USER-DRIVEN ASSOCIATION RULE MINING USING A LOCAL ALGORITHM

Claudia Marinica[1], Andrei Olaru[1,2] and Fabrice Guillet[1]
[1]*LINA Laboratory, Ecole Polytechnique de l'Universite de Nantes*
rue Christian Pauc BP 50609 44306 Nantes Cedex 3, France
[2]*Departement of Computer Science, University Politehnica of Bucharest*
Splaiul Independentei 313 060042 Bucharest, Romania

Keywords: Association Rules, Mining Algorithms, User-Driven Mining, Rule Interestingness, Subjective Measures.

Abstract: One of the main issues in the process of Knowledge Discovery in Databases is the Mining of Association Rules. Although a great variety of pattern mining algorithms have been designed to this purpose, their main problems rely on in the large number of extracted rules, that need to be filtered in a post-processing step resulting in fewer but more interesting results. In this paper we suggest a new algorithm, that allows the user to explore the rules space locally and incrementally. The user interests and preferences are represented by means of the new proposed formalism - the Rule Schemas. The method has been successfully tested on the database provided by Nantes Habitat.

1 INTRODUCTION

Knowledge Discovery in Databases is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data (Fayyad et al., 1996). Association Rule Mining (Agrawal et al., 1993) is an important technique of data mining, an Association Rule being defined as an implication of the form $a \rightarrow b$, that shows that the presence of itemset a in a transaction implies, with a certain confidence c, the presence of itemset b in the same transaction.

Association rules may obtain valuable information from large databases, but the number of rules extracted by classic algorithms is so large that it is impossible for un user to find himself the interesting ones, knowing that most rules are not interesting, already known or with no consequence in action (Piatetsky-Shapiro and Matheus, 1994), (Silberschatz and Tuzhilin, 1995). Generally, the interestingness of rules depends on objective (statistical) measures as the support and the confidence. Complementary, subjective measures were proposed in order to extract those interesting rules as compared to user background knowledge and user expectations. The two main subjective measures of interest are unexpectedness – rules surprising to the user – and actionability – rules helping the user to take actions (Silberschatz and Tuzhilin, 1995).

Although, usually, the reduction of the number of rules is done in the post-processing phase. Thus, the entire set of association rules is mined, most of them being, after all, considered as not interesting and eliminated. As a consequence, the whole process become quite inefficient (Silberschatz and Tuzhilin, 1995).

This paper proposes to introduce post-mining principles into the mining step, focusing on interesting rules without the necessity of extracting all rules existing in the database. The user may explore the rule space incrementally, a small amount at each step (Blanchard et al., 2007), starting from his/her own beliefs and knowledge and discovering rules related to these goals: confirming rules, specialized rules, generalized rules or exception rules. At each step the user chooses the most relevant rules for further exploration. This approach is based on a novel, flexible and unitary specification language that we propose in order to represent user interests – the Rule Schema with the 4 Operations that can be applied on – and on a novel local mining algorithm that we designed, generating a local set of candidate rules – as all possible rules that may result from applying the Operations to the Rule Schemas. This way, global post-processing is avoided in favor of a local and focused rule explo-

Marinica C., Olaru A. and Guillet F. (2009).
USER-DRIVEN ASSOCIATION RULE MINING USING A LOCAL ALGORITHM.
In *Proceedings of the 11th International Conference on Enterprise Information Systems - Artificial Intelligence and Decision Support Systems*, pages 200-205
DOI: 10.5220/0002003002000205
Copyright © SciTePress

ration that however is globally valid.

The proposed method has been tested on a real-life database, provided courtesy of the Nantes Habitat[1] agency. Several interesting experiments have been carried out and relevant results have been obtained.

The paper is organized as follows. Section 2 presents the research domain and reviews related works. Section 3 describes the Rule Schema formalism with the proposed Operations that the user can perform on, and presents the local mining algorithm. Results are discussed in section 4. Finally, section 5 presents the conclusion.

