Reducing Redundancy in Characteristic Rule Discovery by Using IP-Techniques

Tom Brijs, Koen Vanhoof and Geert Wets

Limburg University Centre, Faculty of Applied Economic Sciences,

B-3590 Diepenbeek, Belgium

{tom.brijs; koen.vanhoof; geert.wets}@luc.ac.be

Abstract

The discovery of characteristic rules is a well-known data mining technique and has lead to several successful applications. Unfortunately, typically a (very) large number of rules is discovered during the mining stage. This makes monitoring and control of these rules extremely costly and difficult. Therefore, a selection of the most promising rules is desirable. In this paper, we propose an integer programming model to solve the problem of selecting the most promising subset of characteristic rules. The proposed technique allows to control a user-defined level of overall quality of the model in combination with a maximum reduction of the redundancy extant in the original ruleset. We use real-world data to evaluate the performance of the proposed technique against the well-known RuleCover heuristic.

1 Introduction

Data mining is the automated search for hidden, previously unknown and potentially useful information from large databases. Moreover, data mining is a crucial phase in the KDD (Knowledge Discovery in Databases) process [Fayyad, Piatetsky-Shapiro & Smyth 1996]. In fact, two important goals of KDD can be identified, more specifically *prediction*, i.e. the use of training data to construct a model to predict unknown values of future instances, and *description*, i.e. the search for interesting patterns and their (re)presentation in an easy, understandable format. In this paper, we are especially interested in the latter objective, namely description.

One of the most well-known data mining techniques to extract descriptive information from data is the discovery of characteristic rules. Among the advantages of characteristic rules are clearly its natural representation and the ease of integration of the discovered rules with background knowledge. Several successful applications [Viveros, Nearhos & Rothman 1996; Ali, Manganaris & Srikant 1997; Bloemer, Brijs, Swinnen & Vanhoof 1998] demonstrate the usefulness of this technique. However, also disadvantages of characteristic rules can be identified. Firstly, often a large number of rules is discovered during the mining stage. This makes monitoring and control of these rules extremely costly and difficult. Secondly, characteristic rules often suffer from being *incomplete*, i.e. not all instances are covered by the set of discovered rules, and *not being mutually exlusive*, i.e. some instances may be covered by more than one rule. Previous researchers have already highlighted this problem. In their study on the *interestingness* of association rules Klementtinen, Mannila, Ronkainen, Toivonen & Verkamo [1994] concluded: 'A problem that remains is redundancy. Large amounts of rules could potentially be pruned, if there were appropriate ways to remove redundant or nearly redundant rules'. Indeed, with characteristic (and association) rule discovery, instances may be covered by multiple rules with the consequence that some rules may be overlapping, i.e. describing the same database rows. In this paper, we specifically focus on this problem of redundancy.

The objective of this paper consists of constructing a method which is able to reduce the redundancy extant in the set of rules discovered during the mining stage. However, we also want to influence the pruning process in a sense that some overall measure of quality of the reduced set can be controlled. Indeed, several characteristic rule quality measures exist and therefore we should take some measure of overall model quality into account during the pruning process.

The outline of the remainder of this paper is as follows. In section 2, we introduce the discovery of characteristic rules and present a graphical illustration of the key issue of this paper, i.e. redundancy reduction. Section 3 provides an overview of the previous work on the issue of *interestingness* in order to put the key issue of this work in a global perspective. Section 4 formalizes the concrete problem of redundancy reduction and introduces a novel solution to reduce the redundancy in a set of characteristic rules by using integer programming techniques. In section 5, we compare our models with the well-known RuleCover heuristic and discuss the empirical results. Finally, section 6 summerizes our work.

2 Problem situation

2.1 Characteristic rules

Let $A = \{a_1, a_2, \ldots, a_k\}$ be a set of literals, called attributes. Let D be a database of instances, where each instance I is a set of attributes such that $I \subseteq A$. Associated with each instance is a unique identifier, called its *TID*. We say that an instance I *contains* X, a set of some attributes in A, if $X \subseteq I$.

