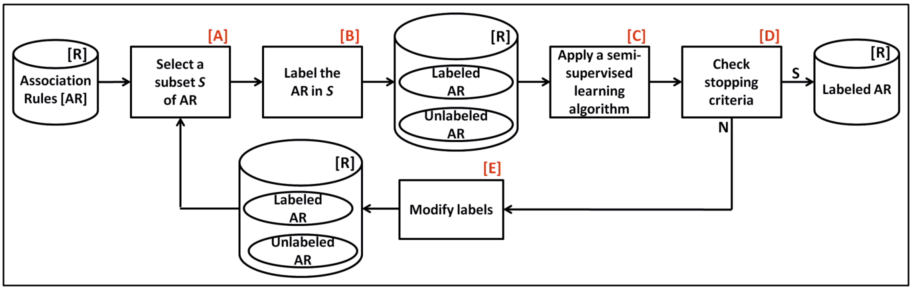

Fig. 1. The proposed framework

The proposed framework provides a post-processing strategy which has not been investigated yet in the literature (see Section 2) and which is flexible enough, since:

- many selection criteria, regarding Step [A], can be used. In this work, a combination of coverage, interest and rule size was used. A rule $r : a \Rightarrow c$ covers a rule $r' : b \Rightarrow d$ if $b \subseteq a$ and $d \subseteq c$; thus, the number of rules covered by r gives it coverage. The subset S is selected as described in Algorithm 1. Basically, n rules are chosen from R, in which $n/2$ are general and $n/2$ are specific. A rule is general if it has 2 or 3 items (rule size = 2 or 3) and specific if it has 4 or 5 items (rule size = 4 or 5) – we worked with rules composed of a maximum of 5 items (see Section 4). Since the rules in R form a lattice, the idea was to carry out a bidirectional search in this exploration space. First of all, the algorithm tries to pick up these n rules from the rules implicitly labeled as [I], also considering their coverage (the higher the coverage the better). Implicit means that the rule was automatically labeled by the SSL algorithm; explicit, on the other hand, means that the user saw the rule and labeled it[1]. If there are more than n implicit [I] rules, the non selected rules are set as unlabeled to ensure that the rules' labels can change through the iteration (details in Step [E]). If there aren't enough rules in this first subset (coverage+interest), the remaining rules to complete the amount of n are selected from the unlabeled ones, also considering their coverage. Specific details are presented in Algorithm 1 with comments. Other criteria will be explored in future works, since they correlate with the SSL methods in use – graphs based methods, for example, provide centrality measures (betweenness, closeness, etc.).

Algorithm 1. Step [A] procedure

```
Input: $R$: an association rule set, $n$: number of rules to be selected.
Output: $S$: subset of rules to be labeled by the user.
 1: $S_1$ := rules in $R$ implicitly labeled as [I] ordered by coverage (highest to lowest)
 2: $S_2$ := unlabeled rules ordered by coverage
 3: $nr := n/2$ {number of general and specific rules to be selected from $S_1$}
 4: $S_g$ := the first $nr$ general rules from $S_1$
 5: $S_e$ := the first $nr$ specific rules from $S_1$
 6: {if there aren't enough rules in $S_g$ and $S_e$, complete the sets with rules containing opposite sizes
    – ensures to select among the implicit [I] ones}
 7: if ($|S_g| < nr$) then $S_g := S_g \cup (nr - |S_g|)$ next specific rules from $S_1$ end if
 8: if ($|S_e| < nr$) then $S_e := S_e \cup (nr - |S_e|)$ next general rules from $S_1$ end if
 9: $S_1 := S_1 - (S_g \cup S_e)$
10: Set the remaining rules in $S_1$ as unlabeled and update $R$ {if more than $n$ rules appear in $S_1$, set
    the non selected ones as unlabeled – ensures changes in the rules' labels during the iteration}
11: {if there aren't enough rules in $S_g$ and $S_e$, complete the sets with the unlabeled ones containing
    the same sizes}
12: if ($|S_g| < nr$) then $S_g := S_g \cup (nr - |S_g|)$ next general rules from $S_2$ end if
13: if ($|S_e| < nr$) then $S_e := S_e \cup (nr - |S_e|)$ next specific rules from $S_2$ end if
14: {if there aren't enough rules in $S_g$ and $S_e$, complete the sets with the unlabeled ones containing
    opposite sizes}
15: if ($|S_g| < nr$) then $S_g := S_g \cup (nr - |S_g|)$ next specific rules from $S_2$ end if
16: if ($|S_e| < nr$) then $S_e := S_e \cup (nr - |S_e|)$ next general rules from $S_2$ end if
17: $S := S_g \cup S_e$
```