2 RELATED WORK

2.1 Association Rule Mining

The association rule mining technique (Agrawal et al., 1993) is applied over databases described as $D = \{I, T\}$. Let $I = \{I = 1, I_2, ..., I_p\}$ be a set of attributes (called items) and let $T = \{t_1, t_2, \ldots t_n\}$ be the transaction set. Each transaction $t_i = \{I_1, I_2, \ldots, I_m\}$ is a set of items, such as $t_i \subset I$ and each set of items, X, is called itemset.

An *association rule* is an implication $X \to Y$, where X and Y are two itemsets and $X \cap Y = \emptyset$. This rule holds on T with the *confidence c* if $c\%$ of transactions in T that contain X, also contain Y. The rule has *support s* in transaction set T if $s\%$ of transactions contain $X \cup Y$.

2.2 Mining Algorithms

Over the last decade, a great variety of rule mining algorithms was proposed starting with classic algorithms, passing by condensed representation algorithms and, ending with, incomplete set algorithms and inference generation techniques (Ceglar and Roddick, 2006). Classic algorithms, classified in candidate generation and pattern growth algorithm, extract all valid itemsets from data. Apriori (Agrawal et al., 1996) is the basic pattern generation algorithm and an important base for many algorithms proposed since. This type of algorithms is described on two steps: identifying candidate itemsets and validating the candidates. However, it is difficult to modify the pruning strategy in order not to extract all patterns, but only the ones that might be interesting to the user. In fact, most of the usual pattern-mining algorithms extract all patterns that have a minimum support. Moreover, most algorithms extract patterns, and not rules, association rules being built by using the mined itemsets.

Pattern growth algorithms, extended by several approaches, were introduced for the first time with the fundamental algorithm FP-Growth (Pei and Han, 2000). This algorithm simplifies the process of itemset generation creating complex hyperstructures: one structure describing the items with their frequency and the other being represented by the frequent pattern tree. Nevertheless, this technique is not convenient for big databases, due to memory problems.

Condensed representation algorithms and incomplete set of algorithms generate a subset of valid itemsets from which all itemsets can be derived, but this idea is not related to our work.

Closer to our paper ideas, Padmanabhan and Tuzilin proposed the inference generation reduction incorporating user knowledge in the mining algorithm (Padmanabhan and Tuzhilin, 1998). Thus, starting from a user belief and applying an Apriori-like local algorithm, a set of rules satisfying several constraints is generated.

2.3 Subjective Measures

Subjective measures and the integration of user beliefs were first discussed in the paper of Silberschatz and Tuzhilin (Silberschatz and Tuzhilin, 1995), in which the most important two subjective interestingness measures are introduced: unexpectedness and actionability. Three cases of unexpectedness are generally considered, comparing discovered rules and previous knowledge: unexpected condition, unexpected conclusion, and both condition and conclusion unexpected (Liu et al., 1999).

The concept of rule templates is introduced by Klemettinen (Klemettinen et al., 1994), as items in two lists: inclusion – rules that are interesting, and restriction – rules that are not interesting. However, all rules must be extracted and the search is not done locally. Moreover, the formalism is very simple and not too flexible. A logical representation and comparison for user beliefs has been suggested (Padmanabhan and Tuzhilin, 1998), but this approach is fairly limited. In (Li, 2006) the author proposed a framework for the discovery of a family of optimal rule sets for a range of interestingness metrics.

Liu presented (Liu et al., 1997) an interesting representation of user beliefs. It contains three levels of specification: General Impressions (GI), Reasonably Precise Concepts (RPC) and Precise Knowledge (PK). All three formalisms use items in a tax-

[1]We would like to thank Nantes Habitat, the Public Housing Unit in Nantes, France, and especially to Ms. Cristelle Le Bouter for supporting this work.

onomy, therefore allowing only for *is-a* relations between items. Moreover, using three different levels of specification might be difficult to use, if the user wants to combine their features.

A very early paper proposed the integration of special structures for the representation of domain knowledge and constraints (Anand et al., 1995): *Hierarchical Generalization Trees (HG-Trees)*, *Attribute Relationship Rules (AR-rules)* and *Environment Based Constraints (EBC)*. Using item taxonomies was suggested in (Srikant and Agrawal, 1995), but, however, taxonomies are limited to *is-a* relations. There are many advantages in using ontologies instead (Phillips and Buchanan, 2001).