Definition 1 Characteristic Rule
A *characteristic rule* is an implication of the form $Y \Rightarrow X$, where $X \subset A$, $Y \in A$, and $X \cap Y = \varnothing$. ■

Definition 2 Completeness
The rule $Y \Rightarrow X$ is s% *complete*, if X covers s% of the instances satisfying Y. ■

Definition 3 Discriminant power
The rule $Y \Rightarrow X$ is c% *discriminant*, if X covers (100 - c) % of the $\neg Y$ instances. ■

Different approaches to the characteristic rule discovery problem exist. Recently, Maeda, Maki and Akimori [1998] proposed the CHRIS (Characteristic Rule Induction by Subspace search) rule induction algorithm which uses a *utility measure* to define the interestingness of rules. It is designed to extract a user-defined number of rules that have the highest utility measures among all possible rules.

The LCHR algorithm [Cai, Cercone, Han 1991] takes database rows relevant to the target class, i.e. the concept to be described, and adopts the *least commitment principle* (that is, commitment to minimally generalized concepts) by ascending a concept tree only when necessary. LCHR produces rules that hold for all the examples in the target class. However, the authors also propose extensions to the LCHR technique to produce characteristic rules in the case of exceptions or noisy data.

The KID3 algorithm [Piatetsky-Shapiro 1991] finds all simple exact (or almost exact) characteristic rules of the form $cond(A) \Rightarrow cond(B)$. KID3 is implemented with a hashing structure and can be run in parallel. Extending KID3 to more complex conditions is also discussed.

In this paper, a different approach to characteristic rule induction is applied. We use association rules to generate all characteristic rules for a given class that have a minimum support within that given class. More specifically, this procedure involves the discovery of all *frequent itemsets* in that class exceeding a minimum user-defined support threshold. In fact, by using a minimum support threshold, we are certain that all discovered rules are minimaly s% *complete*. The discovery of frequent itemsets has been studied extensively in the literature on association rules [Mannila 1997; Agrawal & Srikant 1994; Agrawal, Imielinski & Swami, 1993] of which the following provides a short formal overview.

Definition 4 Frequency of an itemset
$\sigma(X, D)$ represents the frequency of itemset X in D, i.e. the fraction of transactions in the database D that contain X. ■

Definition 5 Frequent itemset
Itemset X is called frequent in D, if $\sigma(X, D) \geq s$ with s the *frequency threshold*. ■

If we constrain D to be the collection of transactions for which the target class equals Y, then the frequency of the itemset X explicitly determines the *completeness* of the rule $Y \Rightarrow X$. Thus by looking for *frequent itemsets* X within the collection of transactions with target attribute Y, we can be sure to retain all characteristic rules $Y \Rightarrow X$ having a minimum completeness of s%.

A typical approach [Agrawal, Mannila, Srikant, Toivonen & Verkamo 1996] to discover all frequent itemsets X is to use the knowledge that all subsets of a frequent itemset must also be frequent. This insight simplifies the discovery of all frequent itemsets considerably, i.e. first find all frequent itemsets of size 1 by reading the data once and recording the number of times each attribute A occurs. Then form *candidate* itemsets of size 2 by taking all pairs $\{B, C\}$ of attributes such that $\{B\}$ and $\{C\}$ both are frequent. The frequency of the candidate itemsets is again evaluated against the database. Once frequent itemsets of size 2 are known, candidate itemsets of size 3 can be formed; these are itemsets $\{B, C, D\}$ such that $\{B, C\}$, $\{B, D\}$ and $\{C, D\}$ are all frequent. This process is continued until no more candidate itemsets can be formed. Then, the presentation of characteristic rules is easy, i.e. for each frequent itemset X_i a rule is constructed of the form $Y \Rightarrow A_1 \wedge ... \wedge A_k$ with $A_1, ..., A_k \in X_i$.