- many classes, regarding Step [B], can be considered. In this work, the classes [I] and [NI] were used as labels. However, the task of specifying the classes could be allowed to the user, providing more flexibility.
- hypothetically, any SSL method, regarding Step [C], can be used, provided that the methods assumptions be correct. In this work, the self-training method based on kNN was initially used, as seen in Algorithm 2 – as mentioned before,

[1] In the first execution of step [A] S_1 is empty, since there are no implicity labeled rule in R – all rules in R start as unlabeled – and, therefore, Algorithm 1 starts in line 12.

the training set is the association rule set R composed with labeled and unlabeled rules. However, other methods will be explored in future works.

- many stopping criteria, regarding Step [D], can also be considered. In this work, the process is executed until a subset G of gold rules be found (details in Section 4). As in the items above, other stopping criteria will be explored in future works.
- the criterion related to the change of the labels, regarding Step [E], ensures the iterativity of the process and the update of the labels. Once the stopping criterion has been checked (Step [D]), some of the rules' labels can be modified in the next round of Step [C] due to the user's knowledge and/or interests that will be obtained during the next iteration (Steps [A]+[B]). This means that some of the labels can change depending on the next user information – these new information can direct the user to another subset of the exploration space. Thereby, this step modifies the implicit rules labeled as [NI] to unlabeled (the implicit rules labeled as [I] are modified to unlabeled in Step [A] due to the same reasons (see Step [A])).

Algorithm 2. Self-training with kNN

```
Input: $R$: training set with labeled and unlabeled data (i.e., rules); $d$: distance function; $k$: number of neighbors.
Output: $R$: training set with labeled data.
$L$ := labeled data from $R$; $U$ := unlabeled data from $R$
repeat
    Select an unlabeled instance $u := \text{argmin}_{u \in U} \min_{l \in L} d(u, l)$
    Set $u$'s label with the same label of its nearest $l$ neighbor
    Remove $u$ from $U$; add $u$ with its label to $L$
until $U$ is empty
$R := L$
```

As seen, the proposed framework opens many researches possibilities, since it can be instantiated in many different manners. In this paper, the framework's ideas are explored through simple algorithms in order to demonstrate its feasibility (see Section 4). However, it can be noticed that: (a) the user is constantly directed to the potentially [I] patterns of the domain, which minimize his exploration effort through a reduction in the exploration space, since his knowledge and/or interests are iteratively propagated; (b) the user's subjectivity is considered, although his knowledge and/or interests be implicitly obtained, without using any formalism, making this specification task simpler. Besides, the framework can be seen as a hybrid approach since it contains characteristics of some other post-processing approaches described in Section 2. Finally, the framework also allows that some other post-processing approaches be used as an initial step in order to select and label some rules before starting Step [C] – in this case, the input is a rule set R containing both labeled and unlabeled rules (scenario obtained after applying Steps [A]+[B]).

4 Experiments

Some experiments were carried out in order to demonstrate the feasibility of the framework. Looking at Figure 1, it can be seen that, first of all, a rule set

R is needed. Thus, in order to evaluate the process, six data sets were used, which were divided in relational and transactional. In all of them the rules were extracted with an Apriori implementation[2] with a minimum of two items and a maximum of five items per rule.

The relational were Weather-Nominal (5;14), Contact-Lenses (5;24), Balloons (5;76) and Hayes-Roth (5;132). The numbers in parentheses indicate, respectively, number of attributes and number of instances. The first two are available in Weka[3]; the other two in the UCI Repository[4]. Before extracting the rules, the data sets were converted to a transactional format, where each transaction was composed by pairs of the form "attribute=value". Besides, in order to produce a suitable number of rules a minimum support (min-sup) of 0.0% and a minimum confidence (min-conf) of 0.0% were used in Weather-Nominal, Contact-Lenses and Balloons – in fact, all possible combinations were generated in each one. The values of min-sup=2.5% and min-conf=0.5% were used to Hayes-Roth set. 722 rules were obtained for Weather-Nominal, 890 for Contact-Lenses, 772 for Balloons and 889 for Hayes-Roth.

