

Efficient Mining of Dissociation Rules

Mikołaj Morzy

Institute of Computing Science
Poznań University of Technology
Piotrowo 2, 60-965 Poznań, Poland
`Mikolaj.Morzy@put.poznan.pl`

Abstract. Association rule mining is one of the most popular data mining techniques. Significant work has been done to extend the basic association rule framework to allow for mining rules with negation. Negative association rules indicate the presence of negative correlation between items and can reveal valuable knowledge about examined dataset. Unfortunately, the sparsity of the input data significantly reduces practical usability of negative association rules, even if additional pruning of discovered rules is performed. In this paper we introduce the concept of dissociation rules. Dissociation rules present a significant simplification over sophisticated negative association rule framework, while keeping the set of returned patterns concise and actionable. A new formulation of the problem allows us to present an efficient algorithm for mining dissociation rules. Experiments conducted on synthetic datasets prove the effectiveness of the proposed solution.

1 Introduction

Operational databases and enterprise data warehouses contain limitless volumes of data. Valuable knowledge is hidden in these data under the form of trends, regularities, correlations, and outliers. Traditional database and data warehouse querying models are not sufficient to extract this knowledge. The value of the data, as provided by traditional databases, can be greatly increased by adding the means to automatically discover useful knowledge from large volumes of gathered data. Data mining, a novel research discipline, aims at the discovery and extraction of useful, previously unknown, non-trivial, and ultimately understandable patterns from large databases and data warehouses [1]. Data mining uses methods from statistics, machine learning, artificial intelligence, databases, and other related disciplines to extract unknown, utilitarian, and interesting rules and regularities in order to assist users in informed decision making.

One of the most successful and widely used data mining techniques is association rule mining. Association rules [3] represent the patterns of co-occurrence of items in a large collection of sets. An example of an association rule is an expression of the form *'wine' \wedge 'grapes' \Rightarrow 'cheese' \wedge 'white bread'*, which represents the fact that purchasing wine and grapes implies purchasing cheese and white bread in the same transaction. Association rules can be easily applied to

countless application domains. For instance, in market basket analysis the collection of sets corresponds to the database of customer transactions and each set corresponds to a set of products purchased by a customer during a single visit to the store. Discovered rules can be used for organizing cross-sales, designing mail catalogs, or reorganizing shelf layout.

Association rules capture only the "positive knowledge", i.e., sets of items comprising associations are always positively associated. One might be interested in discovering "negative knowledge" expressed as negative associations between items. An example of such pattern is an expression of the form '*FC Barcelona jersey*' $\Rightarrow \neg$ '*Real Madrid cap*' $\wedge \neg$ '*Real Madrid scarf*', which represents the fact, that a customer who purchases an FC Barcelona jersey will almost never buy either a cap or a scarf with Real Madrid logo. Another type of a pattern conveying "negative knowledge" is an expression where only certain elements in the antecedent or consequent are negated. An example of such an expression is '*beer*' \wedge '*sausage*' \Rightarrow '*mustard*' $\wedge \neg$ '*red wine*', which represents the fact, that transactions containing beer and sausages usually contain mustard and do not contain red wine. Negative associations can be successfully used in several application domains to identify conflicting or complementary sets of products.

Unfortunately, incorporating negation into association rule framework is very difficult. Due to the sparsity of data measured as the ratio of the average number of items per transaction to the total number of possible items, the number of possible association rules with negation is huge. Discovered patterns are valid, if they are useful and utilitarian. For association rules with negation the number of rules is unmanageable, thus not feasible in practice. This phenomenon can be easily explained as follows: in an average transaction, only a small fraction of items is present. At the same time, almost all possible items are not present in every transaction. Therefore, each transaction supports a huge number of patterns containing negation. Post-processing of association rules with negation and pruning coincidental rules is also a difficult and tedious task.