2.2 Redundancy: graphical problem illustration

In most applications, typically a (very) large number of characteristic rules is discovered. Furthermore, because characteristic rules describe properties that are common to many or all instances of a class, different rules may describe different properties of the same instances. Consequently, mutual exclusivity in the discovered set of rules cannot be guaranteed, i.e. some instances in the database are covered by multiple rules. While the above-mentioned parameters *completeness* and *discriminant power* present can be used to filter less-interesting rules, these measures do not guarantee mutual exclusivity in the discovered set of characteristic rules. Therefore, other methods are needed to reduce the level of redundancy that is present in the discovered set of characteristic rules. Graphically, redundancy can be represented as follows:

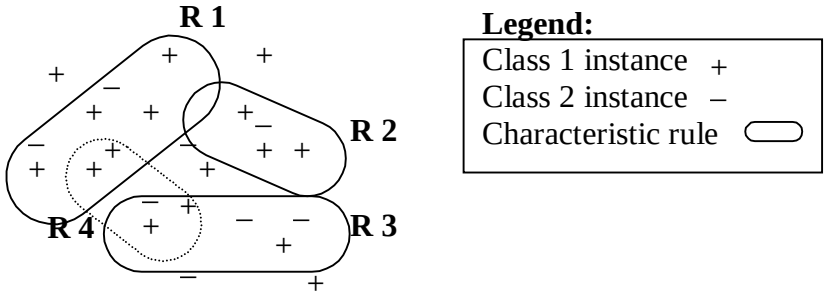

Figure 1: Redundancy in characteristic rules

In figure 1, it can be observed that rule number 4 (dashed line) does not cover any instances in addition to the instances already covered by the other rules (rule 1 and rule 3) in the model. We suggest that rule 4 is redundant and therefore, it should be discarded. However, one must be careful in cutting away characteristic rules from the model, because:

1. Discarding rules can result in reducing the covered instance space and this may not be recommended.

2. The final set of selected rules should describe as many positive instances as possible and as few negative instances as possible when compared to the original ruleset (i.e. discriminant power).

3 Previous work

Gago and Bento [1998] propose a distance metric[1] between rules to select the most heterogeneous set of rules that together gives a good coverage of the instance space. The method however has several drawbacks. Fist of all, the method can only be applied if the underlying data follows a uniform distribution. Secondly, three weight parameters are specified in the distance function but there is no concrete guidance for reasonable values of these parameters. Thirdly, outliers in the data can significantly affect the percentage of overlap of two rules. And finally, the distance function can return negative values while distance in literature is generally assumed to be positive.

In another approach, the use of *rule covers* was proposed by [Toivonen, Klemettinen, Ronkainen, Hätönen & Mannila 1995] to reduce redundancy in a discovered set of association rules. The RuleCover algorithm is a heuristic method and its results will be used as a benchmark against the results of our integer programming method (see section 4).

In fact, the selection of a 'minimum redundancy' set of rules can be seen in the larger framework of discovering *interesting* rules. Indeed, typically in data mining, only a small fraction of the rules generated may actually be of interest to the user. In this context, measures of rule *interestingness* must be used to filter out less interesting rules. In general, two types of rule interestingness measures can be defined, i.e. subjective and objective measures. Subjective measures are user-dependent, this means that each user may have different ideas about the interestingness of the discovered set of rules. Subjective interestingness measures include *unexpectedness*[2] [Silberschatz & Tuzhilin 1995, Liu & Hsu 1996, Padmanabhan & Tuzhilin 1998, Freitas 1998] or *actionability* [Piatetsky-Shapiro & Matheus 1994, Adomavicius & Tuzhilin 1997]. The user can also define *templates* [Klemettinen, Mannila, Ronkainen, Toivonen & Verkamo 1994], *general impressions* [Liu, Hsu & Chen 1997] or define *item constraints* [Srikant, Vu & Agrawal 1997]. On the other hand, objective measures of rule interestingness are based on the structure of the rules and