The transactional were Groceries (9835;169) and Sup (1716;1939). In this case, the numbers in parentheses indicate, respectively, number of transactions and number of distinct items. The first one is available in the R Project for Statistical Computing through the package "arules"[5]. The last one was donated by a supermarket located in São Carlos city, Brazil. With the Groceries data set 1092 rules were generated using a min-sup of 0.7% and a min-conf of 0.5% and with Sup 1149 rules considering a min-sup of 1.25% and a min-conf of 0.5%. The values, as in the cases above, were chosen experimentally.

Given a rule set R, Step [A] is executed. Thus, the number n of rules to be selected, in order to built S, was set to 4 (see Algorithm 1). In Step [B], the user must label the subset S of rules. As human evaluations in distinct data sets are difficult to obtain, a labeling process was simulated as presented in Algorithm 3. Consider that a set G of gold rules exists, i.e., a set containing the interesting rules of the domain that would be obtained by the user if he had evaluated all the rules in the set. For each rule to be labeled, it is computed its distance to each rule in G and the shortest distance is stored. After that, using this distance information, it is checked if the stored distance is $\leq$ a given threshold t, i.e., if the distance of the current rule in relation to a gold rule is small. If so, the rule is labeled as [I]; otherwise as [NI]. In the experiments, t was set to 0.3. The distance function used was the same as in the self-training method (see below). The set G was built as follows: (a) regarding the relational data sets, the C4.5[6] classifier was executed and the rules expressed through the decision trees were used to compose G – although the rules in G contain as consequent the classes of the

[2] Developed by Christian Borgelt: `http://www.borgelt.net/apriori.html`

[3] `http://www.cs.waikato.ac.nz/ml/weka/`

[4] `http://archive.ics.uci.edu/ml/`. In Ballons the four available sets were joined. In Hayes-Roth, the first attribute, regarding an ID number, was not considered.

[5] `http://cran.r-project.org/web/packages/arules/index.html`

[6] In fact, the J48 version available in Weka was executed with the default parameters.

sets, the rules in R contain any pair "attribute=value", since all possible relations were extracted; (b) regarding the transactional data sets, r rules were randomly selected to compose $G - r$ was set to a value representing less than 1% of the total of rules in R to maintain the same pattern as in the relational ones. The number of gold rules in G was: Weather-Nominal G=5 (0.69%), Contact-Lenses G=4 (0.45%), Balloons G=7 (0.91%), Hayes-Roth G=12 (1.35%)[7], Groceries G=7 (0.64%) and Sup G=9 (0.78%).

Algorithm 3. Algorithm used to simulate the labeling process.

```
Input: $S$: a subset $S$ of unlabeled rules;
    $G$: a set of gold rules; $t$: threshold; $d$:
    distance function.
Output: $S$: a subset $S$ of labeled rules.
for all $r \in S$ do
    $d(r) := \min_{g \in G} d(r, g)$
end for
for all $r \in S$ do
    if $d(r) \leq t$ then $r := [I]$ end if
    if $d(r) > t$ then $r := [NI]$ end if
end for
```

Table 1. Configurations used to apply the proposed framework

Data sets	Weather-Nominal; Contact-Lenses; Balloons; Hayes-Roth; Groceries; Sup
Step [A]	Algorithm 1; n=4
Step [B]	Algorithm 3; t=0.30; distance function: d_{jacc}
Step [C]	Algorithm 2; k=1; distance function: d_{jacc}
Step [D]	until set G is found

[1] Different values for n, t and k were tested, being the ones here presented the most suitable to the proposed framework.

In relation to kNN, used as the base to execute the self-training method (see Algorithm 2), k was set with 1. Regarding the distance function, an adaptation of Jaccard measure was used: $d_{jacc}(r_1, r_2) = 1 - \frac{\text{Jacc}+\text{Jacc}_A+\text{Jacc}_C}{3}$. The function considers the average similarity, using Jaccard, among all the items in the rules ($\text{Jacc} = \frac{|\{items\ in\ r_1\} \cap \{items\ in\ r_2\}|}{|\{items\ in\ r_1\} \cup \{items\ in\ r_2\}|}$), all the items appearing only in the antecedents of the rules ($\text{Jacc}_A = \frac{|\{items\ in\ r_1\ antecedent\} \cap \{items\ in\ r_2\ antecedent\}|}{|\{items\ in\ r_1\ antecedent\} \cup \{items\ in\ r_2\ antecedent\}|}$) and all the items appearing only in the consequents of the rules ($\text{Jacc}_C = \frac{|\{items\ in\ r_1\ consequent\} \cap \{items\ in\ r_2\ consequent\}|}{|\{items\ in\ r_1\ consequent\} \cup \{items\ in\ r_2\ consequent\}|}$). This strategy distinguishes the similarity between the rules concerning their items' position. Otherwise, only the itemset similarity is measured, not considering the rules' implication. Finally, in relation to the stopping criterion, the process was executed until all the set G of gold rules was found. Table 1 summarizes the configurations used in the experiments.

