Data Min Knowl Disc (2007) 14:409–431
DOI 10.1007/s10618-006-0062-6

Using metarules to organize and group discovered association rules

Abdelaziz Berrado · George C. Runger

Received: 26 April 2005 / Accepted: 20 November 2006 / Published online: 10 February 2007

Abstract The high dimensionality of massive data results in the discovery of a large number of association rules. The huge number of rules makes it difficult to interpret and react to all of the rules, especially because many rules are redundant and contained in other rules. We discuss how the sparseness of the data affects the redundancy and containment between the rules and provide a new methodology for organizing and grouping the association rules with the same consequent. It consists of finding *metarules*, rules that express the associations between the discovered rules themselves. The information provided by the metarules is used to reorganize and group related rules. It is based only on data-determined relationships between the rules. We demonstrate the suggested approach on actual manufacturing data and show its effectiveness on several benchmark data sets.

Keywords Item sets · Data sparseness · Clustering rules · Classification · Rules pruning

1 Introduction

Mining association rules from massive data often results in a massive set of rules. The large number of rules makes it overwhelming to extract the desired

Responsible editor: M. J. Zaki.

A. Berrado (✉)· G. C. Runger
Department of Industrial Engineering, Arizona State University, Tempe, AZ, USA
e-mail: berrado@asu.edu

G. C. Runger
e-mail: runger@asu.edu

information by simply analyzing the rules individually. The rules are often highly redundant. Organizing the rules is a key to summarizing them and making them easily understandable and interpretable by the data analyst. In this paper, we suggest a new approach to reorganizing and grouping the redundant rules with the same consequent by unveiling their mutual relationship and their containment in other more general rules. We find new rules that we call *metarules* which reveal relationships between the discovered association rules. We use the information provided by the metarules to reorganize and group the related rules. The metarules are also useful for pruning the more specific, possibly overfitting, rules. We apply the same association algorithm used to generate the rules to derive the metarules. This approach postpones ad hoc, analyst preferences for rules until an organized summary of the rules is generated. The organized collection presents only data-determined relationships between the rules. Then preferences can later be applied to the organized collection.

Our focus is on summarizing a subset of association rules with the same consequent. Our basic approach could likely be extended to general association application, but we do not explore that extension here. The rest of this paper is organized as follows. A brief introduction to association rules is given in Section 2. We also cite and summarize published work related to the topic at hand. Containment and overlap of rules is discussed in Section 3. In Section 4, we explain how metarules are generated and used to organize the discovered rules. An example in Section 5, illustrates the application of the suggested methodology to actual manufacturing data and shows its advantage over existing approaches to clustering rules. In Section 6, experimental results concerned with the effectiveness of the suggested approach with several well known data sets are given. Section 7 is a conclusion.

2 Background for association rules mining and related work

Association rules mining emerged as a technique for finding interesting rules from transactional databases (Agrawal et al. 1993). More specifically, it was initially used to reveal associations in commercial data from a database of transactions each representing the set of items purchased by a customer. The association analysis identifies items purchased together.

An association rule is an expression of the form: $A \rightarrow C$, where A and C are subsets of the set of items. Here A is referred to as the set of antecedents and C as the set of consequents. The subsets A and C are disjoint. The importance of a rule is evaluated by its support and confidence. The support of a rule is the fraction of all transactions where the set of antecedents A and the set of consequents C apply simultaneously. The support of a rule is a measure of its importance in terms of the number of transactions where the rule applies. The confidence of a rule is the fraction of the set of transactions containing the set of antecedents A which contain the set of consequents C. Furthermore, the confidence of a rule quantifies the strength of the association between the set of antecedents and the set of the consequents. Minimum support and confidence thresholds are usually pre-specified before mining for association rules. Using a low-support threshold uncovers all the underlying regularities in the data but also results in a high number of association rules, most of which are redundant and/or contained in other rules.

Several efforts have been deployed to tackle the problem of summarizing and pruning the huge number of rules. Klemettinen et al. (1994) suggested a method using templates which allow the user to retrieve only the rules that are of interest. Other approaches, (Liu and Hsu 1996; Silberschatz and Tuzhilin 1996; Liu et al. 1997; Padmanabhan and Tuzhilin 1998) find those unexpected rules by comparing the discovered rules to the pre-defined user's knowledge about the domain. These methods allow the user to view only rules that are of interest but they don't prune or summarize the rules. Ng et al. (1998) and Srikant et al. (1997) require the user to specify constraints or restrictions regarding the items that are associated simultaneously in the mined association rules. Using these item constraints certainly reduces the number of the resulting rules but the mined rules still need to be pruned and summarized.

Another approach to pruning the rules is called *Minimum Improvement* (Bayardo et al. 2000). A rule is pruned if the difference between its confidence and the confidence of any of its proper subrules (a proper subrule is a simplification of the rule formed by removing one or more conditions from its antecedents) is less than the pre-specified *Minimum Improvement*. This method would not be effective when the data is sparse because many overlapping rules won't be sensitive to a low-*Minimum Improvement* threshold and thus won't be pruned. Furthermore, this method does not perform a summarization of the rules that remain after pruning.

Statistical tests have been utilized for pruning potentially uninteresting rules generated due to sampling. Bay and Pazzani (2001) provided a search algorithm for mining contrast sets with pruning rules. Contrast sets refer to conjunctions of attributes and values that differ meaningfully in their distribution across groups or classes of interest. They applied a significance test to remove the insignificant contrast sets. Huang and Webb (2005a) also developed an insignificance filter for automatically discarding insignificant rules during rule discovery with the OPUS search algorithm from data with undiscretized quantitative attributes. Huang and Webb (2005b) suggested a new derivative rule filter using a *t*-test for pruning a class of insignificant rules called *Derivative Rules* that are not successfully removed using existing rule pruning techniques.

A chi-squared test was also used as the basis for pruning Liu et al. (1999). The dependence between the antecedents and the consequent of a rule is evaluated by a χ^2 test. A rule is pruned if its χ^2 statistic is lower than the pre-specified threshold value which corresponds to a desired significance level. In sparse domains this technique will fail to prune many rules. Furthermore Liu et al. (1999) suggested a method for summarizing the remaining rules into a special subset of associations called *the direction setting rules* which give a global picture of the underlying relationships in the domain. Clearly the completeness of the set of *direction setting rules* depends on the results of the previous pruning step.

Toivonen et al. (1995) introduced a technique for pruning the discovered rules by forming *rule covers*. A cover is a subset of rules that covers the entire database. Although this method reduces the number of rules significantly, it does not conserve all the information embedded in the discovered rules because a greedy algorithm is used to find a good cover.

Chawla et al. (2004) proposed an adaptive local pruning method for association rules using directed hypergraphs. The method uses *Association Rules Networks* as a graphical method to prune rules by associating with hypercycle and reverse hyperedges. The pruning is local because it takes place in the context of a goal node. That is, a rule that is considered redundant for a particular goal node may become important for another goal node. Unlike our approach presented herein, this method does not provide a global organization of the discovered association rules.