[1] The distance metric is inspired by a measure to calculate the distance between cases
[2] The more surprising a rule, the more interesting it is for the user. This measure is also called *surprisingness.*

the statistics associated with them. Objective measures include *J-measure* [Smyth & Goodman 1991, Wang, Tay & Liu 1998], *certainty* [Hong & Mao 1991], *RI* [Piatetsky-Shapiro 1991] and *strength* [Dhar & Tuzhilin 1993]. More recent measures of objective interestingness include *R-interestingness* [Srikant & Agrawal 1995], *intensity of implication* [Suzuki & Kodratoff 1998, Guillaume, Guillet & Philippé 1998], *discrimination* [Gray & Orlowska 1998]. Kamber and Shinghal [1996] propose specific measures of rule interestingness for characteristic rules based on *necessity* and *sufficiency*.

Specifically with respect to the issue of *redundancy* not so much work has been carried out. Hoschka and Klösgen [1991] deal with the problem of redundancy in their Explora system. It uses partial orderings of attributes and attribute sets to avoid presenting several kinds of redundant knowledge. Bayardo [1997] proposes a pruning strategy called *redundancy exploitation*. The idea is to prevent continued effort at classifying instances already classified by existing rules with high confidence. In the research community involved in validation and verification of knowledge based systems, redundancy has mainly been studied from the syntactical point of view [Preece 1991, Preece & Shingal 1994, Van Harmelen 1996].

4 Solution: integer programming

4.1 Algebraic problem definition

Consider the following *instance-rule* matrix (see table 1):

Table 1: instance-rule matrix

Index *j* →

Index *i* ↓	Rule *1*	Rule *2*	Rule ...	Rule *K*
Instance *1*	1	0		1
Instance ...	1	1		1
Instance *I*	0	1		0
Instance *J*	0	0		1
Instance ...	1	0		0
Instance *N*	0	1		0

The matrix shows *K* rules and *N* instances. Depending on the number of classes, the instance space is subdivided in two or more groups. For example, in the above matrix two groups of instances can be identified. One group belongs to a first (positive) class and carries the index values 1 to *I*, the other group of instances belongs to a second (negative) class and carries the index values *J* to *N*. The matrix shows whether a particular instance *i* is covered by a certain characteristic rule *j* or not. Formally, we define:

$$\boldsymbol{d_{i,j}} = \begin{cases} 1 & \text{if instance } i \text{ is covered by rule } j \\ 0 & \text{if instance } i \text{ is } \underline{\text{not}} \text{ covered by rule } j \end{cases}$$

Now, consider the formulation of the following Integer Programming [IP] model:

4.2 Model specification

MODEL 1: Maximal redundancy reduction

Let: i = number of instances
j = number of characteristic rules

Given: $d_{i,j}$

Decision variables: x_j

Target function: $MinZ = \sum_{i=1}^{I} \sum_{j=1}^{K} d_{i,j} * x_j$

Subject to: $\forall i (i = 1 \rightarrow I) : \sum_{j=1}^{K} x_j * d_{i,j} \geq 1$

The decision variable x_j is binary-valued and specifies whether characteristic rule j will be included in the final ruleset. The target function specifies that the model should look for patterns that have a minimum overlap of instances as possible in the group of positive (class 1) instances. This means that the model searches for characteristic rules that are as far apart as possible in the class 1 instance space. The constraint in the model is used to ensure that the original class 1 instance space is not reduced so that the final ruleset will still cover all class 1 instances that were covered by the original ruleset. Otherwise, when neglecting this constraint, the model would select no rules at all because the objective function forces the model to select as few characteristic rules as possible.