In our opinion, the main problem of previously proposed solutions is the complexity and the size of models, which effectively hinder the usefulness and practical applicability of these sophisticated models. In this paper we introduce a novel concept of dissociation rules. Our goal is to allow users to find negatively associated sets of items while keeping the number of discovered patterns low. We concentrate on such formulation of the problem, which results in a compact and usable set of patterns. By simplifying the model we sacrifice the abundance of discovered patterns for the simplicity and intelligibility of the result, making our model attractive for end users. Our main contribution includes the introduction of dissociation rules and the development of the DI-Apriori algorithm for mining dissociation rules. We conduct several experiments on synthetic datasets that compare our algorithm to a straightforward naive approach. The results of the experimental evaluation prove the feasibility of the presented proposal.

The paper is organized as follows. In Sect. 2 we review the related work on the subject. We present basic definitions used throughout the paper in Sect. 3. The naive approach and the DI-Apriori family of algorithms are presented in Sect. 4.

Section 5 contains the results of the experimental evaluation of the proposed solution. The paper concludes in Sect. 6 with the future work agenda.

2 Related Work

The first proposal to discover strong, exact, and approximate rules over the tuples contained in a relational table was formulated in [12]. The notion of association rule mining was introduced in [3]. In [4] the authors introduced the Apriori algorithm that quickly became the seed for many other algorithms for discovering frequent itemsets. The idea of mining "negative" information was first presented in [5] where the authors introduce the concept of excluding associations. They present a versatile method for finding associations of the form $A \wedge B \wedge \neg C \Rightarrow D$, where $A \wedge B \Rightarrow D$ does not hold due to insufficient confidence. Such a rule represents the fact that "A and B imply D when C does not occur". Their solution is to transform the database into a trie structure and extract both positive association rules and excluding association rules directly from the trie.

An algorithm for discovering strong negative association rules using taxonomy of domain knowledge was presented in [13]. This fundamental work introduced the concept of the interestingness of a rule measured in terms of the unexpectedness of the rule. A rule is unexpected if its support significantly deviates from the expected support. The authors propose to use the taxonomy of items along with the uniformity assumption to discover itemsets with support significantly lower than the expected support computed from the taxonomy. Another method for mining both positive and negative association rules is presented in [14]. The authors define a new measure for rule importance that combines support, confidence, and interestingness of a rule. Using this measure the authors introduce novel concepts of frequent and infrequent itemsets of potential interest that are used for mining positive and negative association rules.

An interesting algorithm for mining both positive and negative association rules is presented in [6]. The authors constrain themselves to finding confined negative association rules of the form $\neg X \Rightarrow Y$, $\neg X \Rightarrow \neg Y$, or $X \Rightarrow \neg Y$, where the entire antecedent or consequent is a conjunction of only negated or a conjunction of only non-negated terms. These rules are a subset of the generalized negative association rules, for which its antecedents or consequents can be expressed as a conjunction of negated or non-negated terms. The authors acknowledge that their approach is not general enough to capture all types of negative rules. However, limiting the algorithm to the discovery of confined negative association rules only allows the authors to develop an efficient method based on the correlation coefficient analysis.

The problem of mining generalized negative association rules has been attacked in [9]. Itemsets are divided into derivable and non-derivable based on the existence of certain rules (functional dependencies) in the dataset. The authors present an efficient method of concise representation of a huge number of patterns with negation using negative border and rule generators. Furthermore, an efficient algorithm for mining rules with negation is presented that uses

variations of candidate itemsets and error counts of rules. Finally, [8] discusses inverse Apriori-like method for mining sporadic rules, which are rules with very low support and high confidence.

Somehow related to the negative association rule mining is the problem of discovering unexpected patterns [11]. The authors propose to use prior background knowledge acquired from domain experts to serve as a set of expectations and beliefs about the domain. They combine this prior knowledge with association rule mining algorithm to discover patterns that contradict expert expectations. Similar research on exception rules was conducted in [7, 10]. Exception rules, sometimes also referred to as surprising patterns, represent an unexpected deviation from a well-established fact and allow negated terms to appear in patterns.