Several methods use distance-based clustering to group association rules. Toivonen et al. (1995) defined the distance between two association rules as the number of rows where the rules differ. A new normalized distance metric was presented by Gupta et al. (1999) to cluster the rules. Lent et al. (1997), clustered association rules with a geometric-based algorithm. This application was limited to rules with two attributes in the set of antecedents. Defining a distance metric between two rules is sensitive to the asymmetric relationship between the rules. That is, if a rule is viewed as the set of rows that satisfy the rule, there is often considerable containment and overlap between the sets of rows that satisfy each rule.

Association rule mining is also used in classification. The approach of integrating classification and association rules mining (known as associative classification) is not new to the machine learning community. CBA (Liu et al. 1998) generates association rules with consequent restricted to a class attribute. Such class association rules (CARs) were also found using CBA-RG (Liu et al. 1998) which is adapted from Apriori (Agrawal et al. 1993). The discovered CARs are then used for classification. The CARs are sorted based on their confidence then support and finally based on the order in which they were discovered. Then a subset C of high precedence rules from CARs are chosen to cover the data set. This subset of rules is augmented with a default class (majority class of uncovered data) and then used for classification. A rule is chosen to classify a data instance based on high precedence. CMAR (Li et al. 2001) or classification based on multiple association rules, is more efficient than CBA in terms of finding the association rules, it adopts a variant of an efficient frequent pattern mining, FP-growth (Han et al. 2000), to find the rules and stores them efficiently in a CR-tree which is a prefix tree structure. More specific and lower confidence rules are then pruned. The rules are thereafter pruned based on database coverage. CMAR uses a coverage threshold to select database coverage. After pruning the rules, CMAR determines the class label of a data instance based on a weighted χ^2 analysis of multiple rules and not by the rule with highest precedence like in CBA. Both CBA and CMAR use the pessimistic-error-rate pruning in C4.5 (Quinlan et al. 1992) to prune the discovered association rules. In this paper, we suggest using metarules as an alternative to pruning and organizing the discovered class association rules.

Association rules were also considered in the case were the data set contains quantitative attributes. Srikant and Agrawal (1996) suggested the first algorithm of the quantitative case, it uses discretization of the quantitative data. Fukuda et al. (1999) also provided an efficient algorithm using computational geometry and sampling methods for efficiently mining quantitative association rules. Their solution was however, limited to rules with a categorical consequent. Aumann and Lindell (2003) introduced a new definition of quantitative association rules based on statistical inference theory. Their work focuses on situations with rules of one quantitative attribute or categorical attributes in the antecedent set and one quantitative consequent.

Mining association rules with multiple quantitative antecedents without prior discretization still remains an active research problem. Significant efforts have been made to develop discretization methods. Dougherty et al. (1995) supplied a review of several existing methods and classified them according to three major axes: global versus local, supervised versus unsupervised, and static versus dynamic. Yang and Webb (2002) conducted a thorough comparison of several of the existing methods when employed for naive-Bayes classifiers. Liu et al. (2002) provided a description of existing discretization methods and suggested some guidelines on how to choose a discretization method under various settings. In our example, we use a naive discretizer on the continuous attributes. The focus of our work is to organize generated rules regardless of the methodology used within rule discovery.

3 Containment and overlap of rules

One cause of the large number of rules is the redundancy among the rules due to the sparseness of the data in high-dimensional spaces. The sparseness of the data is due essentially to the curse of dimensionality. Scott and Thompson (1983), showed that the data gets sparser as the dimensionality gets higher, that local neighborhoods of points in high dimensions are mostly empty, and that even in the case of uniform distributions, data is concentrated at the borders of the volume of interest. We illustrate through the following example how the sparseness of the data results in a redundancy among the rules.

Let A, B, and C denote three categorical variables of interest: A with two categories, A_1 and A_2, B with three categories, B_1, B_2, and B_3 and C with two categories C_1 or C_2. Furthermore, assume that the distribution of the data is summarized in Tables 1 and 2.

Table 1 Distribution of the data when C is at the level C_1

		B		
		B_1	B_2	B_3
A	A_1	1	100	0
	A_2	0	100	1

Table 2 Distribution of the data when C is at the level C_2

	B			
		B_1	B_2	B_3
A	A_1	0	0	15
	A_2	0	0	10

Table 3 Discovered association rules with C_1 as a consequent

Rule index	Antecedents	Consequent	Support Frequency
r_1	A_1	C_1	101
r_2	A_1 and B_2	C_1	100
r_3	A_2	C_1	101
r_4	A_2 and B_2	C_1	100
r_5	B_2	C_1	200

The numerical values in the cells in Tables 1 and 2, indicate the number of observations in each cell. Note that some cells have few (or zero) observations.

Suppose that we are interested in the association rules that have C_1 as a consequent. It follows that we find five rules with C_1 as a consequent and support frequency of at least 100. The rules are summarized in Table 3 where the first column, labeled rule index, represents an arbitrary index that refers to each rule. In the case under study, all the rules have the same consequent C_1. The effect of sparseness is the generation of several rules.

The distribution of the data suggests that the categories of the variable A do not affect the density of the data in the different cells of Table 1. Nevertheless, the category A_1 appears as an antecedent of the rules r_1 and r_2, and the category A_2 appears as an antecedent of the rules r_3 and r_4. Notice that the rules r_1 and r_2 are derived from nearly the same rows (or examples). It is clear then that they convey the same information. The same thing is also true for rules r_3 and r_4. In contrast, the rule r_5 resulted from the high density of category B_2 and the sparseness of the data in the other categories of B.

Hence, the sparseness of the data in some regions of the space results in a high number of redundant association rules, most of which are not important. This makes it difficult to find important or good rules among the resulting rules especially in a high-dimensional space. Detecting those regions where the rules are redundant and/or contained in other rules is then a key to grouping and pruning the discovered rules. This type of relationship between rules is difficult to capture with a clustering algorithm based on a distance metric. Examples are presented in Sect. 5.5.

4 Finding metarules to organize and prune the discovered rules

4.1 Finding metarules

We propose a novel technique for grouping and then pruning the discovered rules which share the same consequent. It consists of finding one-way associations between those discovered rules. One-way association rules refer to rules with one antecedent and one consequent. We call these associations between the rules *metarules*. The approach simply and easily summarizes the asymmetric relationships between the rules. We apply the same association algorithm in a new role.

Let $I = \{i_1, i_2, \ldots, i_p\}$ be a set of p items, and $D = \{d_1, d_2, \ldots, d_n\}$ be the set of all n data rows, where each data row contains a subset of items from I. Let $R = \{r_1, r_2, \ldots, r_m\}$ represent the set of the m discovered association rules with the same consequent of interest obtained from D.

We say that a rule r_i from R is supported by the data row d_j or that data row d_j supports rule r_i if all the antecedents of the rule r_i are items of d_j and we refer to this relationship by the following expression: $r_i \subseteq d_j$. Define a new set of transactions $Q = \{q_1, q_2, \ldots, q_l\}$ where $l \leq n$ such that every element q_j of Q is a subset of rules from R such that:

$$q_j = \{r_i \in R \mid r_i \subseteq d_j\}.$$

In other words, each rule is considered an item. Also, each data row from D is mapped to the subset of rules from R which it supports and this subset of rules (when nonempty) corresponds to a transaction in Q. Note that if every data row supports at least one rule then $l = n$.