While providing an optimal solution to the redudancy reduction problem, the model presented above still suffers from a few imperfections:

First of all, many real world problems are characterized by certain levels of noise in the data caused by inconsistencies in the data (i.e. the same entry is labeled as belonging to two different classes). Therefore, the model should be adapted to account for certain levels of noise in the data. Secondly, we should take the discriminant power of characteristic rules into account (see definition 3). Indeed, when selecting rules for the final (reduncancy reduced) ruleset, it is appropriate to select characteristic rules that, as a group, cover as few negative (class 2) instances as possible.

To accomplish this, we introduce explicit, user-defined bounds on the coverage of positive and negative instances of the final ruleset. More specifically, consider the statements below:

α = proportion of class 1 instances that are covered by the final ruleset

100 - α = proportion of class 1 instances that are uncovered by the final ruleset

β = proportion of class 2 instances that are uncovered by the final ruleset

100 - β = proportion of class 2 instances that are covered by the final ruleset

When all class 1 instances are covered by the final ruleset, then α = 100 %. However, to account for a certain level of noise in the data, we specify that it is allowed to leave a certain proportion (100 - α) of the positive (class 1) instances uncovered. To control the discriminant power, i.e. the proportion of negative (class 2) instances that are covered by the final ruleset, we specify that no more than (100 - β) percent of the negative (class 2) instances can be covered.

Integrating these improvements into the former model results in the following model.

MODEL 2: Incorporating noise and discriminant power

Let: i = number of instances
j = number of patterns

Given: $d_{i,j}, W_1, W_2, \alpha, \beta$

Decision variables: x_j

Target function: $MinZ = W_1 \sum_{i=1}^{I} \sum_{j=1}^{K} d_{i,j} * x_j + W_2 \sum_{i=J}^{N} \sum_{j=1}^{K} d_{i,j} * x_j$

Subject to:

$$\forall i (i = 1 \rightarrow I) : \sum_{j=1}^{K} x_j * d_{i,j} + s_i \geq 1$$

$$\forall i (i = J \rightarrow N) : \sum_{j=1}^{K} x_j * d_{i,j} - M * t_i \leq 0$$

$$\sum_{i=1}^{I} s_i \leq (100 - \alpha) * I$$

$$\sum_{i=J}^{N} t_i \leq (100 - \beta) * (N - I)$$

In the above model, s_i and t_i represent *slack*-variables and M represents an infinitely large number. The W_1 and W_2 parameters are continuous weight values to correct for possible bias in the target function as a result of a different number of instances in each class. When class 2 contains more instances than class 1, then $W_1 > W_2$, i.e. $W_1 = (N - I)/I$. The *slack*-variables enable us to control the coverage of class 1 and class 2 instances by the final ruleset. For instance, when the sum of all s_i equals 10, this implies that the final ruleset is allowed to leave 10 positive instances uncovered. Therefore, the final two restrictions specify that no more than 100 - α percent of the class 1 instances may be left uncovered and that no more than 100 - β percent of the class 2 instances can be covered.

The model, however, is not guaranteed to reach an optimal solution, depending on the choice of the values of the parameter values α and β. For example, if α and β are to high, reaching an optimal solution may be impossible. Indeed, it will then be difficult for the model to find a good set of rules having a low degree of redundancy but also covering at least α percent of class 1 instances and covering less than 100 - β percent of class 2 instances. When discussing the empirical results (section 5), we will elaborate on this and propose guidelines for appropriate settings for the α and β parameters.

5 Empirical evaluation

To assess the performance of the proposed method, we will use the results of a previous research [Bloemer, Brijs, Swinnen & Vanhoof 1988]. In short, in the latter study, data from a customer satisfaction survey, carried out by a leading Belgian bank, were used to identify characteristic rules for dissatisfaction. With these rules, latently dissatisfied customers were identified. It turned out that 29 characteristic rules for dissatisfaction were found to be *interesting*[3]. However, closer observation of the discovered set of rules revealed considerable redundancy. Therefore, as a post-processing step, the integer programming methods, presented in section 4.2, will be used to reduce the redundancy and select a smaller set of rules.

