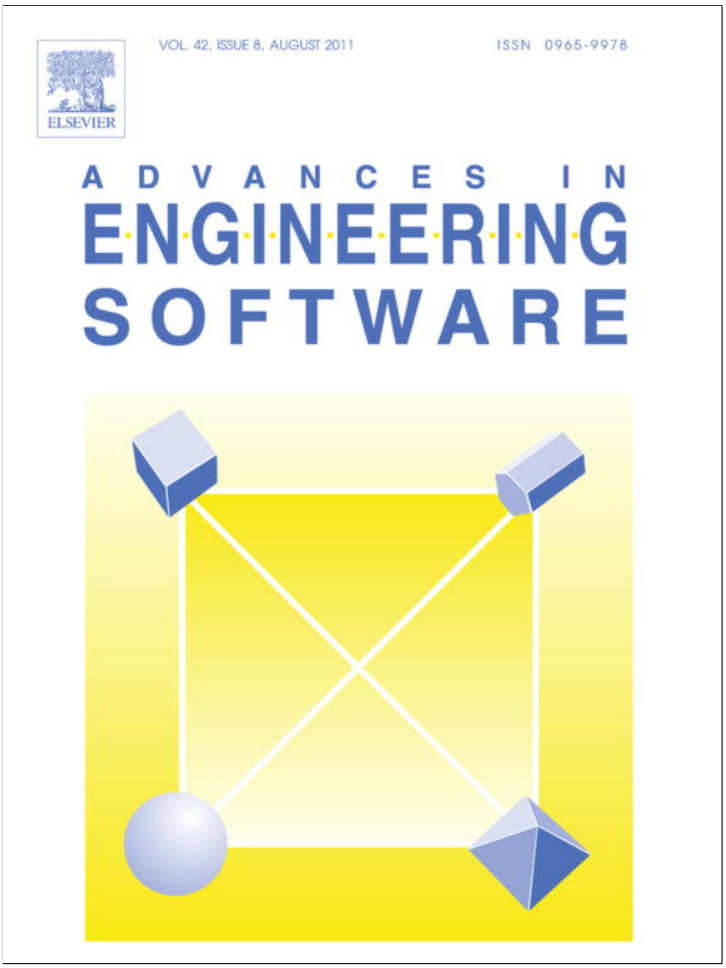


Advances in Engineering Software 42 (2011) 566–576

Contents lists available at ScienceDirect

Advances in Engineering Software

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

RM-Tool: A framework for discovering and evaluating association rules

Cristóbal Romero *, José María Luna, José Raúl Romero, Sebastián Ventura

University of Córdoba, Dept. of Computer Science and Numerical Analysis, 14071 Córdoba, Spain

ARTICLE INFO

Article history:
Received 19 October 2010
Accepted 10 April 2011
Available online 17 May 2011

Keywords:
Association rule mining
Rule evaluation
Rule visualization
Correlation analysis
Principal component analysis
Clustering

ABSTRACT

Nowadays, there are a great number of both specific and general data mining tools available to carry out association rule mining. However, it is necessary to use several of these tools in order to obtain only the most interesting and useful rules for a given problem and dataset. To resolve this drawback, this paper describes a fully integrated framework to help in the discovery and evaluation of association rules. Using this tool, any data mining user can easily discover, filter, visualize, evaluate and compare rules by following a helpful and practical guided process described in this paper. The paper also explains the results obtained using a sample public dataset.

1. Introduction

Association rule mining (ARM) is one of the most popular and well-known Data Mining (DM) techniques for discovering relationships and correlations among attributes from large datasets. ARM produces IF-THEN statements composed of *attribute-value* pairs known as itemsets [13]. Initial researches were largely motivated by the market basket analysis and have served as starting point for many other different domains including epidemiology, clinical medicine, fluid dynamics, astrophysics, counter-terrorism, education, etc., i.e. those areas in which the relationship between items can provide knowledge of interest to human users.

Although ARM is a widely used DM technique, it also presents some problems or drawbacks such as: (i) the discovery of a large number of association rules; (ii) not all the discovered rules are relevant or interesting; and (iii) a high execution time and memory size is required. Over the past decade a variety of algorithms have been developed to address these issues through the refinement of search and prune strategies, and the use of alternative data structures and dataset organizations [13]. In fact, most of the research efforts were initially devoted to improve algorithmic performance and, in the second place, to reduce the output set by allowing the possibility of imposing constraints on the desired results. On the other hand, it is very important to evaluate and post-mine the rules obtained in order to find the most interesting rules for each specific problem. In this sense, the use of objective interestingness measures has been the traditional suggestion [29], although subjective measures are becoming increasingly important [27], i.e., measures that are based on subjective factors controlled by the user. Most of the subjective approaches involve user participation in order to obtain the most interesting rules based on users previous knowledge. Finally, a factor that is of major importance in determining the quality of the rules extracted is their comprehensibility. This aspect of rule quality is often overlooked due to the subjective nature of comprehensibility, which cannot be measured independently of the user using the system [17]. Some basic techniques have been proposed to improve the comprehensibility of discovered rules, such as, reducing the resulting set size, constraining the number of items and which specific items are in the antecedent or consequent of the rule, or both [8]. Another way to improve the comprehensibility of the rules is to incorporate domain knowledge and semantics, and use terminology that is common and well-known to the user [9]. A different approach that has also been used to facilitate the comprehensibility of discovered rules is the visualization of the corpus of the rules extracted in a graph mode [22]. ARM have been integrated with visualization techniques in order to allow users to drive the association rule finding process, giving them control and visual cues to facilitate the understanding of both the process and its results.

Because of the aforementioned issues, several different tools may be necessary to perform each different subtask: one tool for discovering rules, another for evaluating the rules obtained, yet another for visualizing rules, and so on. Therefore, in this paper we propose and explore an integrated, fully functional and scalable association rule mining framework, called RM-Tool (rule mining tool), which lets the user do all these tasks within the same environment. The paper is organized as follows: Section 2 briefly

* Corresponding author. Tel.: +34 957212257.
E-mail address: cromero@uco.es (C. Romero).

0965-9978/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.advengsoft.2011.04.005

reviews existing ARM tools and introduces RM-Tool; Section 3 describes RM-Tool features; Section 4 shows the process of discovering association rules using RM-Tool; Section 5 describes in a practical and tutorial way how the discovered rules can be evaluated; and finally, Section 6 outlines some concluding remarks and future research lines.

2. Background for ARM and tools

Association rule mining has received considerable attention over the last decade since the Apriori algorithm [1] was published and subsequently improved upon by the appearance of a lot of other new algorithms. The task involving the discovery of association rules is usually divided into two main subtasks [13]: firstly, to find those itemsets whose occurrences exceed a predefined support threshold, i.e., frequent itemsets (aka. large itemsets); secondly, to generate association rules from those large itemsets that are constrained by another threshold, such as a minimal confidence. Frequent association rules are the most popular and well researched method for discovering interesting relationships. However, other different types of association rules can be distinguished such as infrequent and class association rules.

- **Frequent** association rules [1] are implications of the form $A \rightarrow C$, where A and C are disjoint sets of items, that satisfy a user-specified minimum support and a minimum confidence at the same time. The support of the rule is the proportion of the number of transactions including A and C in a dataset. The confidence of an association rule is the proportion of transactions containing A which also contains C.
- **Infrequent** rules are also known as rare, unusual, exceptional or sporadic association rules [20]. They are similar to the concept of frequent association rule but have low support and high confidence in contrast to frequent association rules which are determined by a high support and confidence levels. Infrequent itemsets are those that only appear together in very few transactions or some very small percentage of transactions in datasets.
- **Class** association rules [21] are a special type of association rule that describe an implicative co-occurring relationship between a set of items and a predefined class. This type of association rule is expressed as follows: "IF antecedent (input-attributes) THEN consequent (class)". So, class association rules are a type of target-constraint association rule which has one and only one predetermined target, i.e., the class. This type of *focused* rule-mining leads to a set of independent and comprehensible rules that have one (desired) element in the consequent.