If we find one-way association rules from the set Q, we will determine all the one-way associations between the rules from R. The metarules take the form $r_i \to r_j$ where r_i and r_j are rules from R. The resulting metarules provide a summary of the containment and overlap between the rules. The confidence threshold for mining metarules is defined similarly to an ordinary association rule but applied to the set Q. The support threshold for metarules mining can be set to 0% in order to uncover all the relationship between all the rules. Because metarules only calculate rules with one antecedent and consequent the calculations are simpler than a full association analysis and the support is easily set to 0%. Let $MR = \{mr_1, mr_2, \ldots, mr_k\}$ refer to the set of the k discovered metarules, obtained from Q. If we analyze the metarules from MR, we can understand the relationship between the association rules from R.

Also, a graphical presentation of the metarules is defined here. Each rule from R is represented by a node and each metarule connects two nodes with a directed arc, such that the originating node is the antecedent of the metarule and the destination node is its consequent.

We illustrate the suggested approach to mining metarules through the same example that we used in Sect. 3. Recall that we found five rules so that $R = \{r_1, r_2, \ldots, r_5\}$. Also, $n = 227$. Note that $l = 227$ and it corresponds to the number of data rows which support the antecedents of at least one of five rules from R. Mining for association rules from the data cases for the corresponding Q results in the eight metarules summarized in Table 4. The minimum confidence and support thresholds used for this example are, respectively, 85 and 0%.

In general the support and confidence thresholds would be set for metarules in identically the same manner as the usual rules. Because many rules are redundant, confidences as high as 90% or more can be used. Our following examples use 100%. A higher confidence threshold results in a smaller set of metarules. Large numbers of metarules indicate many relationships between the rules. The support threshold for metarules mining that we use is specified to 0% in order to uncover all the relationships between all the rules.

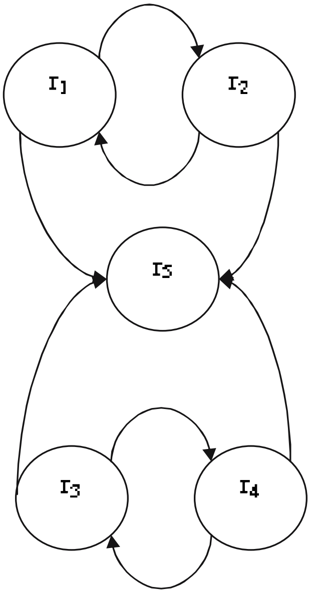
Note that our approach could likely be extended by using other metrics instead of or in addition to support and confidence to assess the interestingness of the discovered rules or metarules (Tan et al. 2002), but we do not explore that extension here.

The metarules from Table 4 reveal the dependency between the rules of interest: $r_1, \ldots, r_5$. The graph in Fig. 1 illustrates the relationship between the five rules. The graph is composed of five nodes, one for each rule and eight directed arcs, one for each metarule. For example, the metarule $r_1 \rightarrow r_2$ creates

Table 4 Discovered metarules

Metarule index	Antecedents	Consequent	Support (%)	Confidence (%)
mr_1	r_2	r_1	44	100
mr_2	r_1	r_2	44	86
mr_3	r_2	r_5	44	100
mr_4	r_4	r_3	44	100
mr_5	r_3	r_4	44	90
mr_6	r_4	r_5	44	100
mr_7	r_1	r_5	44	86
mr_8	r_3	r_5	44	90

Fig. 1 Relationship between the rules



an arc originating from the node r_1 and ending at the node r_2. Figure 1 shows the complete relationship between the five rules.

Figure 1 shows that the rules r_1 and r_2 show a mutual relationship with each other. We conclude that the rules r_1 and r_2 are approximately equivalent. That is, the two rules are supported by the same data rows. The same thing is true for the rules r_3 and r_4. We also notice that there are arcs originating from the nodes r_1–r_4 toward the node r_5 and none in the other direction, we say that the rule r_5 is less specific than the other rules. That is, the data rows that support each of the rules r_1–r_4 are contained in the rows that support the rule r_5. In Sect. 4.3, formal definitions for specificity of a rule and equivalence between rules are provided.

4.2 Finding independent subgroups of rules

Using the metarules to build a graphical representation can divide the rules from R into disjoint subgroups of rules, if they exist. That is, there might be no arcs that link the rules spanning a subgroup to the rules spanning the other subgroups. We illustrate this by considering a more complex example in Sect. 5. Each cluster of rules explains the data in a local region of the high-dimensional space.

Using the metarules for the organization of the discovered rules into clusters or subgroups of rules differs from the distance-based clustering algorithms used for the same purpose (Toivonen et al. 1995; Gupta et al. 1999). Our method does not need a distance metric to find the subgroups of rules. The asymmetric relationships generated from the containment and overlap between the rules makes it challenging to define an appropriate distance metric to be used for clustering the rules.

4.3 Reorganizing the equivalent and the more specific rules

Let r_i be a rule from R and let $OUT(r_i)$ refer to the subset of rules r_j from R defined as follows:

$$OUT(r_i) = \{r_j \in R \mid r_i \to r_j \in MR\}.$$

Also, let $IN(r_i)$ refer to the subset of rules r_j from R such that $r_j \to r_i$ is a metarule from MR:

$$IN(r_i) = \{r_j \in R \mid r_j \to r_i \in MR\}.$$

Definition 1 Let r_i and r_j be two rules from R, we say that r_i is more specific than r_j or that r_j is more general than r_i, if the following condition is met:

1. $r_j \in OUT(r_i)$ and $r_j \notin IN(r_i)$.

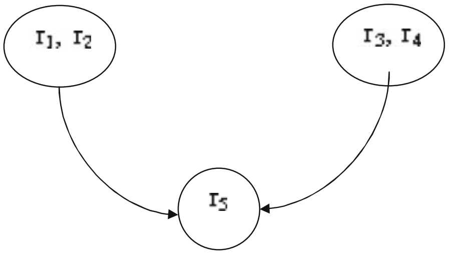


Fig. 2 Relationship between the rules after combining the equivalent rules

Definition 2 Let r_i and r_j be two rules from R, and let c_{ij} and c_{ji} denote respectively the confidences of the metarules $r_i \rightarrow r_j, r_j \rightarrow r_i$.
we say that r_i and r_j are equivalent if the following three conditions are satisfied:

1. $r_i \in OUT(r_j)$ and $r_i \in IN(r_j)$,
2. $OUT(r_i) \setminus \{r_j\} = OUT(r_j) \setminus \{r_i\}$,
3. $IN(r_i) \setminus \{r_j\} = IN(r_j) \setminus \{r_i\}$,

Where $OUT(r_i) \setminus \{r_j\}$ refers to the subset of $OUT(r_i)$ excluding the rule r_j, that is $OUT(r_i) \setminus \{r_j\} \cup \{r_j\} = OUT(r_i)$. Note that if r_i and r_j are equivalent and $c_{ij} = c_{ji} = 100\%$ then the two rules are supported by exactly the same data rows. That is, they are different representations of the same relationship.