3 Basic Definitions

Let $L = \{l_1, l_2, \dots, l_n\}$ be a set of literals called *items*. Let D be a database of variable-length transactions, and $\forall t_i \in D : t_i \subseteq L$. A transaction t_i supports an item x if $x \in t_i$. A transaction t_i supports an itemset X if $\forall x \in X : x \in t_i$. The *support* of an itemset X , denoted as $support_D(X)$, is the ratio of the number of transactions in D that support X to the total number of transactions in D . Given two itemsets $X, Y \subset L$, the support of the itemset $X \cup Y$ is called the *join* of X and Y . An itemset containing k items is called a *k-itemset*. An itemset with the support higher than the user-defined threshold *minsup* is called a *frequent itemset*. Let L_D denote the set of all frequent itemsets discovered in the database D . The *negative border* of the collection of frequent itemsets, denoted as $Bd^-(L_D)$, consists of minimal itemsets not contained in the collection of frequent itemsets. Formally, $Bd^-(L_D) = \{X : X \notin L_D \wedge \forall Y \subset X, Y \in L_D\}$.

Given user-defined thresholds of minimum support and maximum join, denoted as *minsup* and *maxjoin*, respectively, where *minsup* > *maxjoin*. An itemset Z is a *dissociation itemset*, if $support_D(Z) \leq maxjoin$ and Z can be divided into disjoint itemsets X, Y , such that $X \cup Y = Z$, $support_D(X) \geq minsup$, and $support_D(Y) \geq minsup$. Dissociation itemsets are used to generate dissociation rules. A dissociation rule is an expression of the form $X \not\Rightarrow Y$, where $X \subset L$, $Y \subset L$, and $X \cap Y = \emptyset$. Furthermore, $support_D(X \cup Y) \leq maxjoin$, $support_D(X) \geq minsup$, and $support_D(Y) \geq minsup$. X is called the antecedent of the rule and Y is called the consequent of the rule. A dissociation rule $X \not\Rightarrow Y$ represents the fact that, although items contained in X and items contained in Y often occur together when X and Y are considered separately, items contained in $X \cup Y$ occur together very rarely. A dissociation rule $X \not\Rightarrow Y$ is *minimal*, if $\nexists X' \subseteq X, Y' \subseteq Y$ such that $X' \not\Rightarrow Y'$ is a valid dissociation rule.

Three statistical measures are used to describe the statistical significance and strength of the rule. The *support* of the rule $X \not\Rightarrow Y$ is the smaller ratio of the number of transactions that support either the antecedent or the consequent of the rule to the total number of transactions.

$$support_D(X \not\Rightarrow Y) = \min\{support_D(X), support_D(Y)\}$$

We decide to redefine the notion of rule support purposely. The support of the rule is used mainly for post-processing of discovered rules to select rules of interest. In this case, users are likely to be interested in selecting rules that pertain to statistically significant itemsets contained in either the antecedent or the consequent of the rule. The *join* of the rule is used to measure the quality of the rule expressed as the rarity of the rule,

$$join_D(X \not\Rightarrow Y) = support_D(X \cup Y)$$

Again, we choose to use the term *join* for the measure known as the support of the rule in traditional association rule mining. We decide to do so in order to avoid confusion, as the relative importance of a rule increases with the increase of the traditional support of the rule, whereas in the case of dissociation rules the most important rules are the ones with very low values of the join measure.

The confidence of the rule $X \not\Rightarrow Y$ is the ratio of the number of transactions that support the antecedent and do not support the consequent of the rule to the number of transactions that support the antecedent of the rule.

$$\begin{aligned} confidence_D(X \not\Rightarrow Y) &= \frac{support_D(X) - support_D(X \cup Y)}{support_D(X)} = \\ &= 1 - \frac{join_D(X \not\Rightarrow Y)}{support_D(X)} \end{aligned}$$

The problem of discovering dissociation rules can be formulated as follows. Given a database D and thresholds of minimum support, confidence, and maximum join, called *minsup*, *minconf*, and *maxjoin*, respectively. Find all dissociation rules valid in the database D with respect to the above mentioned thresholds. The thresholds are used in the following way. The *minsup* is used to select statistically significant itemsets for antecedents and consequent of generated rules. The *maxjoin* threshold provides an upper limit of how often the elements constituting the antecedent and the consequent of the rule are allowed to appear together in the database D . Finally, the *minconf* threshold is used only for the post-processing of rules and selecting the strongest rules. Note that given the values of *minsup* and *maxjoin*, the confidence of each generated dissociation rule has a lower bound $\underline{confidence}_D = (1 - maxjoin/minsup)$.