Nowadays, there are a great number of specific tools and DM software that carry out association rule mining. Some examples of commercial ARM software are Magnum Opus[1] and WizRule[2]; free association software include ARMiner[3], ARTool[4] or DM-II[5]; research specific proposals are CIECoF [12], EPRules [26] and MIRAGE [31]; and other publicly available and general purpose software for DM are KEEL[6], Rapid Miner[7] and Weka[8]. Table 1 compares some of their main capabilities such as the types of rules discovered, and if they provide filtering, visualization, evaluation and Predictive Model Markup Language (PMML) output; versus to those of RM-Tool software proposed in this paper.

Table 1
Comparison of association rule mining tools.

Tool	Type of rules (Num. of Alg.)	Rule filtering	Visualization of rules	Evaluation of rules	PMML output
ARMiner	Frequent (2)	Yes	No	Yes	No
ARtool	Frequent (5)	Yes	No	Yes	No
CIECoF	Frequent (1)	Yes	No	Yes	Yes
DM-II	Class (1)	No	No	Yes	No
EPRules	Class (1)	Yes	No	Yes	No
KEEL	Frequent (4) Class (4)	No	No	Yes	No
Magnum Opus	Frequent (1)	Yes	No	Yes	No
MIRAGE	Frequent (2)	No	Yes	No	No
Rapid Miner	Frequent (5)	No	No	Yes	No
Weka	Frequent (4) Class (1)	No	No	Yes	No
WizRule	Frequent (1)	Yes	No	No	No
RM-Tool	Frequent (5) Infrequent (3) Class (2)	Yes	Yes	Yes	Yes

As shown in Table 1, RM-Tool is the only tool that offers all capabilities. In fact, our tool provides a great number of ARM algorithms implemented (10 in total) to discover different types of rules (frequent or traditional, class or classification, and even infrequent or rare). It allows specifying filters before and after the algorithm is applied (a priori and a posteriori); whereas other tools usually only allow filtering before, but not after. It is one of the few tools that show the rules not only in a table but also in a graph, and the only one that also visualizes the ontology of the related domain. It provides not only rule evaluation measures like the other tools but also defines new evaluation measures and other evaluation techniques including correlation analysis, Principal Component Analysis (PCA) and clustering. It is the only tool that lets users indicate the specific rules that are useful in resolving their problems. Furthermore, it is one of the few tools that generate a PMML file in XML-based language which provides a way for applications to define statistical and data mining models and share models between compliant DM systems. So, RM-Tool is the most complete ARM framework of all the current tools available, allowing the user to discover, visualize, filter and evaluate association rules in the same environment.

3. General description of RM-Tool

RM-Tool has been developed in Java using Swing and Java Internationalization API. It consists of two main components (see Fig. 1):

- **Rule discovering:** The aim of this component is to discover a set of rules using ARM algorithms. It is possible to configure the execution of the algorithms by providing some parameters. Firstly, data sources are accepted in one of the following formats: CSV (Comma-Separated Value) used by most of the spreadsheets and DM tools, DAT (Data file) used by KEEL [2], or ARFF (Attribute-Relation File Format) used by Weka [30]. Next, users need to select and configure one ARM algorithm out of all those available and they can also specify some filters in order to constrain the rules generated. Finally, ARM algorithms can be executed either locally on the same computer or on a remote computer connected to the Internet, where the algorithm server application is running. Then, the association rules obtained are automatically saved both in a PMML file

[1] http://www.giwebb.com/.
[2] http://www.wizsoft.com/.
[3] http://www.cs.umb.edu/~laur/ARMiner/.
[4] http://www.cs.umb.edu/~laur/ARtool/.
[5] http://www.comp.nus.edu.sg/~dm2/.
[6] http://www.keel.es/.
[7] http://rapid-i.com/.
[8] http://www.cs.waikato.ac.nz/ml/weka/.

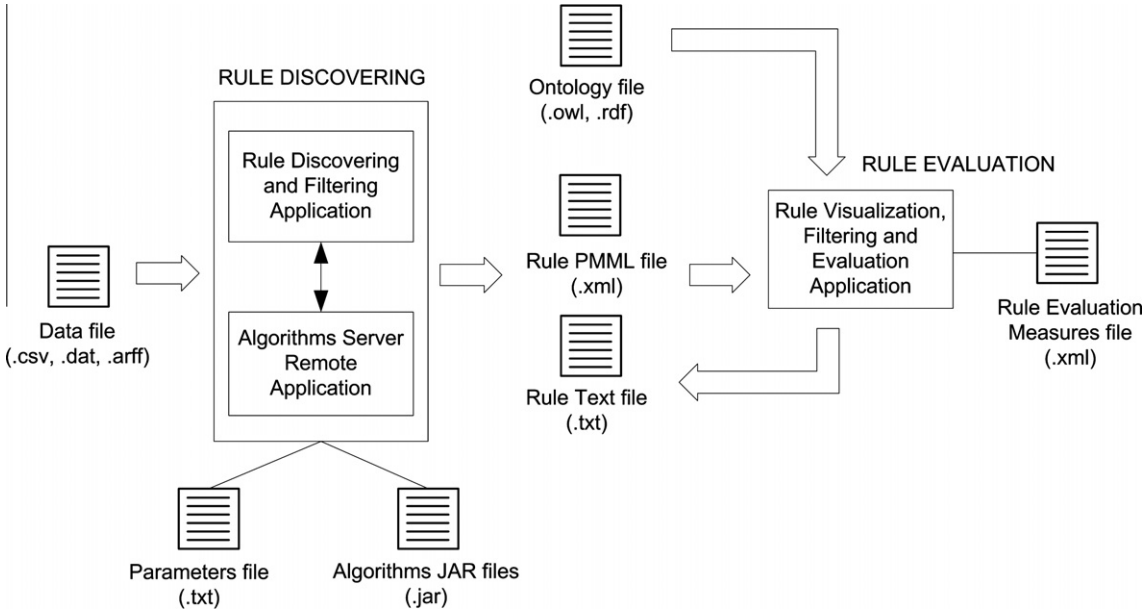


Fig. 1. General functional flow of the RM-Tool.

and in a text file. It is important to notice that new algorithms can be incorporated into the application easily by simply providing all their Java classes zipped in a JAR (Java ARchive) file and using a predefined text configuration file.

- **Rule evaluation:** The aim of this component is to evaluate the set of previously discovered rules. Users only have to provide a PMML file with the association rule model obtained, but they can also provide a file in RDF (Resource Description Framework) or OWL (Ontology Web Language) format containing the description of the ontology related to the data domain. The user can specify and add new filters to the rules and select previously defined rule evaluation measures or define new ones that will be saved in a XML (eXtensible Markup Language) file. Then, all the rules are shown along with the evaluation measures as a table, or they can also be visualized graphically with their related ontology. Starting from this table of rules and measures, some evaluation techniques that can be applied include the correlation of measures, the PCA of measures and a clustering of the rules. Finally, users can also select which are the most useful and interesting rules for them and this information is saved in the rules file so that it can be shared with other users.

In the next sections, these two applications are described in more detail.

4. Discovering association rules

RM-Tool allows the execution of several ARM algorithms to discover three different types of association rules, providing a GUI that is simple and easy to use (see Fig. 2).

Fig. 2 shows the interface for executing and configuring ARM algorithms. RM-Tool uses an INI (INItialization) file in order to pass all the parameters introduced by the user in the GUI (see Fig. 2) to the algorithm. The INI file format is a de facto standard for text format configuration files with a simple basic structure in which each line contains a name-value pair, delimited by an equal sign (name = value). The RM-Tool configuration file stores the following parameters: *DataFile* (full path of the input data file), *OutputFile* (full path of the rule output file), *MinSupport* (minimum support threshold), *MinConfidence* (minimum confidence threshold), *MaxAntecedent* (maximum number of items on rule antecedent), *MaxConsequent* (maximum number of items on rule consequent) and *NumRules* (maximum number of rules discovered). Users can also select some (a priori) filters to constrain the generated rules. In fact, users can specify if an attribute or a specific value of an attribute should or should not occur in the antecedent or consequent of the rules. Finally, users also indicate whether they want the algorithm to be executed locally or remotely by providing the IP (Internet Protocol) address and the port where the application server is located.