Lastly, for each data set, a comparison between the proposed framework and a traditional post-processing approach was done. In this case, the rules in R were ranked, considering the average rating obtained through 18 objective measures, as follows ([*Evaluation Measures*] approach (see Section 2)): (i) the value of 18 measures was computed for each rule; (ii) each rule received 18 ID's, each one corresponding to its position in one of the ranks related to a measure; (iii) the average was then calculated based on the ranks' position (ID's). Thereby, based on this rank (the higher the better), the number of rules the user would have to analyze, through this ordered list, to search for all $g \in G$, was computed.

[7] In fact, 19 rules were obtained through J.48. However, only 12 were considered, since not all of them were extracted through Apriori due to min-sup and min-conf constraints.

The measures used were Added Value, Certainty Factor, Collective Strength, Confidence, Conviction, IS, ϕ-coefficient, Gini Index, J-Measure, Kappa, Klosgen, λ, Laplace, Lift, Mutual Information (asymmetric), Novelty, Support and Odds Ratio. Details about the measures can be found on [3]. The choice regarding the post-processing approach, used to carry out the comparison, was based on its frequently application in many domains. Besides, it is simple to use and doesn't need any extra information to be processed. Others, like [*Summarization*] through ontologies, require domain specifications, which would imply on domain oriented experiments. However, other types of comparisons must be done in future works.

5 Results and Discussion

Considering the configurations in Table 1, the proposed framework was executed for each data set. Table 2 presents the obtained results. The columns of the table store:

- #rules: number of rules in the rule set R;
- #[I]: number of rules classified as [I] to be exhibited to the user in final of the process (after Step [D]). The pattern X/Y [Z%] indicates: X: number of rules classified as [I] in the end of the execution; Y: number of rules in X explicitly evaluated by the user as [I]; Z: exploration space reduction in relation to the number of rules in R. Looking at Table 2, regarding the Weather-Nominal data set, it can be noticed that: (a) 52 rules were labeled as [I] in the end of the process, in which 23 of them were explicitly classified by the user; (b) this set contains 7.20% of the rules in R and, therefore, the user would obtain a exploration space reduction of 92.80% (100-((52/722)*100));
- #rules evaluated: number of rules explicitly labeled by the user (Step [B]). The number in "[]" also indicates the exploration space reduction in relation to the number of rules in R. Looking at Table 2, regarding the Weather-Nominal data set, it can be noticed that: (a) 100 rules were explicitly labeled by the user; (b) the user would explore 13.85% of the rules in R and, therefore, would obtain a exploration space reduction of 86.15% (100-((100/722)*100));
- #gold rules: number of rules in G found in the end of the process (after Step [D]). In fact, since the stopping criterion occurs when all the set G is found, the pattern X/Y indicates: X: number of gold rules in G; Y: number of rules in X explicitly evaluated by the user. Looking at Table 2, regarding the Weather-Nominal data set, it can be noticed that the 5 rules in G were found, in which 3 of them were explicitly labeled by the user – this means that the other 2 were implicitly classified;
- #iterations: number of iterations executed;
- #explored rules: since the user explicitly labels x rules and, in the end, y rules are returned to him, the total of rules the user explores is: ((#[I] - #$[\mathrm{I}]_E$) + #rules evaluated). Looking at Table 2, regarding the Weather-Nominal data set, the user would explore the 52 rules in the [I] set, minus the 23 rules already seen, plus the 100 rules labeled during Step [B] ((52 - 23) + 100). This number leads to an exploration space reduction of 82.13% (100-((129/722)*100)).

For each data set, there are two lines: the first one regarding the framework results; the second one regarding the results of the traditional post-processing approach. Thereby, based on a rank list built through objective measures, as discussed in Section 4, Table 2 presents the number of rules the user would have to analyze to search for all the rules $g \in G$. Looking at Table 2, regarding the Weather-Nominal data set, it can be noticed that all the 5 gold rules were found after analyzing the first 120 best classified rules. Thereby, in this case, 120 iterations were made, since each iteration represents an explored rule. Besides, "#[I]" = "#rules evaluated" = "#explored rules" since all the seen rules were evaluated and, once they were on a rank list, all of them were considered as [I] until the stopping criterion was reached. The result set leads to an 83.38% (100-((120/722)*100)) exploration space reduction.