If we group each pair of equivalent rules and represent them by the same node, the graph from Fig. 1 is reduced to the graph in Fig. 2. This more clearly explains the relationships.

The computational overhead in the suggested approach could be broken down into the following three components:

- The first one is concerned with creating the new set of transactions Q, which takes *mn* computations to create.
- The second is concerned with metarules mining, which is simply the same as the computational complexity of a priori because it is used to find metarules from Q. We note that only one-way association rules (rules with a single antecedent and single consequent) are calculated so that this step is much simpler than a full association rule analysis.
- The third is concerned with the computational complexity of simplifying metarules, which is quadratic to the number of metarules.

4.4 Pruning the discovered rules using metarules

Although the objective of the metarules approach is to group and organize rules, the approach can also be directly used to prune the rules. First consider equivalent rules. Suppose that rules r_i and r_j are found equivalent using metarules. Now, if the antecedent itemset of r_j is contained within the antecedent set of r_i, then r_j can be pruned because it is more complex than r_i. For the example at hand, this means that r_2 with antecedent $\{A_1, B_2\}$ can be pruned because it is contained in r_1 with antecedent $\{A_1\}$. Similarly, r_4 can be pruned since it is contained in r_3. This leads to the following definition.

Definition 3 Let r_i and r_j be two rules from R, and let c_{ij} and c_{ji} denote the confidences of the metarules $r_i \rightarrow r_j$ and $r_j \rightarrow r_j$, respectively. We say that r_j is more *complex* than r_i if the following two conditions are satisfied:

1. r_i and r_j are equivalent according to Definition 2, and,
2. The antecedents of r_i is a proper subset of the antecedents of r_j.

Note that Definition 3 implies that $c_{ji} = 100\%$, because the support of r_j is completely included in the support of r_i. The examples that follow show that a large number of rules are related through this definition of equivalence and complexity. Consequently, pruning based on this simple definition of equivalence and complexity is useful. This might be expected from sparse data in high dimensions.

More generally, metarules can indicate clearly which rules are more specific, and they can be directly used for pruning. However, for rules that do not satisfy Definition 3, the decision to prune has some consequences. Even equivalent rules may result in ambiguous pruning decisions. For example, when r_1 has antecedent $\{A_1, B_2\}$ and r_2 has antecedent $\{C_1, D_2\}$ our solution is to merge the rules to a common node, but not make a pruning decision. More importantly, the fact that a rule is more specific than another rule does not imply that it should be pruned. It may or may not be overfitting the data, because it has a smaller support than the more general one. Subject knowledge can be used instead to decide which of the more specific rules should be pruned. We recommend that this task be delayed until after the pruning of the equivalent rules has taken place. For the example at hand, if subject knowledge indicates that the two nodes with more specific rules, $\{r_1$ and $r_2\}$ and $\{r_3$ and $r_4\}$, can be pruned, then we are left only with the rule r_5. Here r_5 is $B_2 \rightarrow C_1$ and it is a reasonable summary of the data in Table 1.

Although metarules directly apply to the issue of pruning specific rules in the presence of more general ones, this topic is not further explored in this paper. Instead, we propose to simplify the rules as much as possible before subject matter expertise is invoked. Consequently, the rule reductions that we present in the following examples are conservative. They are based only on our summary graphics and with nodes joined based on equivalence defined in Definition 2, without the pruning of more specific rules.

5 Application

Assume that we are interested in determining which process variables are responsible for a defective product from a manufacturing process. Also, assume that the variables being investigated are all categorical. If we gather data about the outcome of the process (good or defective output), given certain process settings (operating conditions), and if we consider each observation as a transaction, we can use association rules mining to associate defects with certain process settings. Process settings refer to the process variables being set at certain levels. Because we are trying to explain a defective output, the consequent of interest for the rules is “defective output.” Each element in the set of antecedents represents a process variable set at some level. Note that looking for a specific consequent such as a defective output, narrows down the pool of rules that need to be analyzed. This subset of rules can however still be massive and the rules highly redundant and thus they still need to be pruned and summarized.

We now illustrate the advantage of our approach by organizing the association rules discovered from actual data provided by a major manufacturer. The data has been coded and no actual variables or process names are used. The data was collected from a manufacturing process with 35 numerical variables and a binary response indicates whether the output is good, G, or rejected, R. The number of observations is approximately 10,000.

5.1 Discretization of the numerical variables

Because the data provided is composed of 35 numerical variables and a binary response, we had to discretize the numerical feature space before generating the association rules. As discussed in Sect. 2, several discretization methods are available. For initial results, we adopted a simple and naive method known as Equal Frequency Discretization to partition each of the 35 continuous attributes individually. For simplicity, each attribute was divided into four intervals. To discretize a variable x_i, we determined its minimum and maximum values, then sorted all values in ascending order and divided the range into four intervals so that each contains 25% of the data. As mentioned previously, although better quality rules might be obtained with an alternative discretizer, our objective is to summarize rules generated in any manner.

We identified each variable by a number between 1 and 35. For example, the number 1 refers to the variable x_1. Because each variable was partitioned into four intervals, we used the following indexing to refer to the intervals of each variable: the intervals corresponding to variable x_i are, respectively, i_1–i_4. Because the class label (column 36) was already categorical, it remained unchanged. As an example of the discretized data, consider the first data row: the measured values of the 35 variables were represented by the following record where the last element, R, on the record is the class label. 1_4, 2_1, 3_1, 4_1, 5_2, 6_1, 7_2, 8_4, 9_2, 10_2, 11_2, 12_2, 13_1, 14_3, 15_2, 16_4, 17_3, 18_1, 19_4, 20_1, 21_2, 22_4, 23_2, 24_3, 25_3, 26_1, 27_1, 28_1, 29_4, 30_4, 31_4, 32_3, 33_4, 34_2, 35_3, R.

5.2 Rule generation

A search was made for all possible rules with R as a consequent and the complete set of rules that resulted from this analysis are displayed in Table 5. We