After clicking the "*Run*" button, the algorithm selected is executed and two new files comprising the rules mined are generated: the file in PMML format and another file in text format, which is displayed on the screen. As an example, Fig. 3 shows the rules obtained when executing the Apriori algorithm using the configuration shown in Fig. 2 and the weather[9] dataset. This is a publicly available dataset that shows several situations where the weather is or is not suitable to play sports, depending on the current forecast, temperature, humidity and wind. It has 14 instances with five nominal attributes (outlook, temperature, humidity, windiness and play).

Fig. 3 shows the total number of frequent itemsets; the total number of rules generated and pruned; the rules mined and the execution time. The information provided for each rule is the antecedent and consequent in the form of IF-THEN rules while the values of the support, confidence and lift measures are in brackets. For example, the first rule shows that if today is a sunny day but the humidity is high then the recommendation is not to play sport at a low support value (21% of instances), high confidence (100% of instances) and high lift (2.8 value).

Nowadays, RM-Tool provides the following 10 ARM algorithms: Apriori [1], Apriori-Infrequent [25], Apriori-Rare or Arima [28], Apriori-Inverse [19], AprioriT [5], DIC [4], FP-Growth [16], TFP [6], CBA [21] and TFPC [7]. The following sections will show the performances and some results obtained with these algorithms using weather dataset. All the algorithms have been executed using two configurations (two different values of the support threshold and the same confidence). Table 2 shows the result obtained which shows the name of each algorithm, the type of rules each algorithm discovers, the support and confidence thresholds,

[9] http://www.hakank.org/weka/weather.nominal.arff.

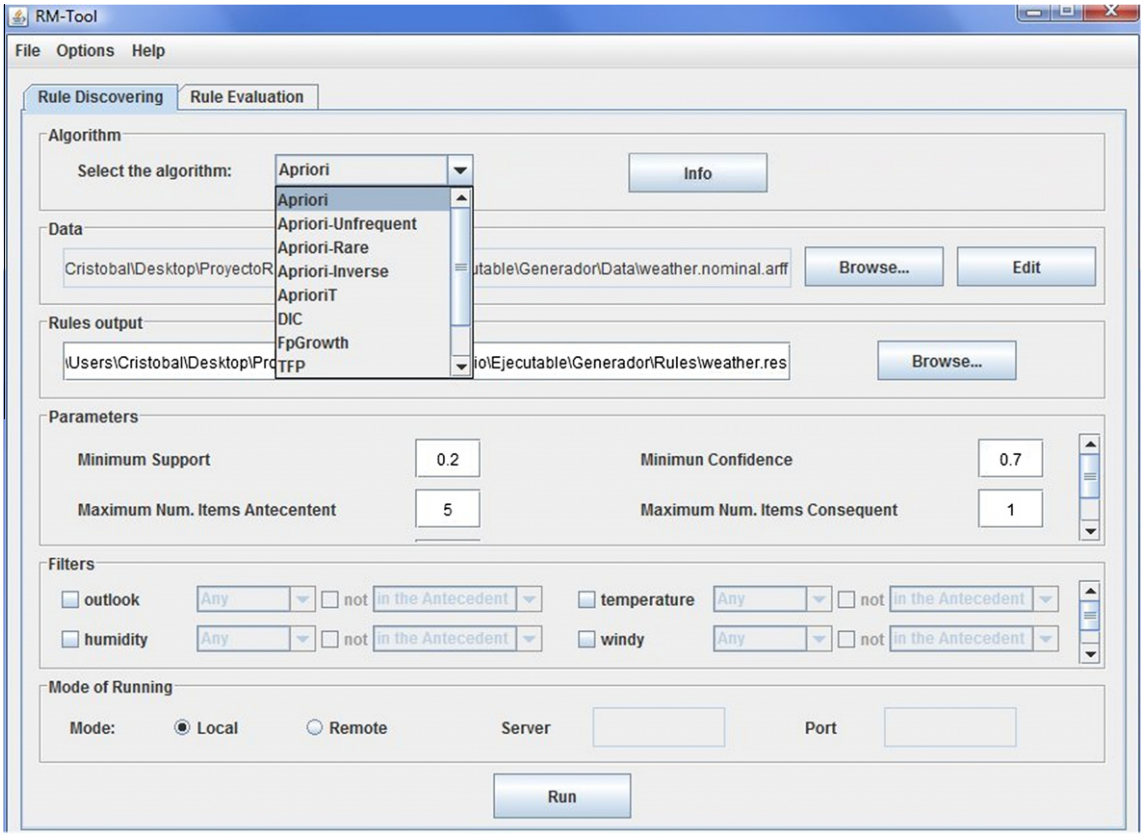


Fig. 2. Rule discovering window.

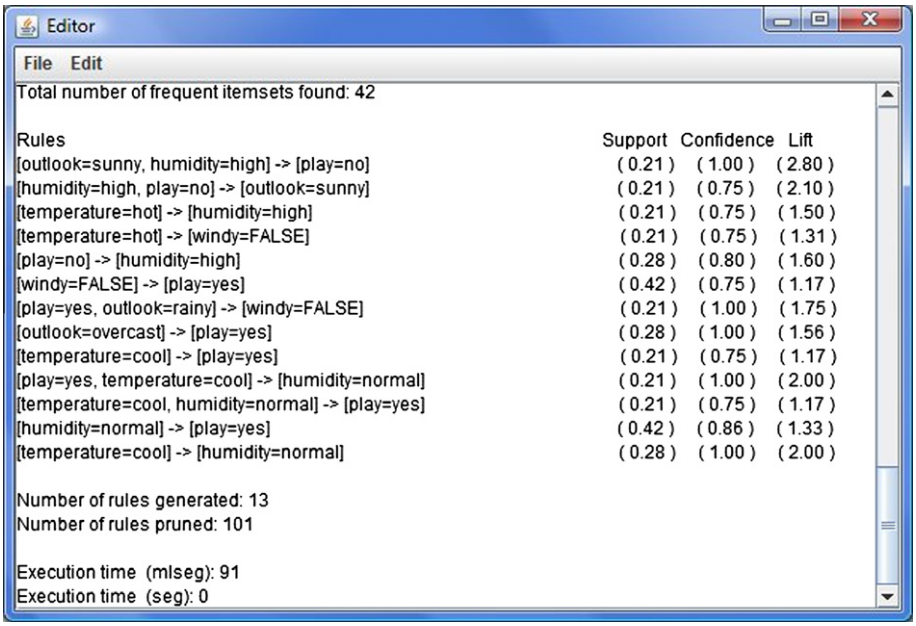


Fig. 3. Sample rules file in text format.

the number of frequent/infrequent itemsets obtained, the number of rules discovered, and the total execution time in milliseconds.

Table 2 shows that for all the algorithms and using these parameters, generally a small number of rules are discovered in a limited time. It can also be observed that infrequent ARM algorithms discover a lower number of rules and use fewer itemsets. Notice that Apriori-Rare discovers one rule more than the other proposals although it uses more execution time. As for frequent ARM algorithms, all discover exactly the same number of rules and also use the same number of frequent itemsets. The only difference found is the execution time which shows DIC to be the fastest and Apriori the slowest. Finally, the two class ARM algorithms obtain very different results. In fact, CBA uses more frequent itemsets but discovers fewer rules than TFPC.

Table 2
Results obtained when executing all the ARM algorithms for two different configurations.