Table 2. Results obtained through the proposed framework, and through a traditional post-processing approach, considering Table 1 configurations

Data set	#rules	#[I]	#rules evaluated	#gold rules	#iterations	#explored rules
Weather-Nominal	722	52/23 [92.80%]	100 [86.15%]	5/3	25	129 [82.13%]
		120 [83.38%]		5/5	120	120 [83.38%]▲
Contact-Lenses	890	38/31 [95.73%]	224 [74.83%]	4/3	56	231 [74.04%]▲
		319 [64.16%]		4/4	319	319 [64.16%]
Balloons	772	154/1 [80.05%]	4 [99.48%]	7/0	1	157 [79.66%]▲
		229 [70.34%]		7/7	229	229 [70.34%]
Hayes-Roth	889	102/64 [88.53%]	308 [65.35%]	12/9	77	346 [61.08%]▲
		443 [50.17%]		12/12	443	443 [50.17%]
Groceries	1092	13/13 [98.81%]	488 [55.31%]	7/7	122	488 [55.31%]▲
		1020 [6.59%]		7/7	1020	1020 [6.59%]
Sup	1149	27/27 [97.65%]	328 [71.45%]	9/9	82	328[71.45%]▲
		1146 [0.26%]		9/9	1146	1146 [0.26%]

Evaluating the results, it can be noticed that:

Weather-Nominal. 52 of the 722 rules in R were labeled as [I], in which 23 of them were explicitly evaluated by the user, being 3 of the 23 in the set G. This [I] set leads to an exploration space reduction of 92.80%. In order to find all the 5 gold rules in G, 25 iterations were executed, enforcing the user to explicitly evaluate 100 rules, implying on an exploration space reduction of 86.15%. After labeling 100 rules and exploring 29 of the unseen [I] rules (29=52-23; 129=100+29), an exploration space reduction of 82.13% was obtained. Compared to the rank list, it would be necessary 120 iterations in order to find all the 5 gold rules in G, leading to an exploration space reduction of 83.38%. Thus, in this case, the traditional approach presents a better performance compared to the proposed framework (▲ sign), although near values have been obtained.

Contact-Lenses. 38 of the 890 rules in R were labeled as [I], in which 31 of them were explicitly evaluated by the user, being 3 of the 31 in the set G. This [I] set leads to an exploration space reduction of 95.73%. In order to find all the 4 gold rules in G, 56 iterations were executed, enforcing the user to explicitly evaluate 224 rules, implying on an exploration space reduction of 74.83%. After labeling 224 rules and exploring 7 of the unseen [I] rules (7=38-31; 231=224+7), an exploration space reduction of 74.04% was obtained. Compared to the rank list, it would be necessary 319 iterations in order to find all the 4 gold rules in G, leading to an exploration space reduction of 64.16%. Thus, in this case, the proposed framework presents a better performance compared to the traditional approach (▲ sign).

Balloons. 154 of the 772 rules in R were labeled as [I], in which 1 of them were explicitly evaluated by the user. This [I] set leads to an exploration space reduction of 80.05%. In order to find all the 7 gold rules in G, 1 iteration was executed, enforcing the user to explicitly evaluate 4 rules, implying on an exploration space reduction of 99.48%. After labeling 4 rules and exploring 153 of the unseen [I] rules (153=154-1; 157=4+153), an exploration space reduction of 79.66% was obtained. Compared to the rank list, it would be necessary 229 iterations in order to find all the 7 gold rules in G, leading to an exploration space reduction of 70.34%. Thus, in this case, the proposed framework presents a better performance compared to the traditional approach (▲ sign).

Hayes-Roth. 102 of the 889 rules in R were labeled as [I], in which 64 of them were explicitly evaluated by the user, being 9 of the 64 in the set G. This [I] set leads to an exploration space reduction of 88.53%. In order to find all the 12 gold rules in G, 77 iterations were executed, enforcing the user to explicitly evaluate 308 rules, implying on an exploration space reduction of 65.35%. After labeling 308 rules and exploring 38 of the unseen [I] rules (38=102-64; 346=308+38), an exploration space reduction of 61.08% was obtained. Compared to the rank list, it would be necessary 443 iterations in order to find all the 12 gold rules in G, leading to an exploration space reduction of 50.17%. Thus, in this case, the proposed framework presents a better performance compared to the traditional approach (▲ sign).