We will compare the results of our integer programming method against those obtained from the heuristic method of *rule covering* proposed by [Toivonen, Klemettinen, Ronkainen, Hätönen & Mannila 1995]. In short, the RuleCover algorithm works as follows: a *greedy* algorithm uses an original set Γ (containing the entire set of characertistic rules) and then iteratively selects a rule $X_i \Rightarrow Y$ to move it into Δ. In each pass the rule is selected that covers the maximum number of instances that are left over after having deleted the instances that were covered by the rule that was selected during the previous pass. This process continues until no instances are left over. At the end, Δ contains the minimum rule cover of Γ. In paragraph 5.1 and 5.2, the results of the empirical research will be highlighted.

5.1 Maximal redundancy reduction (Model 1)

In the first experiment we compare the RuleCover heuristic with the proposed integer programming model to select a ruleset with minimal redundancy. In fact, this means that for IP-model 1 in section 4.2, the redundancy in class 1 has to be minimized, regardless of the performance of the ruleset in class 2, i.e. without worrying about the discriminant power of the final ruleset.

Empirical results show that the IP-model is able to select fewer rules than the RuleCover algorithm. RuleCover returns 15 rules while the integer programming algorithm returns only 13 rules that are able to cover *all* class 1 instances. Furthermore, the redundancy is significantly different when comparing the two techniques. When calculating the average number of times each class 1 instance is covered by the final ruleset, for the RuleCover algorithm this figure amounts to 5.02 whereas for the optimal IP-model this figure only amounts to 4.34. This again illustrates that RuleCover is a heuristic and therefore it cannot guarantee an optimal solution for the redundancy reduction problem. 11 out of the 15 rules that were selected by the RuleCover algorithm were also selected by our method. However, it must be clear that no attention is paid to the discriminating power of the resulting ruleset. In fact, it is possible that RuleCover returns more rules, i.e. it produces more redundancy, but that the discriminant power in terms of the coverage of class 2 instances is better (i.e. it covers fewer class 2 instances) than the one obtained by the

[3] Defined as the difference between the percentage coverage of positive instances and the percentage coverage within the total group of instances (i.e. positive and negative)

integer programming model. Therefore, we introduce model 2 to incorporate noise and discriminant power.

5.2 Incorporating noise and discriminant power (model 2)

Firstly, the number of negative (class 2) instances that is covered by the final ruleset, as a whole, is an important indicator of the discriminant power of the final ruleset, and therefore it should play a leading role in the selection of rules for the final ruleset. Secondly, noise in the data may cause the retention of too many characteristic rules in order to cover all positive (class 1) instances. Indeed, characteristic rule discovery involves looking for rules that summerize one or more properties common to all (or many) instances of a certain class. Most real-world phenomena, however, are characterized by uncertainty resulting in a certain level of noise in the data. Therefore, setting α equal to 100% would be irrealistic.

From the 15 patterns selected by RuleCover, the first four patterns are able to cover 81% of the class 1 instances, which is very reasonable. However, the ruleset as a whole also covers 62.4% of the class 2 instances (low discriminant power). Model 2 in section 4.2 can be used to select the minimum set of rules that achieves the same coverage of class 1 instances while covering less than 62.4% of the class 2 instances. More specifically, by setting α equal to 81% and (1 - β) equal to 62.4%, the final ruleset selected by the integer programming model covers 59% of the negative (class 2) instances. This indicates that the parameter values for α and β obtained from examining the results of RuleCover are good lower (for α) and upper (for 1 - β) bounds for the parameter values to be used in the optimization model. In general, the coverage of negative (class 2) instances is high. However, while the primary objective of the algorithm is to reduce redundancy and not to maximally discriminate between the two target groups, the following results are more important. In analogy with section 5.1, the degree of redundancy can be expressed as the average number of times each instance is covered by the final ruleset. For the ruleset obtained from the RuleCover heuristic, this figure amounts to 1.54 whereas for the optimal IP-model this figure only amounts to 1.32.