4 Algorithm

The generation of dissociation rules is based on the following lemmas.

Lemma 1. *Let L_D denote the set of all frequent itemsets discovered in the database D . If $X \not\Rightarrow Y$ is a valid dissociation rule, then $(X \cup Y) \notin L_D$.*

Lemma 1 is trivial. $X \not\Rightarrow Y$ implies that $support_D(X \cup Y) \leq maxjoin \leq minsup$, from which follows that $(X \cup Y) \notin L_D$.

Lemma 2. *If $X \not\Rightarrow Y$ is a valid dissociation rule, then $\forall X' \supseteq X, Y' \supseteq Y$ such, that $X' \in L_D \wedge Y' \in L_D$, $X' \not\Rightarrow Y'$ is a valid dissociation rule.*

From the fact that $X' \supseteq X$ and $Y' \supseteq Y$ follows that $\text{support}_D(X' \cup Y') \leq \text{support}_D(X \cup Y)$. Because $\text{support}_D(X \cup Y) \leq \text{maxjoin}$ and both X' and Y' are frequent, $X' \not\Rightarrow Y'$ is a dissociation rule.

Lemma 3. *$\forall X, Y$ such, that $X \not\Rightarrow Y$ is a valid dissociation rule, there exists $Z \in Bd^-(L_D)$ such, that $(X \cup Y) \supseteq Z$.*

From the definition of the negative border follows that, for each set X , either X is frequent, or X belongs to the negative border, or one of its proper subsets belongs to the negative border. Since $X \not\Rightarrow Y$ is a valid dissociation rule, either $(X \cup Y) \in Bd^-(L_D)$ (and all its proper subsets are frequent), or $(X \cup Y) \notin Bd^-(L_D)$ and it has a proper subset in $Bd^-(L_D)$. Otherwise, $(X \cup Y)$ would have to be frequent and $X \not\Rightarrow Y$ would not be a valid dissociation rule. Lemma 3 is particularly important, because it allows to find all dissociation rules by exploring and extending the negative border of the collection of frequent itemsets.

Similarly to traditional association rule mining, the problem of mining dissociation rules can be divided into two subproblems. The first problem consists in discovering all dissociation itemsets, given thresholds of *minsup* and *maxjoin*. The second problem consists in using discovered dissociation itemsets to generate dissociation rules. The naive approach to generating dissociation rules is the following. First, all frequent itemsets are discovered using the Apriori algorithm [4]. Next, all possible pairs of frequent itemsets are joined to generate candidate dissociation itemsets. Candidate dissociation itemsets that are contained in L_D are pruned based on Lemma 1. Actual support counts of candidate dissociation itemsets are found during a full database scan. This approach is highly ineffective. The number of candidate dissociation itemsets can be large, especially for low values of *minsup* threshold. Pruning performed based on Lemma 1 does not eliminate many candidate dissociation itemsets and many candidates are unnecessarily verified. However, the advantage of the naive algorithm is exactly one database scan to determine all valid dissociation itemsets and dissociation rules.

In order to efficiently discover dissociation rules, we propose the following procedure. We conclude from Lemma 2 that it is sufficient to discover only minimal dissociation rules. All remaining dissociation rules can be generated by extending antecedents and consequents of minimal dissociation rules with frequent supersets. Therefore, we reduce the problem of mining dissociation rules to the problem of mining minimal dissociation rules. We use Lemma 3 to limit the search space of candidate dissociation itemsets to supersets of sets contained in the negative border of the collection of frequent itemsets. Indeed, each set contained in the negative border $Bd^-(L_D)$ is either a candidate dissociation itemset, or is the seed set for a candidate dissociation itemset. Let us assume that the set $\{m, n, o\}$ is contained in the negative border $Bd^-(L_D)$. If $\text{support}_D(\{m, n, o\}) \geq \text{maxjoin}$, then $\{m, n, o\}$ is extended with all frequent 1-itemsets $\{p_i\}$, such that $\{p_i\} \cup \{m, n, o\}$ can be divided into two disjoint frequent