Nevertheless, the number of rules extracted rests by these methods too large making impossible the post-analysis task, and an interactive process could help the user to find small sets of information.

3 USER-DRIVEN EXPLORATION OF THE RULE SPACE

3.1 Rule Schema formalism

Interesting rules are in a certain relation – confirmation or contradiction – with the current beliefs of the user. We propose a new formalism – *Rule Schemas* – in order to express user beliefs and expectations about the associations in the database.

Definition 1. A Rule Schema is represented as follows:

$$rs(Condition \rightarrow Conclusion \ \ [General]) \qquad [s\% \ c\%]$$

where *Condition* and the *Conclusion* contain the items that the user believes to be present in the antecedent and, respectively, in the consequent of the rule. The *General* part contains the items that the user is not sure in which of the two other parts to place.

The three parts – *Condition*, *Conclusion* and *General* – are all Expressions of the form $Expr = \{Expr\} \mid [Expr] \mid Expr? \mid Item$ – a disjunction or a conjunction of Expressions or an optional Expression (the items contained might not be present in the rule). The Rule Schema also contains optional constraints of support and confidence.

This formalism is based on the specification language proposed in (Liu et al., 1999), but it improves it by completely covering the three levels of specification presented in the General Impressions formalism (Liu et al., 1999). If the *Condition* and *Conclusion* are used, the Schema is more like a Reasonably Precise Concept or Precise Knowledge. If the *General* part is used, the Schema is more like a General Impression. The improvement is that a Rule Schema may use all three parts simultaneously.

3.2 Operations on Rule Schemas

Extending previous works (Blanchard et al., 2007), we propose 4 Operations that allow the user to explore the rule space starting from his/her beliefs and knowledge.

Confirmation is the simplest operation that we propose. It filters all rules that contain the items in *Condition* and in *Conclusion* in the antecedent, and respectively, in the consequent, and the items in the *General* part in any of the two sides of the implication. The items in the *General* part may be split in any possible ways between the antecedent and the consequent.

More formally, the searched rules are of the form: $Condition \cup Subset \ \rightarrow Conclusion \cup (General - Subset)$, for all $Subset \subseteq General$

Example. The Rule Schema $rs([A] \rightarrow [B] \ \ [C,D])$ is confirmed by any of the four following rules:

$A,C,D \rightarrow B$
$A,C \rightarrow B,D$
$A,D \rightarrow B,C$
$A \rightarrow B,C,D$

***k*-Specialization**, based on (Bayardo et al., 1999), allows the user to find the rules that have a more particular condition and the same conclusion, and which improve the confidence of the initial rule. That is, a specialization of $a \rightarrow b\ [s1\ c1]$ is $a,c \rightarrow b\ [s2\ c2]$, if $c2 > c1$.

k-Specialization is not performed directly on the database, but on the rules resulting from the Confirmation of the initial Rule Schema. For the rules of the form $Condition \rightarrow Conclusion$ obtained after the Confirmation operation, results of the *k*-Specialization are of the form:

$Condition \cup Set \ \rightarrow Conclusion$, for all $|Set| = k$ and $Set \subseteq (I - (Condition \cup Conclusion))$, with I the full itemset.

Example. For a Rule Schema $rs([A] \rightarrow [B])$ and $I = \{A,B,C,D\}$ the output of the *1*-Specialization operation may be:

$A \rightarrow B\ \ [s1\ c1]$
$A,C \rightarrow B\ \ [s2\ c2]$
$A,D \rightarrow B\ \ [s3\ c3]$

In the example above, it is required that the confidence measure satisfies $c2 \geq c1$ and $c3 \geq c1$. Obviously, for support, $s2 \leq s1$ and $s3 \leq s1$.

***k*-Generalization** is the opposite of *k*-Specialization. This operation finds the rules that have a more general condition implying the same conclusion. Support is expected to be higher and confidence slightly lower. Searched rules by k-Generalization operation are of the form:

$Condition - Set \rightarrow Conclusion$, for all $|Set| = k$ and $Set \subseteq Condition$.