6 Conclusions

In this paper, we introduced two integer programming models to tackle the problem of redundancy in a set of characteristic rules. The first model searches for an optimal selection of rules that is able to maximally reduce redundancy under the constraint of covering all (positive) instances that are covered by the original ruleset. In the second model, the first model was adapted to account for noise in the data and to impose a quality criterion, i.e. discriminant power, on the final ruleset. Both models were empirically tested on real-world data and compared with the well-known RuleCover heuristic. It was found that the IP models are indeed able to produce significantly better results than the RuleCover heuristic. Firstly, in terms of the number of characteristic rules that are retained for the final ruleset. Secondly in terms of the discriminant power of the final rulset and finally also in terms of the total redundancy that remains in the final ruleset.

References

Adomavicius G., and Tuzhilin A. (1997). Discovery of Actionable Patterns in Databases: The Action Hierarchy Approach, *in Proceedings of the Third International Conference of Knowledge Discovery & Data Mining*, The AAAI Press, 111-114.

Agrawal R., Imielinski T., and Swami A. (1993). Mining association rules between sets of items in large databases, *in Proceedings of ACM SIGMOD Conference on Management of Data (SIGMOD'93),*Washington, D.C., USA: ACM, 207-216.

Agrawal R., Mannila H., Srikant R., Toivonen H., and Verkamo A.I. (1996). Fast Discovery of Association Rules, *in Advances in Knowledge Discovery and Data Mining*, The AAAI Press, 307-328.

Agrawal R., and Srikant R. (1994). Fast Algorithms for Mining Association Rules in Large Databases, *in Proceedings of the 20th VLDB Conference*, 487-499.

Ali K., Manganaris S., and Srikant R. (1997). Partial Classification using Association Rules, *in Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, The AAAI Press, 115-118.

Bayardo R. J. Jr. (1997). Brute-Force Mining of High-Confidence Classification Rules, *in Proceedings of the Third International Conference of Knowledge Discovery and Data Mining*, The AAAI Press, 123-126.

Bloemer J., Brijs T., Swinnen G., and Vanhoof K. (1998). Using Association Rules in Customer Satisfaction Studies to Identify Latent Dissatisfied Customers, *in ESOMAR publication series*, ISBN: 92-831-1274-1, 102-110.

Cai Y., Cercone N., and Han J. (1991). Attribute-Oriented Induction in Relational Databases, *in Knowledge Discovery in Databases*, The AAAI Press, 213-228.

Dhar V., and Tuzhilin A. (1993). Abstract-driven pattern discovery in databases, *in IEEE Transactions on Knowledge and Data Engineering*, 5(6).

Fayyad U.M., Piatetsky-Shapiro G., and Smyth P. (1996). From Data Mining to Knowledge Discovery: An Overview, *in Advances in Knowledge Discovery and Data Mining*, AAAI Press / The MIT Press, 1-34.

Freitas A. (1998). On Objective Measures of Rule Surprisingness, *in Proceedings of the Second European Symposium, PKDD'98*, Lecture Notes in Artificial Intelligence 1510, 1-9.

Gago P., and Bento C. (1998). A Metric for Selection of the Most Promising Rules, *in Proceedings of the second European Symposium, PKDD98*, Lecture Notes in Artificial Intelligence 1510, 19-27.

Gray B., and Orlowska M.E. (1998). CCAIIA: Clustering Categorical Attributes into Interesting Association Rules, *in Proceedings of the second Pacific-Asia Conference, PAKDD-98*, Lecture Notes in Artificial Intelligence 1394, 132-143.