itemsets. Let L_D^1 denote the set of all frequent 1-itemsets. Let C_{\nrightarrow} denote the set of pairs of frequent itemsets that are candidates for joining into a dissociation itemset, and let D_{\nrightarrow} denote the set of pairs of frequent itemsets that form valid dissociation itemsets. The outline of the DI-Apriori algorithm is presented in Figure 1.

Require: L_D , the collection of all frequent itemsets
Require: L_D^1 , the collection of all frequent 1-itemsets

- 1: $D_{\nrightarrow} = \{(X, Y) : \{X \cup Y\} \in Bd^-(L_D) \wedge (X, Y).support \leq maxjoin\}$
- 2: $C_{\nrightarrow} = \{(X, Y) : \{X \cup Y\} \in Bd^-(L_D) \wedge (X, Y) \notin D_{\nrightarrow}\}$
- 3: **while** C_{\nrightarrow} grows **do**
- 4: **for all** $(X, Y) \in C_{\nrightarrow}$ **do**
- 5: **for all** $l \in L_D^1$ **do**
- 6: **if** $\{X \cup l\} \in L_D \wedge \{Y \cup l\} \in L_D \wedge \{X \cup Y \cup l\} \notin L_D$ **then**
- 7: $C_{\nrightarrow} = C_{\nrightarrow} \cup \{X \cup Y \cup l\}$
- 8: **end if**
- 9: **end for**
- 10: **end for**
- 11: **for all** $(X, Y) \in C_{\nrightarrow}$ **do**
- 12: compute $support_D(X \cup Y)$
- 13: **end for**
- 14: **end while**
- 15: $D_{\nrightarrow} = \{(X, Y) \in C_{\nrightarrow} : (X, Y).support \leq maxjoin\}$
- 16: **for all** $(X, Y) \in D_{\nrightarrow}$ **do**
- 17: **for all** $X' \in L_D : X' \supseteq X, Y' \in L_D : Y' \supseteq Y$ **do**
- 18: $D_{\nrightarrow} = D_{\nrightarrow} \cup \{(X', Y')\}$
- 19: **end for**
- 20: **end for**
- 21: **for all** $(X', Y') \in D_{\nrightarrow}$ for which the support is unknown **do**
- 22: compute $support_D(X' \cup Y')$
- 23: **end for**
- 24: **for all** $(X, Y) \in D_{\nrightarrow}$ **do**
- 25: **if** $1 - \frac{support_D(X \cup Y)}{support_D(X)} \geq minconf$ **then**
- 26: output $X \nrightarrow Y$
- 27: **else if** $1 - \frac{support_D(X \cup Y)}{support_D(Y)} \geq minconf$ **then**
- 28: output $Y \nrightarrow X$
- 29: **end if**
- 30: **end for**

Fig. 1. DI-Apriori

The DI-Apriori algorithm proceeds as follows. First, the negative border is examined and all itemsets with support less than *maxjoin* are added to the set of valid dissociation itemsets. The remaining itemsets in the negative border form the seed set of candidate dissociation itemsets. While the collection of candidate dissociation itemsets grows, we repeat the following steps. Each candidate dissociation itemset is extended with frequent 1-itemsets that allow

to split the candidate dissociation itemset into a frequent antecedent and consequent. All candidates are verified during a single database pass and their support counts are determined. Candidate dissociation itemsets with support lower than *maxjoin* are added to the set of valid dissociation itemsets. Discovered minimal dissociation itemsets are used to generate the remaining dissociation itemsets by replacing antecedents and consequents by their frequent supersets (lines 16–20). Also, the join measure for newly created dissociation itemsets is computed (lines 21–23). Finally, all dissociation itemsets are used to produce dissociation rules with respect to the provided *minconf* threshold.