Algorithm	Type of rules	Thresholds (Sup/Conf)	No. of frequent/ infreq. itemsets	No. of rules	Execution time (ms)
Apriori-Infrequent	Infrequent	0.4/0.7	13	2	55
Apriori-Infrequent	Infrequent	0.1/0.7	0	0	41
Apriori-Rare	Infrequent	0.4/0.7	10	3	60
Apriori-Rare	Infrequent	0.1/0.7	0	0	104
Apriori-Inverse	Infrequent	0.4/0.7	13	2	56
Apriori-Inverse	Infrequent	0.1/0.7	0	0	42
Apriori	Frequent	0.4/0.7	8	2	57
Apriori	Frequent	0.1/0.7	104	67	108
AprioriT	Frequent	0.4/0.7	8	2	17
AprioriT	Frequent	0.1/0.7	104	67	40
DIC	Frequent	0.4/0.7	8	2	16
DIC	Frequent	0.1/0.7	104	67	24
Fp-Growth	Frequent	0.4/0.7	8	2	19
Fp-Growth	Frequent	0.1/0.7	104	67	28
TFP	Frequent	0.4/0.7	8	2	18
TFP	Frequent	0.1/0.7	104	67	30
CBA	Class	0.4/0.7	13	2	18
CBA	Class	0.1/0.7	158	3	29
TFPC	Class	0.4/0.7	9	4	17
TFPC	Class	0.1/0.7	67	16	29

5. Evaluating association rules

The rules mined by a data mining method should be interesting, novel and useful for the end-user [3]. In order to evaluate the rules discovered, RM-Tool provides such different techniques as filtering, visualization and measures for rule evaluation.

- **Rule evaluation measures** have been used for measuring the interestingness of the rules discovered [14]. These measures are used to rank and select the rules with the highest values in some specific measures depending on their potential interest for the user [29]. Many rule evaluation measures [29] originate in different areas such as machine learning, data mining, statistics, classification, information theory, information retrieval, etc. These measures can be further categorized into three different types, namely subjective, objective and semantic-based measures [14]. RM-Tool only provides objective measures that are data-directed, based on probability and can be obtained from the contingency table, which stores the frequency count that satisfies a given predicate. Table 3 shows the contingency table of a generic rule $A \rightarrow C$, where A is the antecedent of the rule, and C is the consequence of the rule. Both A and C comprises itemsets where each item is a pair *attribute* = *value*. A and C being disjoint sets.
- **Posteriori filtering** is a simple methodology for highlighting the strongest discovered rules. A filter can be defined as a condition that the discovered rules have to fulfill [15]. If a rule does not match the condition, then the rule is discarded. RM-Tool implements two types of filters, one about the form of the rule and the other one about its evaluation measures.
- **Visualization techniques** can help users to understand the data and also to reveal the most interesting associations and patterns to be found in the data [22]. Methodologies developed to visualize association rules are both table and graph based. Table-based techniques are the most common and simple approach to represent association rules in the form of a table where each row represents an association rule and each column the different measures defined (support, confidence, lift, etc.). Graph-based techniques use nodes and edges to represent the associations of items in the rules. For example, the rule $A \rightarrow C$ is represented by a directed graph in which A and C are the nodes. The edge connecting A and C is the arrow pointing from the antecedent to the consequent of the rule. RM-Tool uses both table-based and graph-based techniques for visualizing rules.

Table 3
Contingency table, where $n(X)$ denotes the number of records that satisfy X, and N denotes the total number of records.

	A	$\neg A$	
C	$n(AC)$	$n(\neg AC)$	$n(C)$
$\neg C$	$n(A\neg C)$	$n(\neg A\neg C)$	$n(\neg C)$
	$n(A)$	$n(\neg A)$	N

5.1. Rule evaluation

RM-Tool provides over 30 rule evaluation measures. Table 4 shows some of the most frequent objective measures for rules [14,29]. Notice that the expressions for calculating all these measures can be obtained in the contingency table.

In general, most current ARM tools only provide some of these predetermined evaluation measures. Usually, they just show the user the discovered rules in table-based mode together with the two traditional evaluation rule measures (i.e. support and confidence). As a result users can not specify additional (a posteriori) filters, visualize the rules in a graph-based mode, use other different measures and compare or analyze them. Because of this, it can be very hard for a non-expert user of DM to select the most interesting rules among all the ones obtained using this type of common

Table 4
Examples of rule evaluation measures, where $P(X)$ denotes the probability of X; $P(XY)$ denotes the relative frequency of the intersection of X and Y; and $P(X/Y)$ denotes the conditional probability of X given Y.

Measure	Expression
Confidence	$Confidence(A \rightarrow C) = p(C/A) = \frac{p(CA)}{p(A)}$
Informativity	$Informativity(A \rightarrow C) = -\log_2(C/A)$
Interest or Lift	$Interest(A \rightarrow C) = \frac{p(CA)}{p(C)p(A)}$
Interestingness	$Interestingness(A \rightarrow C) = \sqrt{Interest(A \rightarrow C)^* \frac{p(CA)}{N}}$
Interest function	$Interest\ Function(A \rightarrow C) = p(CA) - p(C)p(A)$
Laplace	$Laplace(A \rightarrow C) = \frac{Support(A \rightarrow C)+1}{Support(A)+2}$
Leverage	$Leverage(A \rightarrow C) = p(C/A) - p(A)^*p(C)$
Novelty	$Novelty(A \rightarrow C) = p(AC) - p(C)^*p(A)$
Support	$Support(A \rightarrow C) = p(AC) = \frac{n(AC)}{N}$
Weighted relative accuracy	$WeightedRelAcc(A \rightarrow C) = p(A)^*(p(C/A) - p(C))$

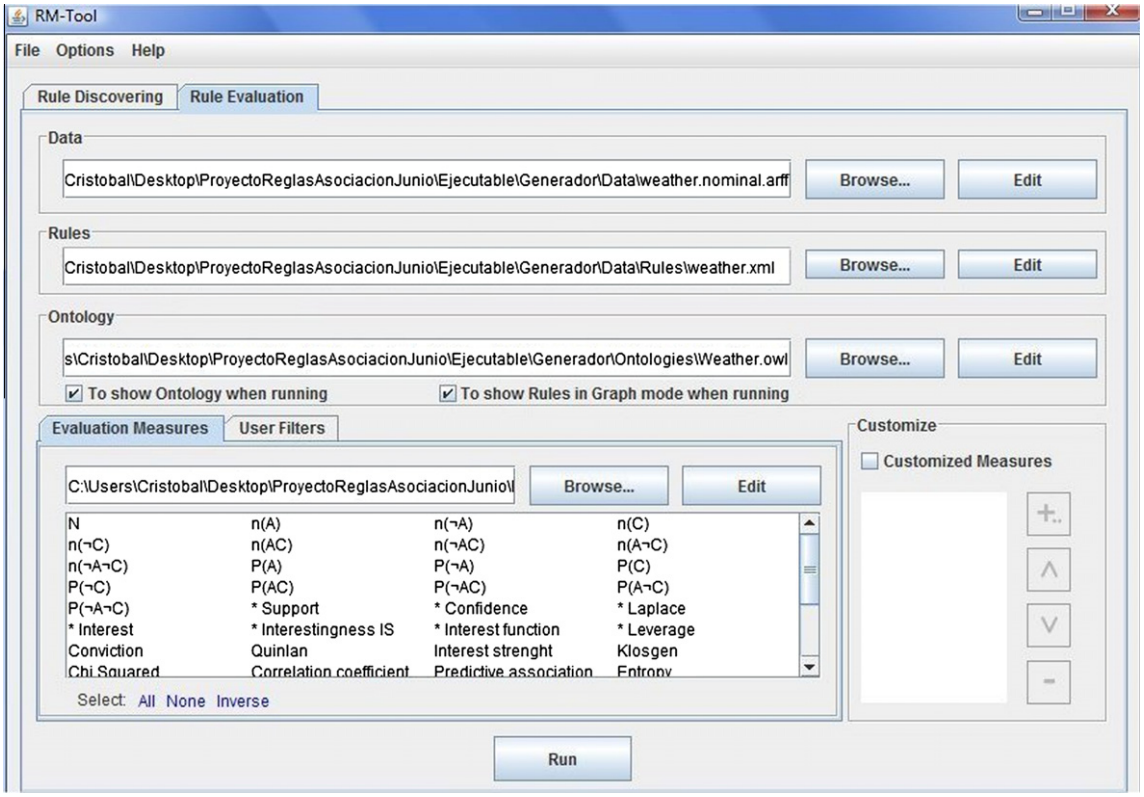


Fig. 4. Rule evaluation window.