Guillaume S., Guillet F., and Philippé J. (1998). Improving the Discovery of Association Rules with Intensity of Implication, *in Proceedings of the second European Symposium, PKDD98,* Lecture Notes in Artificial Intelligence 1510, 318-327.

Hong J., and Mao C. (1991). Incremental Discovery of Rules and Structure by Hierarchical and Parallel Clustering, *in Knowledge Discovery in Databases*, 177-194.

Hoschka R., and Klösgen W. (1991). A Support System for Interpreting Statistical Data, *in Knowledge Discovery in Databases*, 325-345.

Kamber M., and Shinghal R. (1996). Evaluating the Interestingness of Characteristic Rules, *in Proceedings of the second International Conference on Knowledge Discovery & Data Mining,* The AAAI Press, 263-266.

Klemettinen M., Mannila H., Ronkainen P., Toivonen H., and Verkamo A.I. (1994). Finding Interesting Rules from Large Sets of Discovered Association Rules, *in the Third International Conference on Information and Knowledge Management*, ACM Press, 401-407.

Liu B., Hsu W., and Chen S. (1997). Using General Impressions to Analyze Discovered Classification Rules, *in Proceedings of the Third International Conference on Knowledge Discovery & Data Mining*, The AAAI Press, 31-36.

Maeda A., Maki H., and Akimori H. (1998). Characteristic Rule Induction Algorithm for Data Mining, *in Proceedings of the second Pacific-Asia Conference, PAKDD98*, Lecture Notes in Artificial Intelligence 1394, 399-400.

Mannila, H. (1997), "Methods and problems in data mining", *Proceedings of the International Conference on Database Theory*, 41-55.

Padmanabhan B., and Tuzhilin A. (1998). A Belief-Driven Method for Discovering Unexpected Patterns, *in Proceedings of the Fourth International Conference on Knowledge Discovery & Data Mining*, The AAAI Press, 94-100.

Piatetsky-Shapiro G. (1991). Discovery, Analysis, and Presentation of Stron Rules, *in Knowledge Discovery in Databases*, The AAAI Press, 229-248.

Piatetsky-Shapiro G., and Matheus C.J. (1994). The Interestingness of Deviations, *in Proceedings of the AAAI-94 Workshop on Knowledge Discovery in Databases*, 25-36.

Silberschatz A., and Tuzhilin A. (1995). On Subjective Measures of Interestingness in Knowledge Discovery, *in Proceedings of the First International Conference on Knowledge Discovery & Data Mining*, The AAAI Press, 275-281.

Smyth P., and Goodman R.M. (1991). Rule Induction Using Information Theory, in *Knowledge Discovery in Databases*, 159-176.

Srikant R., and Agrawal R. (1995). Mining Generalized Association Rules, *Research Report IBM Research Division.*

Srikant R., Vu Q., and Agrawal R. (1997). Mining Association Rules with Item Constraints, *in Proceedings of the Third International Conference of Knowledge Discovery & Data Mining*, The AAAI Press, 67-73.

Suzuki E., and Kodratoff Y. (1998). Discovery of Surprising Exception Rules Based on Intensity of Implication, *in Proceedings of the second European Symposium, PKDD98*, Lecture Notes in Artificial Intelligence 1510, 10-18.

Toivonen H., Klemettinen M., Ronkainen P., Hätönen K., and Mannila H. (1995). Pruning and Grouping of Discovered Association Rules, *in MLnet Workshop on Statistics, Machine Learning, and Discovery in Databases*, Heraklion, Crete, Greece, April 1995.

Viveros M.S., Nearhos J.P., and Rothman M.J. (1996). Applying Data Mining Techniques to a Health Insurance Information System, *in Proceedings of the 22nd VLDB Conference*, 286-294.

Wang K., Tay S.H.W., and Liu B. (1998). Interestingness-Based Interval Merger for Numeric Association Rules, *in Proceedings of the fourth International Conference on Knowledge and Data Mining*, 121-127.