Our implementation of the DI-Apriori algorithm uses a specialized physical data structure, the DI-tree. DI-tree is a lattice of dissociation itemsets, where each node of the lattice corresponds to a single dissociation itemset (either candidate or valid). DI-tree structure is optimized for fast lookup of dissociation itemsets and their components. On the physical level we have devised two modifications to the original method of the DI-tree traversal (these modifications are dubbed DI*-Apriori and DI⁻-Apriori, we do not describe them in detail due to the lack of space). The main advantage of DI-Apriori is the fact that the number of generated candidate dissociation itemsets is significantly smaller than when using the naive approach. Table 1 summarizes the number of frequent and candidate itemsets processed during the invocation of the naive algorithm in comparison with the number of itemsets tested by the DI-Apriori. The drawback of DI-Apriori is the fact that several database scans are required to compute the supports of dissociation itemsets. We present the results of the experimental evaluation of the proposed algorithm in the next section.

Table 1. Number of itemsets processed by Basic Apriori vs. DI-Apriori

<i>minsup</i>	<i>maxjoin</i>	Basic Apriori		DI-Apriori
		frequent itemsets	candidate itemsets	
5%	1%	83	396	264
4%	1%	214	2496	1494
3%	1%	655	16848	4971

5 Experimental Results

All experiments have been conducted on synthetic datasets generated using the generator from IBM’s Quest Project [2]. Experiments presented in this section use a dataset consisting of 20 000 transactions with an average size of 10 items, the *minsup* threshold is set to 5%, the number of patterns built into the dataset is 300 with an average size of 4 items. The *maxjoin* threshold is set to 3% if not stated otherwise. Figure 2 presents the number of dissociation rules and the number of frequent itemsets discovered when varying the *minsup* threshold.

In this experiment the *maxjoin* threshold is always kept 4% below the *minsup* threshold. We can see a strong correspondence between the number of dissociation rules and frequent itemsets (note the logarithmic scale on the y-axis). The execution time of algorithms when varying the average length of transactions is depicted in Figure 3. We do not observe any significant differences in the execution times of variations of DI-Apriori. Furthermore, all variations of DI-Apriori outperform the naive algorithm (again, note the logarithmic scale on the y-axis). Figure 4 shows the scaling capabilities of DI-Apriori. We are glad to notice that the algorithm scales almost linearly with the size of the database. Finally, Figure 5 presents the execution time relative to the difference between *minsup* and *maxjoin* thresholds. Obviously, the execution time of the naive algorithm does not depend on this parameter. The DI-Apriori algorithms perform slightly better for larger gaps between *minsup* and *maxjoin* thresholds, but the difference in execution times is not significant. We attribute this behavior to the fact that for larger gaps between *minsup* and *maxjoin* thresholds, although the number of dissociation itemsets drops, the number of intermediate itemsets (itemsets that are neither frequent nor rare) increases, thus keeping the size of the DI-tree structure approximately constant.