ARM tool. On the other hand, RM-Tool provides a rule evaluation GUI to help the user to carry out all these evaluation tasks.

More specifically, Fig. 4 shows the RM-Tool user interface for evaluating rules. Observe that the user should firstly select/provide two mandatory files (data and rules) and one additional optional file (ontology). The data file is a local file or URL (Uniform Resource Locator) that contains the RM-Tool dataset in CSV, Weka or Keel format. A second file is the rules file, which is also a local file or URL that contains the rules discovered by the mining algorithm. This file should be submitted in PMML format. Finally, the measures description file is a local file that uses a predefined XML format and maintains the definition of all the evaluation measures as Latex equations, which are composed of different mathematical Latex symbols, functions, and probabilistic and contingency table symbols, such as $n(A)$, $n(C)$, N, $P(A)$, or $P(C/A)$. Predefined measures are also used in order to specify new user-defined measures. For example, we could add the definition of the *Conditional Support* $(A \rightarrow C) = \frac{n(AC)}{n(C)}$, which is a common measure for imbalanced data when using class association rules [32], just by adding the next code/paragraph:

Box 1

```
<measure>
<name>Conditional Support</name>
<description>Measure of imbalanced data</
description>
<equation>$ConditionalSupport = \frac{n(AC)}{n(C)}$</equation>
</measure>
```

We have also developed a wizard and editor in order not to have to write manually in this XML file, since it might be a difficult task for users who are not expert in the specific Latex equation format. In this way, users can define brand-new measures more easily by simply following the steps indicated by the wizard (see Fig. 5). In the first step, users select all the predefined measures they want to use to define the new measure. Then, an editor (Fig. 5 at left) allows the users to define new measures starting from the contingency table symbols and the predefined measures. The interface is intentionally similar to a calculator, where buttons are conceived to select numbers and operations, and a list of elements is also available to select probabilistic symbols and measures. Finally, the wizard (Fig. 5 at right) asks for the name of the new measure and a brief description, and then the new measure is saved into the XML description file.

5.2. Rule filtering

Users can also filter the discovered rules by specifying a maximum or minimum threshold value on any of the evaluation measures; or by selecting only those rules that contain some specific attributes or values either in the antecedent or in the consequent of the rules.

Fig. 6 shows an example of filters specified by a user and applied to the discovered rules. In this case, rules with only one element in the consequent that contain the attribute *play* in the consequent and have a value of interest greater than or equal to 1 will be obtained.

5.3. Rule visualization

As mentioned previously, rules can be shown in a graph mode together with the ontology of the related domain (see Fig. 7). In this last case, the user has to provide or create the ontology in OWL or RDF format. The free and open-source Protege-OWL editor[10] has been used to show and edit an ontology file (see Fig. 7, left).

[10] http://protege.stanford.edu.

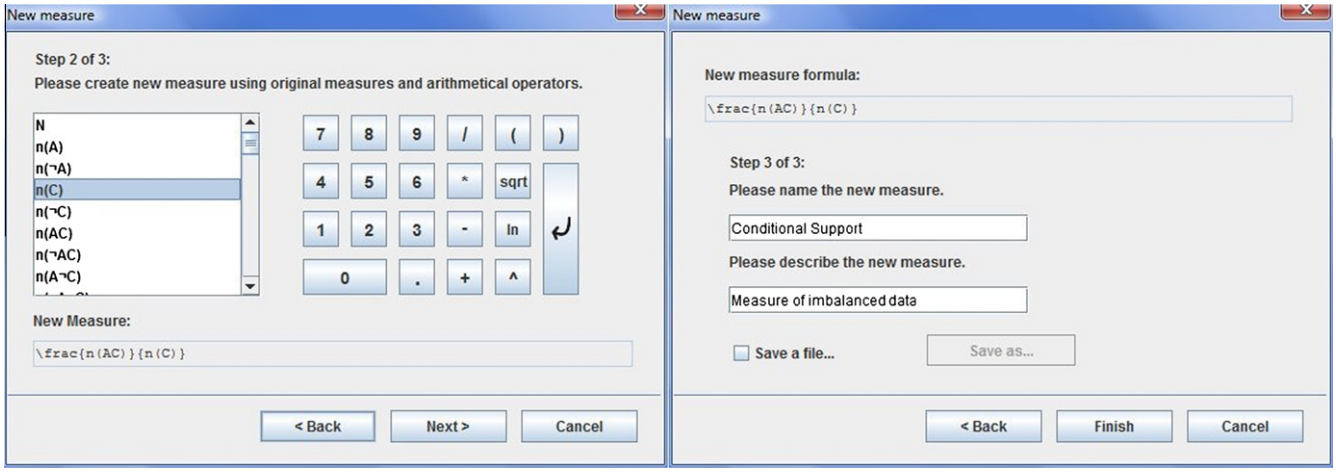


Fig. 5. Wizard and editor windows for defining new rule evaluation measures.

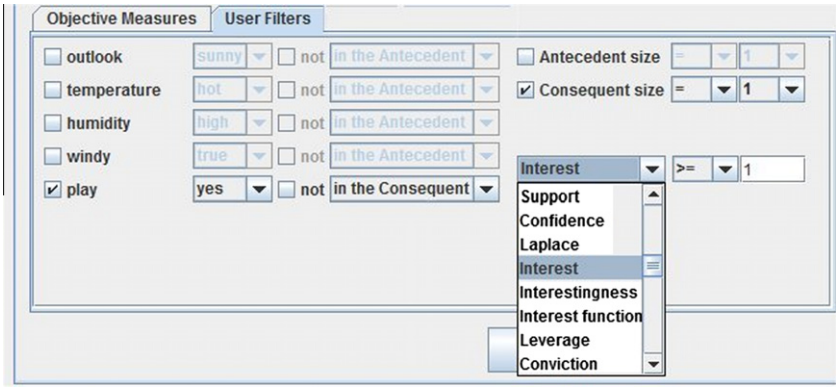


Fig. 6. A posteriori user specified filters.

CLOVER [11] was used to draw a graph-based visualization of the rules. It represents graphs using hierarchic clustering, by means of a simple yet effective rule-based algorithm. CLOVER has been adapted to show rules in the following way (see Fig. 7, right): nodes are frequent itemsets, the directed edges (arrows) indicate the rule or IF-THEN relationship, the edge thickness varies according to the support value of each rule and the confidence of each rule is shown in text mode at the side of the edge.

Back to the case study again, Fig. 7 on the left shows an example of the weather ontology that has three objects/classes: *Atmospheric phenomenon*, *Weather* (which has four properties: *Temperature*, *Humidity*, *Wind* and *Visibility*) and *Outlook* (with three properties: *Sunny*, *OverCast* and *Rainy*). Fig. 7 on the right shows the rules or IF-THEN relationships between frequent itemsets in a graph. It is interesting to note that this graph clearly shows some hidden information about the rules that is not so easy to detect in the text of the rules. For example, the itemset *play* = *yes* is like a drain node (it appears in the consequent of a great number of rules) and almost all the rules are related to them with the exception of the rule IF *humidity* = *high* AND *play* = *no* THEN *outlook* = *sunny*.

RM-Tool always shows the resulting rules in a table, as shown in Fig. 8, comprising the following information for each rule: whether the rule is useful to the user, the number of the rule, the antecedent and consequent elements of the rule, as well as all the values of each evaluation measure.

As we can see in Fig. 8, users can indicate which specific rules are really useful for them by simply selecting the corresponding checkboxes and then providing some explanatory text for the specific usefulness of each rule. This text might be helpful for sharing information with other users. Rules that have been selected as being useful for other users are also shown in different colors, indicating a different number of users. In fact, notice that in Fig. 8 there are only three rules that were previously selected as useful. It is interesting to observe that these are the only rules that have the item *play* = *no*, i.e., these rules serve to predict when the weather is not appropriate for playing any sport. Moreover, the rules can be sorted by any of the measures by simply clicking in the header of a specific column in order to compare different rankings depending on the measure used. Finally, users can also save the table (comprised by rules and measures) in a text file, and so other evaluation techniques can be applied to this table.