Table 5 Generated association rules

Rule index	Antecedents	Consequent	Support (%)	Confidence (%)
r_1	1_2, 13_2, 11_1	R	1.2	81.8
r_2	3_2, 13_2, 11_1	R	1.6	82.1
r_3	3_2, 11_1, 14_4	R	1.1	80.6
r_4	20_2, 3_2, 11_1, 5_1	R	1.1	80.5
r_5	20_2, 3_2, 11_1, 10_1	R	1.1	80.6
r_6	20_2, 4_3, 11_1, 5_1	R	1.1	85.4
r_7	20_2, 4_3, 11_1, 10_1	R	1.1	85.0
r_8	17_2, 3_2, 11_1, 5_1	R	1.0	81.0
r_9	17_2, 3_2, 11_1, 10_1	R	1.0	81.0
r_{10}	12_3, 4_3, 2_2, 11_1	R	1.1	80.6
r_{11}	1_3, 4_3, 2_2, 11_1	R	1.2	80.0
r_{12}	1_2, 3_2, 11_1, 5_1	R	1.7	80.9
r_{13}	1_2, 3_2, 11_1, 10_1	R	1.7	81.0
r_{14}	1_2, 13_2, 11_1, 5_1	R	1.2	87.8
r_{15}	1_2, 13_2, 11_1, 10_1	R	1.2	87.9
r_{16}	1_2, 13_2, 11_1, 33_4	R	1.0	82.9
r_{17}	1_2, 11_1, 14_4, 5_1	R	1.4	86.8
r_{18}	1_2, 11_1, 14_4, 10_1	R	1.4	86.8
r_{19}	1_2, 11_1, 14_4, 8_3	R	1.1	83.7
r_{20}	3_2, 13_2, 11_1, 5_1	R	1.1	91.0
r_{21}	3_2, 13_2, 11_1, 10_1	R	1.2	91.1
r_{22}	3_2, 13_2, 11_1, 8_3	R	1.1	84.1
r_{23}	3_2, 13_2, 11_1, 33_4	R	1.3	81.1
r_{24}	4_3, 2_2, 11_1, 5_1	R	1.5	82.0
r_{25}	4_3, 2_2, 11_1, 10_1	R	1.5	82.0

used a confidence threshold of 80%. Note that the support of the discovered rules is in the order of 1%. It is low because we are interested in explaining the occurrences of a rare consequent *R*.

The discovered rules are redundant and some of them are contained in others. The 35 variables are not all independent from each other. Consider, for example, the rules r_1 and r_{14}. All the antecedents of r_1 are also antecedents of r_{14}. The set of rows that support r_{14} is included in the set of rows that supports r_1 so that r_{14} is more specific than r_1.

5.3 Finding metarules and rule organization

After finding the rules, we followed the method in Sect. 4 to generate the metarules and then used them to organize the discovered association rules. The metarules discovered from the rules from Table 5 are summarized in Table 6. Note that we used a confidence threshold of 80%. Using a higher confidence threshold, for example 90%, would result in fewer metarules. The support threshold used to generate the metarules is 10%. These metarules explain the dependency between the 25 rules from Table 5. For example mr_1 and mr_2 show a mutual dependency between the rules r_6 and r_7.

Table 6 Discovered metarules

Rule index	Antecedents	Consequent	Support (%)	Confidence (%)
mr_1	r_6	r_7	19.6	100
mr_2	r_7	r_6	19.6	99.2
mr_3	r_{25}	r_{24}	28.3	100
mr_4	r_{24}	r_{25}	28.3	100
mr_5	r_{19}	r_{17}	19.6	92.2
mr_6	r_{19}	r_{18}	19.7	93
mr_7	r_8	r_9	19.9	100
mr_8	r_9	r_8	19.9	100
mr_9	r_{22}	r_2	20.7	100
mr_{10}	r_4	r_5	21.9	100
mr_{11}	r_5	r_4	21.9	99.3
mr_{12}	r_{17}	r_{18}	24.8	100
mr_{13}	r_{18}	r_{17}	24.8	92.3
mr_{14}	r_{16}	r_{14}	17.8	82.4
mr_{15}	r_{14}	r_{16}	17.8	92.3
mr_{16}	r_{16}	r_{15}	17.9	93.2
mr_{17}	r_{15}	r_{16}	17.9	82.6
mr_{18}	r_{16}	r_1	19.2	100
mr_{19}	r_1	r_{16}	19.2	81.8
mr_{20}	r_{23}	r_2	26.2	100
mr_{21}	r_2	r_{23}	26.2	81.5
mr_{22}	r_{14}	r_{15}	21.5	100
mr_{23}	r_{15}	r_{14}	21.5	99.2
mr_{24}	r_{20}	r_{14}	16.3	81.1
mr_{25}	r_{21}	r_{14}	16.3	80.5
mr_{26}	r_{14}	r_1	21.5	100
mr_{27}	r_1	r_{14}	21.5	91.6
mr_{28}	r_{20}	r_{15}	16.3	81.1
mr_{29}	r_{21}	r_{15}	16.4	81.3
mr_{30}	r_{15}	r_1	21.7	100
mr_{31}	r_1	r_{15}	21.7	92.3
mr_{32}	r_{20}	r_{21}	20.1	100
mr_{33}	r_{21}	r_{20}	20.1	99.2
mr_{34}	r_{20}	r_1	16.3	81.1
mr_{35}	r_{20}	r_2	20.1	100
mr_{36}	r_{20}	r_{12}	16.3	81.1
mr_{37}	r_{20}	r_{13}	16.3	81.1
mr_{38}	r_{21}	r_1	16.4	81.3
mr_{39}	r_{21}	r_2	20.2	100
mr_{40}	r_{21}	r_{12}	16.3	80.5
mr_{41}	r_{21}	r_{13}	16.4	81.3
mr_{42}	r_{12}	r_{13}	33.6	100
mr_{43}	r_{13}	r_{12}	33.6	99.5

Refer to Fig. 3 for the graphical presentation of the metarules. Figure 3 shows that the discovered association rules are organized into several subgroups as follows:

- A cluster grouping the following eleven rules: $(r_1, r_{16}, r_{14}, r_{15}, r_{12}, r_{13}, r_{20}, r_{21}, r_2, r_{23}, r_{22})$.
- A cluster of three rules: (r_{17}, r_{18}, r_{19}).
- Four clusters that group two rules each: (r_6, r_7), (r_4, r_5), (r_8, r_9), and (r_{24}, r_{25}).

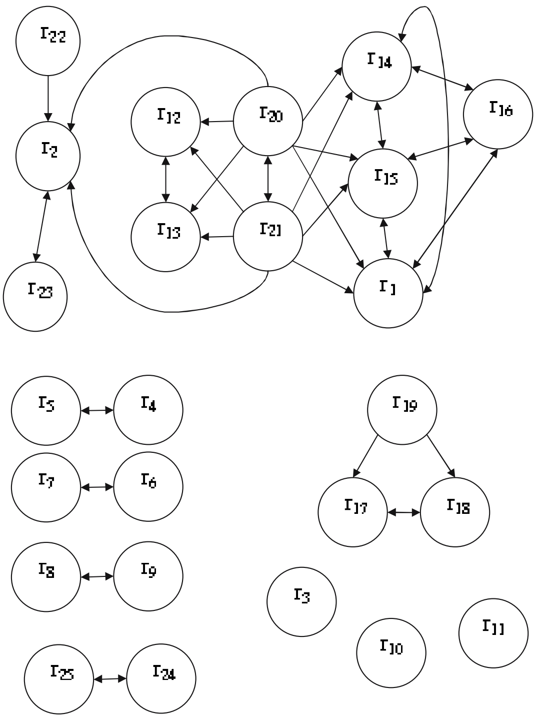


Fig. 3 Discovered metarules

- And finally three individual rules that did not participate in any metarule: r_3, r_{10}, and r_{11}.

With this new organization of the rules into independent subgroups, it is much easier to analyze and understand the rules.