Groceries. 13 of the 1092 rules in R were labeled as [I], in which 13 of them were explicitly evaluated by the user, being 7 of the 13 in the set G. This [I] set leads to an exploration space reduction of 98.81%. In order to find all the 7 gold rules in G, 122 iterations were executed, enforcing the user to explicitly evaluate 488 rules, implying on an exploration space reduction of 55.31%. After labeling 488 rules, an exploration space reduction of 55.31% was obtained. Compared to the rank list, it would be necessary 1020 iterations in order to find all the 7 gold rules in G, leading to an exploration space reduction of 6.59%. Thus, in this case, the proposed framework presents a better performance compared to the traditional approach (▲ sign) – an expressive difference was obtained.

Sup. 27 of the 1149 rules in R were labeled as [I], in which 27 of them were explicitly evaluated by the user, being 9 of the 27 in the set G. This [I] set leads to an exploration space reduction of 97.65%. In order to find all the 9 gold rules in G, 82 iterations were executed, enforcing the user to explicitly evaluate 328 rules, implying on an exploration space reduction of 71.45%. After labeling 328 rules, an exploration space reduction of 71.45% was obtained. Compared to the rank list, it would be necessary 1146 iterations in order to find all the 9 gold rules in G, leading to an exploration space reduction of 0.26%. Thus, in this case, the proposed framework presents a better performance compared to the traditional approach (▲ sign) – an expressive difference was obtained.

Summarizing, it can be observed that: (a) the proposed framework presents excellent results regarding the transactional data sets compared to the objective measures approach, as well as good results regarding the relational data sets; (b) in almost all the cases (5 of 6 (83.33%)) the proposed framework presented better performance compared to the objective measures approach. Thereby, as seen through the experiments, since the user is constantly directed to the potentially [I] patterns of the domain, his exploration effort is minimized through a reduction in the exploration space, once his knowledge and/or interests are iteratively propagated.

6 Conclusion

In this paper a post-processing association rules framework, based on SSL, was proposed. The idea was to treat the post-processing phase as a classification task to: (a) automatically propagate the user's knowledge and/or interests over the rule set, minimizing his effort through a reduction in the exploration space; (b) implicitly obtain the user's knowledge and/or interests, through an iterative and interactive process, without using any formalism. Experiments were carried out in order to demonstrate the framework feasibility. It could be noticed that good results are obtained, using as baseline a traditional post-processing approach.

As seen, the proposed framework opens many researches possibilities, since it can be instantiated in many different manners. Many other configurations can be explored in the framework Steps, mainly regarding the SSL methods related to Step [C]. Furthermore, a case study, with real users, has to be done to analyze the process considering other aspects. Finally, other post-processing approaches could be used as baseline to complement the analysis here presented. However, we think this is the first step to a broad area to be explored.

Acknowledgments. We wish to thank CAPES and FAPESP (2013/12392-0) for the financial support.

References

1. Mansingh, G., Osei-Bryson, K., Reichgelt, H.: Using ontologies to facilitate post-processing of association rules by domain experts. Information Sciences 181(3), 419–434 (2011)
2. Marinica, C., Guillet, F.: Knowledge-based interactive postmining of association rules using ontologies. IEEE TKDE 22(6), 784–797 (2010)
3. Guillet, F., Hamilton, H.J.: Quality Measures in Data Mining. SCI, vol. 43. Springer, Heidelberg (2007)
4. Ayres, R.M.J., Santos, M.T.P.: Mining generalized association rules using fuzzy ontologies with context-based similarity. In: Proceedings of the 14th ICEIS, vol. 1, pp. 74–83 (2012)

5. Carvalho, V.O., Rezende, S.O., Castro, M.: Obtaining and evaluating generalized association rules. In: Proceedings of the 9th ICEIS, vol. 2, pp. 310–315 (2007)
6. de Carvalho, V.O., dos Santos, F.F., Rezende, S.O., de Padua, R.: PAR-COM: A new methodology for post-processing association rules. In: Zhang, R., Zhang, J., Zhang, Z., Filipe, J., Cordeiro, J. (eds.) ICEIS 2011. LNBIP, vol. 102, pp. 66–80. Springer, Heidelberg (2012)
7. Berrado, A., Runger, G.C.: Using metarules to organize and group discovered association rules. Data Mining and Knowledge Discovery 14(3), 409–431 (2007)
8. Zhu, X., Goldberg, A.B.: Introduction to Semi-Supervised Learning, vol. (6). Morgan & Claypool Publishers (2009)