5.4. Other evaluation techniques

Starting from the previously obtained table (see Fig. 8), which is formed by rules and measures, RM-Tool allows other evaluation techniques to be applied, such as: correlation analysis, principal component analysis, and clustering.

5.4.1. Correlation analysis

The correlation analysis or correlation coefficient indicator measures the statistical relationship between two variables. There are several correlation coefficients, often denoted by ρ or r, measuring the degree of correlation. The most popular one is the Pear-

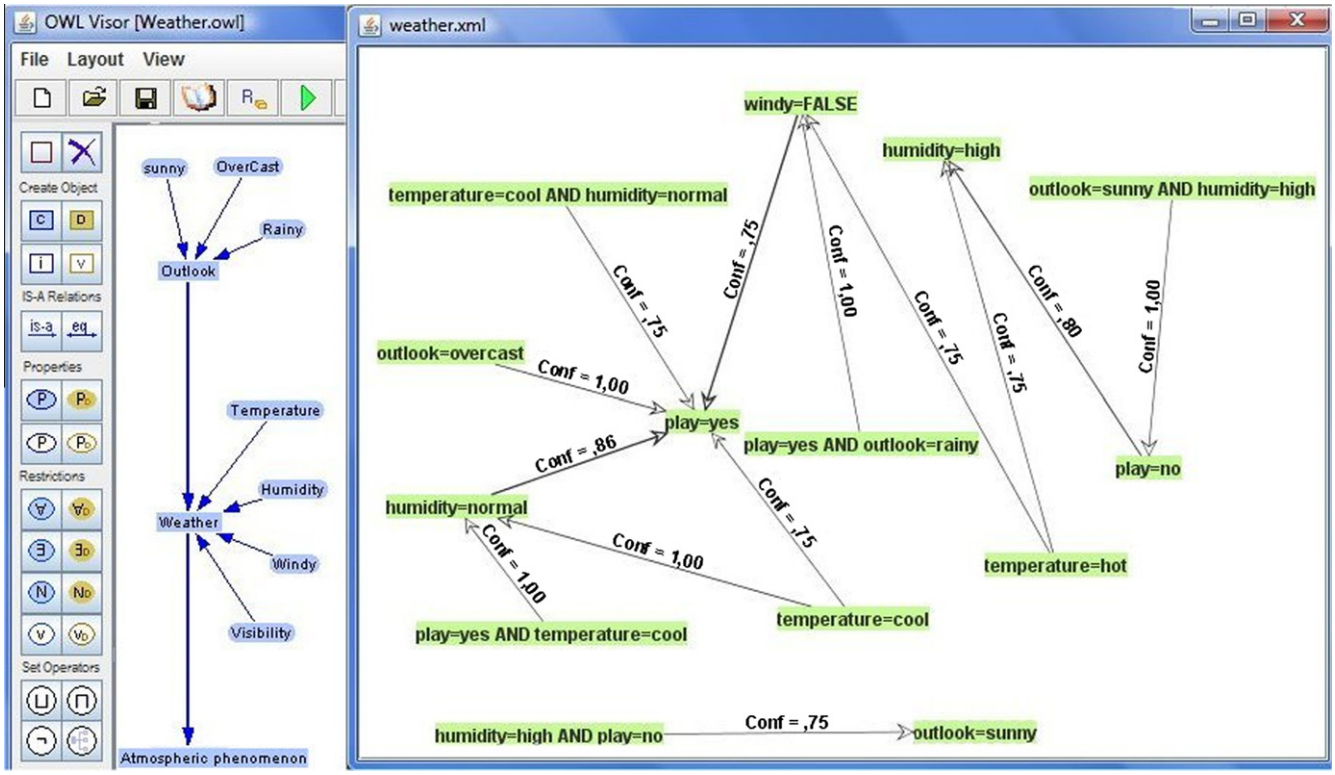
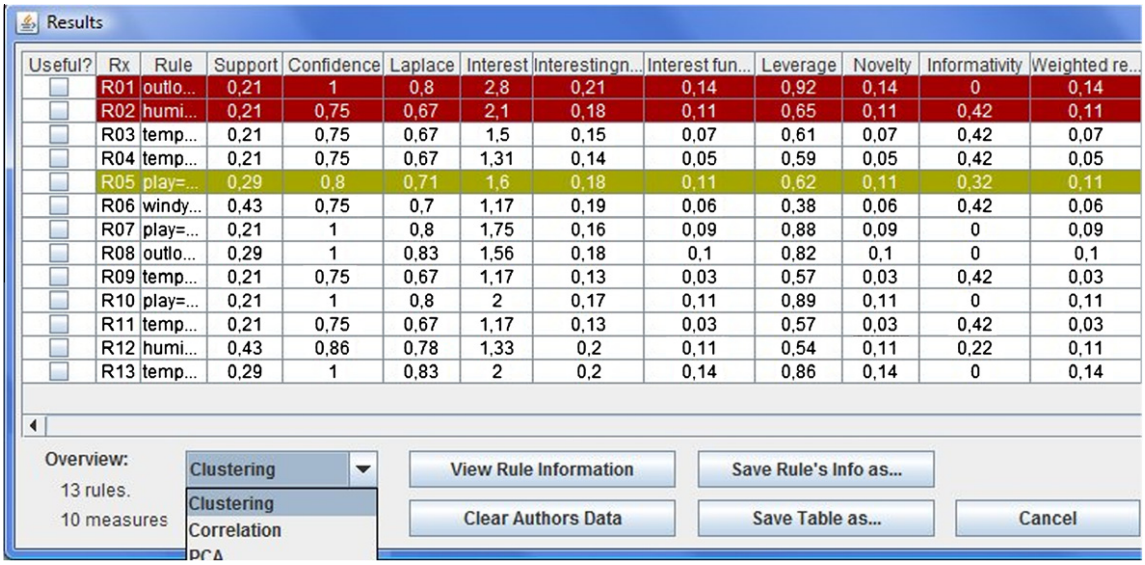


Fig. 7. Ontology and rule graph visualization windows.



Useful?	Rx	Rule	Support	Confidence	Laplace	Interest	Interestingn...	Interest fun...	Leverage	Novelty	Informativity	Weighted re...
	R01	outlo...	0,21	1	0,8	2,8	0,21	0,14	0,92	0,14	0	0,14
	R02	humi...	0,21	0,75	0,67	2,1	0,18	0,11	0,65	0,11	0,42	0,11
	R03	temp...	0,21	0,75	0,67	1,5	0,15	0,07	0,61	0,07	0,42	0,07
	R04	temp...	0,21	0,75	0,67	1,31	0,14	0,05	0,59	0,05	0,42	0,05
	R05	play=...	0,29	0,8	0,71	1,6	0,18	0,11	0,62	0,11	0,32	0,11
	R06	windy...	0,43	0,75	0,7	1,17	0,19	0,06	0,38	0,06	0,42	0,06
	R07	play=...	0,21	1	0,8	1,75	0,16	0,09	0,88	0,09	0	0,09
	R08	outlo...	0,29	1	0,83	1,56	0,18	0,1	0,82	0,1	0	0,1
	R09	temp...	0,21	0,75	0,67	1,17	0,13	0,03	0,57	0,03	0,42	0,03
	R10	play=...	0,21	1	0,8	2	0,17	0,11	0,89	0,11	0	0,11
	R11	temp...	0,21	0,75	0,67	1,17	0,13	0,03	0,57	0,03	0,42	0,03
	R12	humi...	0,43	0,86	0,78	1,33	0,2	0,11	0,54	0,11	0,22	0,11
	R13	temp...	0,29	1	0,83	2	0,2	0,14	0,86	0,14	0	0,14

Fig. 8. Results windows with rules, measures and some evaluation techniques.

son correlation coefficient, which is sensitive mainly to a linear relationship between two variables [24]. A coefficient of +1 means that the dependent variable will always move in step with the independent variable; a coefficient of −1 indicates that the dependent variable will always move opposite to the independent variable; and a coefficient of 0 means that there is no relationship between the movements of the two variables. RM-Tool implements a correlation matrix using the Pearson correlation coefficient between all the evaluation measures, as shown in Fig. 9. Notice that the correlation matrix of N random variables $X_1, \ldots, X_n$ is the $N \times N$ matrix whose i,j entry is the correlation between the variables X_i and X_j.