5.4 Grouping the equivalent rules and pruning the more specific ones

Our ability to analyze the rules is enhanced further after more processing of the rules within each subgroup. Within each subgroup, we simplified the graphical presentation by grouping into the same node the equivalent rules. The reorganized rules are plotted in Fig. 4. The three individual rules which don't belong to any subgroup were not affected by this step. The subgroups with two nodes were reduced to one node. The subgroup with three nodes was reduced to a subgroup with two nodes. Finally, the subgroup with ten nodes was reduced to seven nodes.

Pruning the contained rules according to Definition 3 would result in pruning rules r_{14} and r_{15} since they are more complex than r_1. Notice that some of the subgroups could be simplified further. For instance, if we combine knowledge about the process with information about the more specific rules we might decide to prune rule r_{19} and possibly prune the node containing the rules r_{20} and r_{21} as well. We might also decide to group the rules (r_1, r_{14}, r_{15}) with rule

Fig. 4 Metarules after grouping equivalent rules

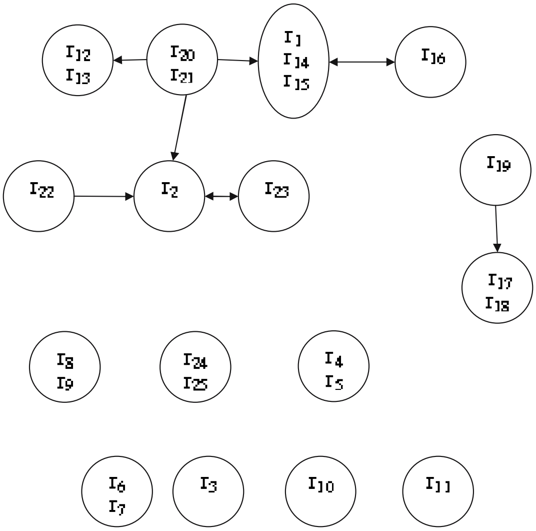
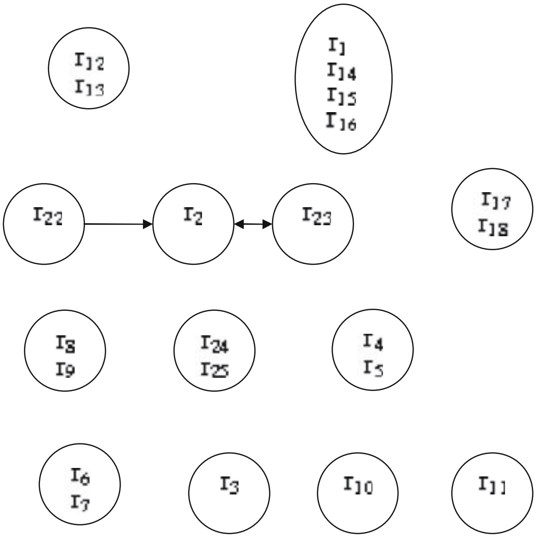


Fig. 5 Metarules after pruning the more specific rules



r_{16} into the same node. The final set of subgroups of rules after grouping and pruning several rules is described in Fig. 5. Each group of rules could then be further analyzed by looking at the variables involved in each of the rules within the same group.

This application justifies our original thesis that finding metarules enhances our understanding of the discovered association rules by braking them down into independent subgroups and pruning some overfitting rules within the subgroups. The reduced subgroups of rules are more actionable than the complete set of discovered rules.

5.5 Comparison with rules clustering approaches using distance metrics

In this section, we compare the groups formed using the metarules approach in Fig. 3 with the clusters formed using two different distance metrics to group association rules. Let d_1 and d_2 refer to the distance metric suggested in (Toivonen et al. 1995) and (Gupta et al. 1999), respectively. We used hierarchical clustering algorithms with both distance metrics and experimented with different linkages. Figures 6 and 7 illustrate, respectively, the dendrograms obtained with Ward linkage using the distance metrics d_1 and d_2, respectively.

First, note that cluster assignment differs when using different distance metrics. Take, for instance, rules 20 and 21. When using d_1 they were initially grouped with rules 1 and 14–16, and then with rules 2, 23, and 22. When using d_2, rules 20 and 21 where grouped with rules 2, 23, and 22 before grouping them with rules 1 and 14–16. Furthermore, Figs. 6 and 7 agree that rules 12 and 13 are first clustered with rules 4, 5, 8, and 9 while the metarules approach suggests that rules 20 and 21 have greater than 80% confidence to rules 12 and 13. We reduced the confidence level to understand the clustering results between rules 12, 13 and 4, 5 and we found the following metarules: $r_4 \rightarrow r_{12}$ with confidence 72.9% and $r_5 \rightarrow r_{12}$ with confidence 72.4% and $r_4 \rightarrow r_{13}$ with confidence 72.9% and $r_5 \rightarrow r_{12}$ with confidence 73.1%, the confidence levels of these metarules are all inferior to those of the metarules mr_{36}, mr_{37}, mr_{40}, and mr_{41} relating rules 12, 13, 20, and 21 found in Table 6.

Another major difference between the metarules in Fig. 3 and the clusters of Figs. 6 and 7 is that the latter do not indicate the nature of the relationships between the clustered rules. Greater detail is obtained from the ability of the metarules to provide insights into containment. For example, mr_{18} indicates 100% confidence for $r_{16} \rightarrow r_1$, but mr_{19} shows only 81.8% confidence

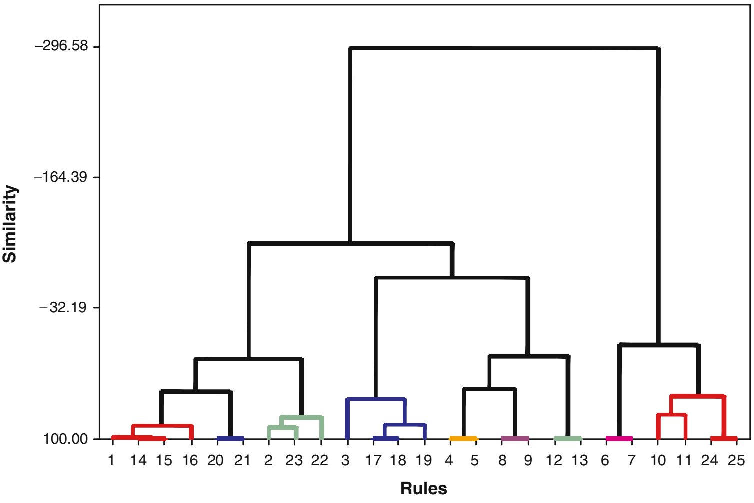


Fig. 6 Dendrogram of clustering association rules with Ward linkage and distance metric d_1