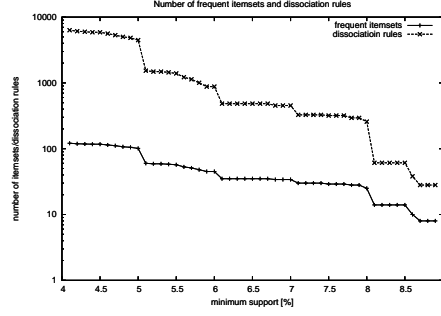


Fig. 2. Number of dissociation rules

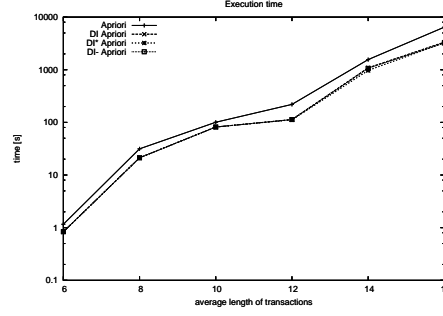


Fig. 3. Execution time

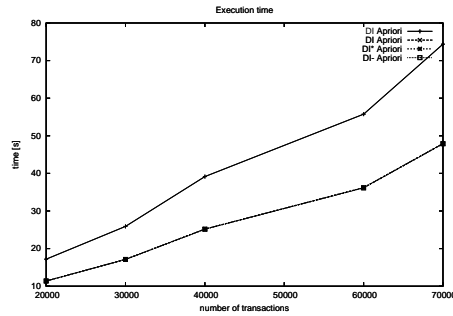


Fig. 4. Execution time

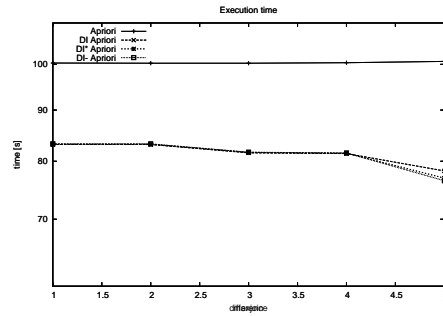


Fig. 5. Execution time

6 Conclusions

This paper initiates the research on dissociation rules. It is a simple model that successfully captures the "negative knowledge" hidden in the data, while keeping the number of discovered patterns low. Main advantages of the proposal are the simplicity, practical feasibility, and usability of the model. Our future work agenda includes further investigation of the properties of the model. One of the most urgent research directions is an experimental comparison of dissociation rules with other types of "negative" association rule models presented in Sect. 2. We are also eager to see how dissociation rules behave on real-world market basket datasets. Finally, we plan to refine the algorithm to scale to very large databases and we intend to develop concise and compact representations for collections of discovered dissociation rules.

References

1. *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, 1996.
2. R. Agrawal, M. J. Carey, and C. F. et al. Quest: A project on database mining. In *Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data, Minneapolis, Minnesota, May 24-27, 1994*, page 514. ACM Press, 1994.
3. R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In *1993 ACM SIGMOD, Washington, D.C., May 26-28*, pages 207–216. ACM Press, 1993.
4. R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *VLDB 1994, September 12-15, Santiago de Chile*, pages 487–499.
5. A. Amir, R. Feldman, and R. Kashi. A new versatile method for association generation. In *Princ. of Data Mining and Knowledge Disc.*, pages 221–231, 1997.
6. M.-L. Antonie and O. R. Zaiane. Mining positive and negative association rules: An approach for confined rules. Technical Report TR04-07, Department of Computing Science, University of Alberta, 2004.
7. F. Hussain, H. Liu, E. Suzuki, and H. Lu. Exception rule mining with a relative interestingness measure. In *PADKK 2000, Kyoto, Japan, April 18-20*, pages 86–97.
8. Y. S. Koh and N. Rountree. Finding sporadic rules using apriori-inverse. In *PAKDD 2005, Hanoi, Vietnam, May 18-20*, pages 97–106, 2005.
9. M. Kryszkiewicz and K. Cichoń. Support oriented discovery of generalized disjunction-free representation of frequent patterns with negation. In *PAKDD 2005, Hanoi, Vietnam, May 18-20*, pages 672–682, 2005.
10. H. Liu, H. Lu, L. Feng, and F. Hussain. Efficient search of reliable exceptions. In *PAKDD 1999, Beijing, China, April 26-28*, volume 1574, pages 194–203, 1999.
11. B. Padmanabhan and A. Tuzhilin. A belief-driven method for discovering unexpected patterns. In *KDD 1998, August 27-31, New York City, New York, USA*, pages 94–100. AAAI Press, 1998.
12. G. Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. In *Knowledge Discovery in Databases*, pages 229–248. AAAI/MIT Press, 1991.
13. A. Savasere, E. Omiecinski, and S. B. Navathe. Mining for strong negative associations in a large database of customer transactions. In *ICDE 1998, February 23-27, Orlando, Florida, USA*, pages 494–502. IEEE Computer Society, 1998.
14. X. Wu, C. Zhang, and S. Zhang. Efficient mining of both positive and negative association rules. *ACM Transactions on Information Systems*, 22(3):381–405, 2004.