Fig. 9 shows the existing correlation between values of the 10 previously defined measures for the 13 rules discovered. As can be noted, there are several positively correlated measures such as Laplace-Confidence, Confidence-Laplace, Weighted relative accuracy-Novelty, Interest function-Weighted relative accuracy, and Novelty-Weighted relative accuracy; and several negatively correlated measures such as Informativity-Confidence, Informativity-Laplace, Confidence-Informativity, and Laplace-Informativity. So,

Correlation Matrix

	Support	Confidence	Laplace	Interest	Interestingn...	Interest fun...	Leverage	Novelty	Informativity	Weighted rel...
Support	1.0	-0.06406...	0.1864...	-0.345...	0.5435496...	0.1198970...	-0.4853...	0.119...	0.050183...	0.11989706...
Confidence	-0.064...	1.0	0.9634...	0.619...	0.5402410...	0.7020942...	0.89061...	0.702...	-0.999717...	0.70209426...
Laplace	0.1864...	0.963482...	1.0	0.504...	0.6618834...	0.7325063...	0.75461...	0.732...	-0.967326...	0.73250633...
Interest	-0.345...	0.619815...	0.5049...	1.0	0.5871869...	0.8028555...	0.77441...	0.802...	-0.616399...	0.80285556...
Interestin...	0.5435...	0.540241...	0.6618...	0.587...	1.0	0.8684139...	0.31780...	0.868...	-0.549855...	0.86841397...
Interest fu...	0.1198...	0.702094...	0.7325...	0.802...	0.8684139...	1.0	0.65844...	1.0	-0.708559...	1.00000000...
Leverage	-0.485...	0.890618...	0.7546...	0.774...	0.3178062...	0.6584437...	1.0	0.658...	-0.885351...	0.65844372...
Novelty	0.1198...	0.702094...	0.7325...	0.802...	0.8684139...	1.0	0.65844...	1.0	-0.708559...	1.00000000...
Informativity	0.0501...	-0.99971...	-0.967...	-0.616...	-0.5498550...	-0.708559...	-0.8853...	-0.708...	1.0	-0.7085590...
Weighted ...	0.1198...	0.702094...	0.7325...	0.802...	0.8684139...	1.0000000...	0.65844...	1.000...	-0.708559...	1.0

Fig. 9. Correlation analysis window.

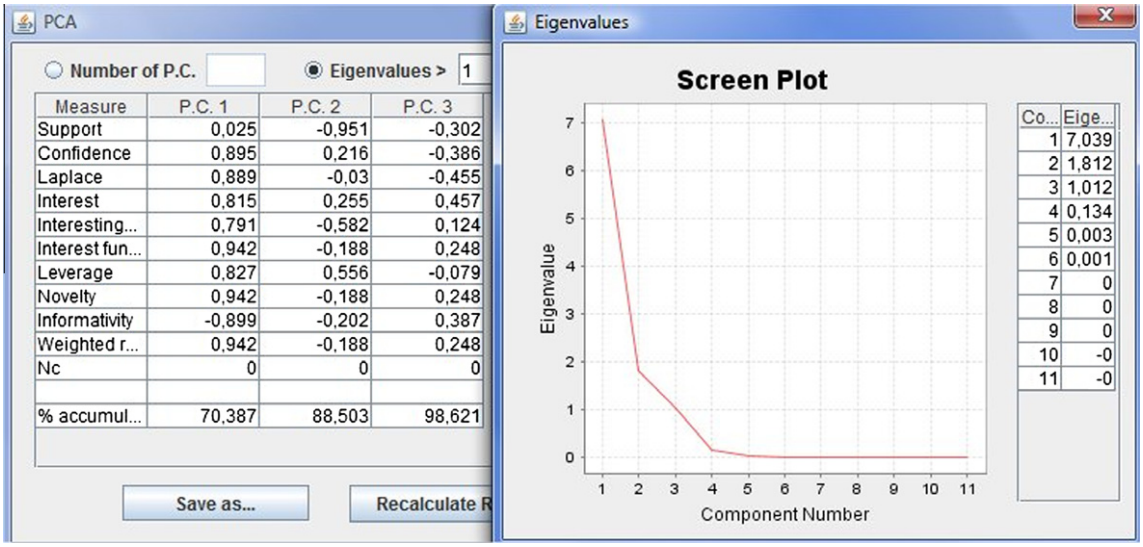


Fig. 10. Principal components analysis windows.

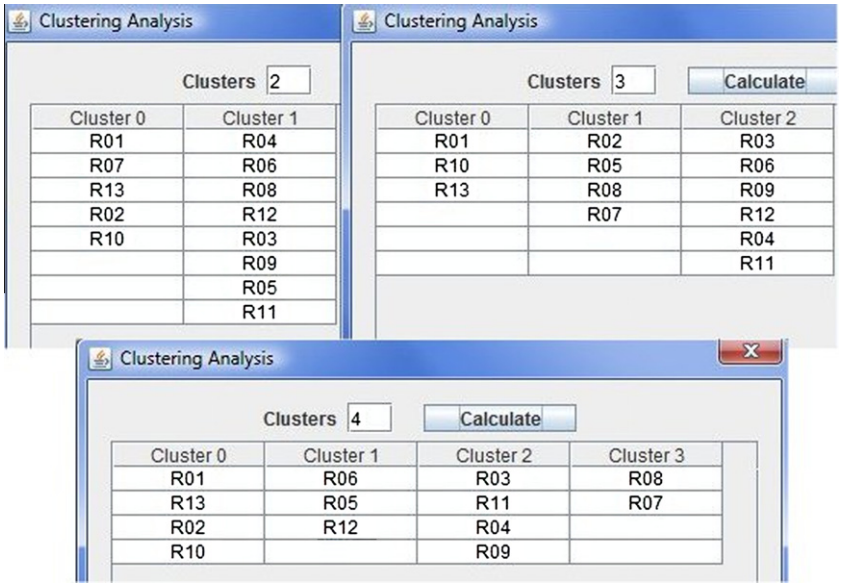


Fig. 11. Clustering analysis windows.

the user can reduce the number of measures used to only those that are not actually correlated, such as the Support, Interest, Interestingness, and Leverage ones, and choose one from each group of correlated measures.

5.4.2. Principal component analysis

PCA [10] is exploratory data analysis for reducing the number of variables. This technique can help the user to group and reduce the number of measures used. RM-Tool implements PCA, so the user

can select either the number of the principal component or the maximum eigenvalue, and optionally show the screen plot, as depicted in Fig. 10.

Fig. 10 shows the PCA obtained when selecting eigenvalues greater than 1 in order to see how many components or groups the 10 previously defined measures can be grouped into. The amount of variance and eigenvalue obtained for each principal component are also shown. Using the screen plot and the eigenvalues, the user can select a number of principal components, normally those with eigenvalues greater than 1 or when the plot curve starts to fall. For the case study used in this section, the results obtained indicate that the three principal components can represent the 10 measures, since they store 98% of the variance of the data. The communality values of each principal component for each measure are also shown along with the screen plot (see Fig. 10 at right). Then, using the communality values for each measure, the user can group the principal components by assigning or classifying each measure in the component where the highest absolute values are shown. For example, these three principal components can be used as new evaluation measures which include almost all the information provided by the original measures. In this way, the number of measures used could be reduced from 10 measures to only 3 meta-measures.