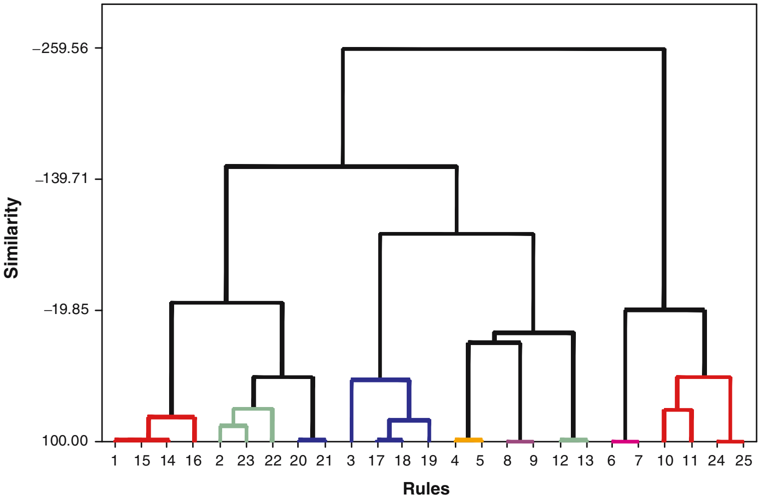


Fig. 7 Dendrogram of clustering association rules with Ward linkage and distance metric d_2

for $r_1 \rightarrow r_{16}$. The clustering results blend these two implications and place r_1 and r_{16} close to each other in the dendrograms, without a clear description of the 100% one-way relationship. Although the distance metrics give a sense of proximity between the rules, they do not indicate the level of containment or overlap of rules. Furthermore, the conclusions above continue to hold when linkages such as complete, average and single were used instead of Ward in the comparisons above.

6 Experimental evaluation

We show the organization provided by metarules on the following six benchmark data sets obtained from the UC Irvine ML repository (Blake and Merz 1998) and a Microarray data set obtained from the Kent Ridge Bio-medical data set repository (Li and Liu 2002).

- Iris Plants data (iris),
- Johns Hopkins University Ionosphere data (ion),
- Statlog Project Heart Disease data (hea),
- Thyroid Disease data (thy),
- Attitudes Toward Workplace Smoking Restrictions data (smo), and
- Mushroom data (mush),
- Ovarian Cancer (OvaCan).

The data sets, as described in Table 7, are of different sizes and varying number and types of attributes. The continuous attributes in the data sets used were discretized using a 4-bin equal-frequency discretization. After discretization we reduced the number of attributes of the Ovarian Cancer data set from 15,154 to

Table 7 Major properties of the data sets considered in the experimentation

Properties	Data sets						
	Iris	Thy	Ion	Smo	Hea	Mush	Ovacan
No of classes	3	3	2	3	2	2	2
No of examples	150	7,200	351	2,855	270	8,124	253
No of attributes	4	21	34	13	13	22	500
No of continuous attributes	4	6	32	2	6	0	500

500 by random selection. The a priori algorithm itself is known to consume too much memory with sparse data sets and this prevents it from returning any rules.

The experimental results are summarized in Table 8 where each row describes the results of one experiment that consisted of the following steps:

- Find the set of rules with the consequent matching the class labels specified in column 2. The support and confidence thresholds used to find association rules are reported in the third column. It should be noted that the thresholds used differ between class labels and data sets since the class labels were not distributed evenly on the data cases. The number of rules found in each case can be found in column 4. We used the default maximum itemset size for a priori except for the Ovarian Cancer data set where we reduced it to four in order to avoid too much memory consumption and enable a priori to return rules.
- Metarules were then mined following the procedure described in Sect. 4.1 with a support threshold of 0% and a confidence threshold of 100%. The number of metarules discovered is reported in column 5.
- After mining the metarules, the definition of rule equivalence given in Sect. 4.3 Definition 2 was applied and the number of remaining nodes after grouping the equivalent rules is reported in column 7. The number of remaining metarules in the simplified graph can be found in column 6. Column 8 is the ratio of the reduced number of metarules to the initial number of metarules. Finally column 9 indicates the ratio of the reduced number of rules to their initial number.

It should be noted that we do not prune the contained or the more specific rules. Our objective is to illustrate the effect of grouping equivalent rules on the metarule graphs.

The experimental results indicate that grouping the equivalent rules in the same node leads to the simplification of the metarules graph. The extent of simplification of the metarules graph varies between the different cases considered both in terms of number of remaining rules and number of remaining metarules; in some cases, such as for both class labels of the mushroom data set, both class labels of the ovarian cancer data set, for the first two class labels of the thyroid disease (thy) data set and for the first two class labels of the smoking (smo) data set, the metarule graphs were reduced more compared to the remaining experimental cases. The greatest simplification of rules was noted for the

Table 8 Experimental results

Data sets	Class labels	%Supp/%Conf of rules	No of rules	No of metarules	No of simplified metarules	No of simplified rules	% of remaining metarules (%)	% of remaining rules (%)
Hea	Absence(1)	10/80	276	1,321	421	177	31.87	64.13
	Presence(2)	10/90	95	250	96	69	38.40	72.63
Smo	Unrestricted(0)	0.05/85	326	7,093	17	50	0.24	15.34
	Prohibited(1)	0.01/85	267	1,683	46	129	2.73	48.31
	Restricted(2)	1/85	204	546	84	137	15.38	67.16
Iris	Setosa	1/95	51	318	102	32	32.08	62.75
	Virginica	1/95	59	334	155	42	46.41	71.19
	Versicolor	1/95	46	138	57	33	41.30	71.74
Ion	Good	10/95	1,115	9,032	1,369	469	15.16	42.06
	Bad	10/95	100	206	52	33	25.24	33.00
Thy	Class (1)	0.01/95	139	7,532	7	17	0.09	12.23
	Class (2)	0.01/95	669	39,920	53	60	0.13	8.97
	Class (3)	25/95	634	6,277	2,638	490	42.03	77.29
Mush	Poisonous	20/95	846	3,64,522	95	27	0.03	3.19
	Edible	20/95	528	85,273	858	93	1.01	17.61
Ovacan	Normal	20/80	56	336	0	8	0	14.28
	Cancer	20/85	1,651	59,572	945	292	1.58	17.68

Table 9 Effect of confidence threshold for metarules mining on the metarules graph representing the rules predicting the poisonous category from the mushroom data set

Confidence	No of metarules	No of Simplified metarules	No of Simplified rules	Remaining metarules(%)	Remaning rules (%)
100%	3,64,522	95	27	0.03	3.19
90%	5,19,275	19	10	0.00	1.18
80%	5,26,514	18	8	0.00	0.95

category Poisonous for the mushroom data set, where the number of rules went down from 846 to 27 rules. The highest simplification of metarules was, however, noted for the category Normal for the Ovarian Cancer data set, the number of metarules was reduced from 336 metarules to none. Finding the equivalent rules makes it easier to process the rules and the reduced metarules graph indicates the relationships between these groups of rules. Note that these results were achieved with a conservative metarule confidence of 100%. Relaxing the confidence threshold for metarules mining would further simplify the metarules graph. The following experiment illustrates how the metarules graph is simplified with the reduction of the confidence threshold for metarules mining. Consider the mushroom data with the class label "Poisonous." As the confidence threshold was reduced from 100 to 90% then 80%, the number of simplified rules and simplified metarules decreased as reported in Table 9.

Reducing the confidence threshold of metarules results in more metarules which in turn leads to more equivalent rules. This naturally leads to a decrease in the percentage of rules and metarules after grouping the equivalent rules. Once all the equivalent rules have been grouped, no further simplification occurs with the reduction of the confidence threshold of metarules mining. Consequently, the analyst can decide on a convenient confidence threshold in order to make the metarules graph match the redundancy to discover in the rules used in the application. A default of 100% confidence still resulted in dramatic reductions in these examples. Each group of equivalent rules in the metarules graph can then be further investigated individually to understand the structure of its representative rules. The simplifications in these examples did not yet use rule pruning based on antecedent itemsets or other criteria. We consider metarules as a first step to learn of the attribute masking and redundancy generated from sparse data in high dimensions that can be followed by further pruning.

7 Conclusion

Finding association rules is an efficient way to uncover all the hidden regularities in massive high-dimensional data. The huge number of discovered rules is a handicap for the human user to assimilate all the information provided by the rules. Since the rules express relationships that exist indeed in the data it is not prudent to arbitrarily prune and eliminate some of the rules. This paper introduces a new approach based on metarules and a graphical display to partition the association rules with the same consequent into independent subgroups of rules without assumptions about relevance of rules. These subgroups of rules are simplified even more after grouping the equivalent rules. The organized collection presents only data-determined relationships between the rules. Domain knowledge can also be applied as appropriate to possibly prune the more specific ones. The data analyst can analyze each of the formed subgroups of rules individually and therefore understand the relationships that hold between the data in different regions of the measurement space.

References

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

Aumann Y, Lindell Y (2003) A statistical theory for quantitative association rules. J Intell Inf Syst 20(3):255–283

Bay SD, Pazzani MJ (2001) Detecting group differences: mining contrast sets. Data Min Knowl Discov 5(3):213–246

Bayardo R, Agrawal R, Gunopulos D (2000) Constraint-based rule mining in large, dense databases. Data Min Knowl Discov J 4:217–240

Blake CL, Merz CJ (1998) UCI repository of machine learning databases. UC Irvine, Deptatment Information and Computer Science. http://www.ics.uci.edu/ mlearn/MLRepository.html

Chawla C, Davis, J Pandey G (2004) On local pruning of association rules using directed hypergraphs. In: Proceedings of the 20th international conference on data engineering, Boston, MA, USA, p 832

Dougherty J, Kohavi R, Sahami M (1995) Supervised and unsupervised discretization of continuous features. In: Proceedings of the international conference on machine learning, Tahoe City, CA, USA, pp 194–202

Fukuda T, Morimoto Y, Morishita S, Tokuyama T (1999) Mining optimized association procedures. Rules for numeric attributes. J Comput Syst Sci 58:1–12

Gupta KG, Strehl A, Ghosh J (1999) Distance based clustering of association rules. In: Proceedings of ANNIE, intelligent engineering systems through articial neural networks, St. Louis, MO, USA, pp 759–764

Han J, Pei J, Yin Y (2000) Mining frequent patterns without candidate generation. In: Proceedings of ACM SIGMOD international conference on management of data, Dallas, TX, USA, pp 1–12

Huang S, Webb GI (2005a) Discarding insignificant rules during impact rule discovery in large databases. In: Proceedings of the SIAM 2005 data mining conference, Newport Beach, CA, USA, pp 541–545

Huang S, Webb GI (2005b) Pruning derivative partial rules during impact rule discovery. In: Proceedings of PAKDD the 9th Pacific-Asia conference on advances in knowledge discovery and data mining. Lecture notes in computer science, vol 3518, Hanoi, Vietnam, pp 71–80

Klemettinen M, Mannila H, Ronkainen P, Toivonen H, Verkamo AI (1994) Finding interesting rules from large sets of discovered association rules. In: Proceedings of the international conference on information and knowledge management, CIKM, Gaithersburg, MD, USA, pp 401–407

Lent B, Swami A, Widom J (1997) Clustering association rules. In: Proceedings of the 13th international conference on data engineering, IEEE Computer Society, Birmingham, UK, pp 220–231

Li J, Liu H (2002) Kent ridge bio-medical data set repository, Institute for Infocomm Research. http://sdmc.lit.org.sg/GEDatasets/Datasets.html

Li W, Han J, Pei, J (2001) Accurate and efficient classification based on multiple class-association rules. In: International conference on data mining, ICDM, San Jose, CA, USA, pp 369–376

Liu B, Hsu W (1996) Post analysis of the learned rules. In: Proceedings of the 13th national conference on artificial intelligence, Portland, OR, pp 828–834

Liu B, Hsu W, Chen S (1997) Using general impression to analyze discovered classification rules. In: Proceedings of the 3rd international conference on knowledge discovery and data mining, Newport Beach, CA, USA, pp 31–36

Liu B, Hsu W, Ma Y (1998) Integrating classification and association rule mining. In: Proceedings of the 4th international conference on knowledge discovery and data mining, KDD, New York, USA, pp 80–86

Liu B, Hsu W, Ma Y (1999) Pruning and summarizing the discovered associations. In: Proceedings of ACM SIGKDD international conference on knowledge discovery and data mining, San Diego, CA, USA, pp 125–134

Liu H, Hussain F, Tan CL, Dash M (2002) Discretization: an enabling technique. Data Min Knowl Discov 6(4):393-423

Ng R, Lakshmanan LVS, Han J, Pang A (1998) Exploratory mining and pruning optimizations of constrained associations rules. In: Proceedings of ACM SIGMOD international conference on management of data, Seattle, WA, pp 13–24

Padmanabhan B, Tuzhilin A (1998) A belief-driven method for discovering unexpected patterns. In: Proceedings of the 4th international conference on knowledge discovery and data mining, KDD, New York, USA, pp 94–100

Quinlan JR (1992) C4.5: program for machine learning. Morgan Kaufmann, Los Atlos, CA

Scott DW, Thompson JR (1983) Probability density estimation in higher dimensions. In: Proceedings of the 15th symposium on the interface, North-Holland, pp 173–179

Silberschatz A, Tuzhilin A (1996) What makes patterns interesting in knowledge discovery systems. IEEE Trans Knowl Data Eng 8(6):970–974

Srikant R, Agrawal R (1996) Mining quantitative association rules in large relational tables. In: Proceedings of the ACM SIGMOD conference on management of data, Montreal, Quebec, Canada

Srikant R, Vu Q, Agrawal R (1997) Mining association rules with item constraints. In: Proceedings of the 3rd international conference on knowledge discovery and data mining, KDD, Newport Beach, CA, USA, pp 67–73

Tan P-N, Kumar V, Srivastava J (2002) Selecting the right interestingness measure for association patterns. In: Proceedings of the 8th ACM SIGKDD, international conference on knowledge discovery and data mining, Madison, WI, USA, p 183

Toivonen H, Klemetinen M, Ronkainen P, Hatonen K, Mannila H (1995) Pruning and grouping discovered association rules. In: Proceedings of the Mlnet workshop on statistics, machine learning, and discovery in databases, Heraklion, Crete, Greece, pp 47–52

Yang Y, Webb GI (2002) A comparative study of discretization methods for naive-bayes classifiers. In: Proceedings of the Pacific Rim knowledge acquisition workshop, PKAW, Tokyo, pp 159–173