5.4.3. Clustering

Clustering is the process that groups objects into classes of similar objects [18]. It consists in unsupervised classifying or partitioning of patterns (observations, data items, or feature vectors) into groups or subsets (clusters). This technique groups records together based on their locality and connectivity within an n-dimensional space. The principle of clustering is to maximize the similarity within an object group and minimize the similarity between object groups. RM-Tool implements K-means [23], which is one of the simplest and most popular clustering algorithms. The K-means algorithm groups objects into k partitions based on attributes. The final objective is to be able to group the rules into different clusters/groups depending on the values of their evaluation measures (see Fig. 11).

Fig. 11 shows the clusters obtained when selecting $k = 2$, 3 and 4 clusters using the 13 rules discovered and the 10 previously defined evaluation measures. Although the number of clusters has changed/increased, it is clear that most of the rules remain grouped in the same clusters. For example, rule 1, 10 and 13 are always in cluster 0; rules 3, 11, 4 and 9 in clusters 1 and 2; rules 8 and 7 in clusters 1 and 3. It shows that these rules (the rules grouped in the same cluster) are very similar from the point of view of their evaluation measures. So, if one of these rules is considered useful or interesting for our needs, then the other rules of the same cluster could also be useful or interesting since they have similar values in their evaluation measures.

6. Conclusions and future work

This paper describes RM-Tool, a complete and fully integrated environment for discovering and evaluating association rules. This framework allows the user not only to discover different types of association rules (frequent, infrequent and class) using a wide range of algorithms, but also to apply filters a priori and a posteriori in order to visualize rules in both tabular and graph modes, to select the most useful rules for each user, to evaluate rules using both predefined and new objective evaluation measures and to apply to them such techniques as clustering, correlation analysis and principal component analysis. This paper has explored these features and explained, in a tutorial and practical way, how a DM user can take advantage of this tool to fulfill his needs.

In the near future, we plan to add more subjective and semantically based measures, which are not yet available in RM-Tool. To do so, we would like to add subjective restrictions indicated by the user to take information about specific semantics into account and to develop brand-new semantically based measures that can use the domain information in the OWL files. In this way, a final user could create new measures specifically geared toward each application domain or dataset. Finally, due to the increasing number of advanced ARM algorithms that are proposed every year, it is necessary to constantly up-date our list of available algorithms to include important novel ones.

Acknowledgments

The authors gratefully acknowledge the support provided by the TIN2008-06681-C06-03 project of the Spanish Inter-Ministerial Commission of Science and Technology (CICYT), the P08-TIC-3720 project of the Andalusian Science and Technology Department, and FEDER funds.

References

[1] Agrawal R, Imielinski T, Swami AN. Mining association rules between sets of items in large databases. In: Proceedings of SIGMOD, Washington DC; 1993. p. 207–16.
[2] Alcalá J, Sánchez L, García S, del Jesus MJ, Ventura S, Garrell JM, et al. KEEL: A software tool to assess evolutionary algorithms to data mining problems. Soft Comput 2009;13(3):307–18.
[3] Bayardo RJ. Agrawal, R. Mining the most interesting rules. In: Proceedings of conference ACM on knowledge discovery and data mining SIGKDD, San Diego, USA; 1999. p. 145–54.
[4] Brin S, Motwani JD, Ulhman JD, Tsur S. Dynamic itemset counting and implications rules for market baseket data. In: Proceedings of the ACM SIGMOD international conference on management of data. Tucson, Arizona; 1997. p. 255–64.
[5] Coenen F, Leng P, Ahmed S. Data structures for association rule mining: T-trees and P-trees. IEEE Trans Knowl Data Eng 2004;16:774–8.
[6] Coenen FP, Leng P, Goulbourne G. Tree structures for mining association rules. J Data Min Knowl Disc 2004;8(1):25–51.
[7] Coenen F, Leng P, Zhang, L. Threshold tuning for improved classification association rule mining. In: Proceeding PAKDD, Hanoi, Vietnam; 2005. p. 216–25.
[8] García E. Romero C, Ventura S, Calders T. Drawbacks and solutions of applying association rule mining in learning management systems. In: International workshop on applying data mining in e-learning, Crete, Greece; 2007. p. 13–22.
[9] García E. Romero C, Ventura S, De Castro C, Calders T. Association rule mining in learning management systems. In: Handbook of Educational Data Mining. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series. Taylor & Francis Group; 2010. p. 93–106.
[10] Fukugana K. Introduction to statistical pattern recognition. Elsevier; 1990.
[11] Freire M, Rodríguez P. A graph-based interface to complex hypermedia structure visualization. In: Proceedings of the working conference on advanced visual interfaces, Gallipoli, Italy; 2004, pp. 163–6.
[12] García E, Romero C, Ventura S, de Castro C. A collaborative educational association rule mining tool. Internet High Educ 2011;14(2):77–88 [Special Issue on Web Mining and Higher Education].
[13] Ceglar A, Roddick JF. Association mining. ACM Comput Surv 2006;38(2):1–42.
[14] Geng L, Hamilton HJ. Interestingness measures for data mining: a survey. ACM Comput Surv 2006;38(3):1–32.
[15] Goethals B, Bussche J. A priori versus a posteriori filtering of association rules. ACM SIGMOD workshop on research issues in data mining and knowledge discovery; 1999. p. 1–5.
[16] Han J, Pei J, Yin Y. Mining frequent patterns without candidate generation. In: Proceedings of the ACM-SIGMOD international conference on management of data, Dallas, Texas, USA; 2000. p. 1–12.
[17] Huysmans J, Baesens B, Vanthienen J. Using rule extraction to improve the comprehensibility of predictive models. Department of decision sciences and information management (KBI) K.U. Leuven KBIWorking Paper No. 0612; 2006. p. 1–56.
[18] Jain AK, Murty MN, Flynn PJ. Data clustering: a review. ACM Comput Surv 1999;31(3):264–323.
[19] Koh Y, Rountree N. Finding sporadic rules using Apriori-Inverse. In: Pacific–Asia conference on knowledge discovery and data mining, Berlin; 2005. p. 97–106.
[20] Ko Y, Rountree N. Rare association rule mining and knowledge discovery. Inform Sci Ref 2009.

[21] Liu B, Hsu W, Ma Y. Integrating classification and association rule mining. International conference on knowledge discovery and data mining; 1998. p. 80–6.
[22] Liu Y, Salvendy G. Visualization to facilitate association rules modeling: a review. Ergonomia IJE&HF 2005;1:11–23.
[23] MacQueen J. Some methods for classification and analysis of multivariate observations. In Proceedings of the fifth berkeley symposium on mathematical statistics and probability. California, USA; 1967. p. 281–97.
[24] Rodgers JL, Nicewander WA. Thirteen ways to look at the correlation coefficient.. Am Statist 1988;42:59–66.
[25] Romero C, Romero JR, Luna JM, Ventura S. Mining rare association rules from e-learning data. In Proceedings of the international conference of educational data mining. Pittsburgh; 2010. p. 171–80.
[26] Romero C, Ventura S, de Bra P. Knowledge discovery with genetic programming for providing feedback to courseware author. User Model User-Adapted Interact 2004;14(5):425–65.
[27] Silberschatz A, Tuzhilin A. What makes patterns interesting in knowledge discovery systems. IEEE Trans Knowl Data Eng 1996;8(6):970–4.
[28] Szathmary L, Napoli A. Valtchev, P. Towards rare itemset mining. In: International conference on tools with artificial intelligence, Washington, DC; 2007. p. 305–12.
[29] Tan P, Kumar V. Interesting measures for association patterns. In: Proceedings of the KDD workshop on postprocessing in machine learning and data mining, Boston, USA; 2000. p. 1–9.
[30] Witten IH, Frank E. Data mining. Practical machine learning tools and techniques with java implementations, Morgan Kaufmann; 1999.
[31] Zaki MJ, Phoophakdee B. MIRAGE: a framework for mining, exploring and visualizing minimal association rules. Technical Report 03-4, Computer Science Department, Rensselaer Polytechnic Institute, Troy, NY; 2003.
[32] Zhang H, Zhao Y, Cao L, Zhang C, Bohoscheid, H. Rare class association rule mining with multiple imbalanced attributes. Rare association rule mining and knowledge discovery, Information Science Reference; 2009.