***k*-Exception** is an important operation, as it finds rules with an unexpected conclusion, in the context of a more specialized condition. That is, for rules of the form $a \rightarrow B$ $[s1\ c1]$ exceptions are of the form $a,\ c \rightarrow \neg B$ $[s2\ c2]$. In (Duval et al., 2007) an exception is considered valuable knowledge if, knowing that the confidence of $c \rightarrow \neg B$ is $c3$, then $c2 \geq c1$, and $c3$ must be fairly low, as it must not be c alone, but the association with a that leads to $\neg B$. k-Exception operation is the k-Specialization of the rules with negated conclusion.

3.3 Local Mining Algorithm

In our approach, the search for interesting rules becomes *local*: rules are searched in the neighbourhood of rules and associations that the user already knows, or that the user believes to be true, specified by means of the Rule Schemas.

Example. Suppose the user wants to find all *1*-Specialization rules of the Rule Schema $rs([A] \rightarrow [\{B,C\}][D])$ $[10\%, 60\%]$. That is, A leads to either B or C, and they are associated with D. Support and confidence must be over 10% and 60% respectively. The full item set is $I = \{A,B,C,D,E,F\}$. The algorithm works as follows:

- first, Expressions are expanded and the *General* part is split between condition and conclusion. Generated rules are then checked against the database and candidates with support lower than 10% are pruned:

  $[A] \rightarrow [B\ D]$ $[25\%\ 67\%]$
  $[A\ D] \rightarrow [B]$ $[25\%\ 44\%]$
  $[A] \rightarrow [C\ D]$ $[8\%\ 26\%]$ – pruned
  $[A\ D] \rightarrow [C]$ $[8\%\ 35\%]$ – pruned

- based on the previous result, *1*-Specialization operation is performed (new items from I are added to the condition). Candidates are checked and pruned if support or confidence are lower than the threshold specified in the Schema or if there is no improvement in confidence.

  $[A\ C] \rightarrow [B\ D]$ $[7\%\ 20\%]$ – pruned for support
  $[A\ E] \rightarrow [B\ D]$ $[17\%\ 62\%]$ – pruned: ancestor's confidence is not improved
  $[A\ F] \rightarrow [B\ D]$ $[20\%\ 83\%]$
  $[A\ D\ C] \rightarrow [B]$ $[7\%\ 35\%]$ – pruned for support
  $[A\ D\ E] \rightarrow [B]$ $[9\%\ 48\%]$ – pruned for support
  $[A\ D\ F] \rightarrow [B]$ $[14\%\ 66\%]$

- results are sorted according to confidence:

  $[A\ F] \rightarrow [B\ D]$ $[20\%\ 83\%]$
  $[A\ D\ F] \rightarrow [B]$ $[14\%\ 66\%]$

There are a number of advantages that this approach has, compared to the Apriori algorithm. They result in great part from the fact that the Rule Schemas are partially instantiated, so the search space is greatly reduced. Moreover, the more the user refines current Rule Schemas, the lower is the number of generated candidate rules.

Compared to Apriori, the number of passes through the databases is lower. Once all candidate rules are generated, only one pass through the database is necessary, to check the support of the candidates. For complexity reasons, in the case of multilevel operations one pass per specificity / generality level is necessary.

One important issue in the presented approach is the number of generated candidate rules. This depends on the operation and on the properties of particular Rule Schemas, as shown below.

For the operation of Confirmation on a Rule Schema $rs(X \rightarrow Y\ [Z])$ (where X,Y,Z are item sets), the number of generated rules is equal to the number of possibilities of splitting the Z set into two subsets: $2^{|Z|}$: for each subset S in Z, S is added to the condition and $Z - S$ to the conclusion. Usually, the number of items in Z will be fairly low.

In Specialization, all the rules of the form $X \cup S \rightarrow Y$ are generated, where $S \not\subset X \cup Y$ and $|S|$ the specificity level. Normally, the number of candidate rules would be $C^{|S|}_{|I|-|X \cup Y|}$ where I contains all items in the database. However, a more efficient implementation explores specialized rules one level at a time. Most level 1 candidates are pruned, so the second level will be based on much fewer rules, reducing the total number of generated candidate rules. The same approach may be used for Exception.

4 RESULTS AND DISCUSSION

We have developed an application that implements the algorithm described above and allows the management of Rule Schemas. The application was tested on a real-life database, provided by Nantes Habitat, a public office managing social accommodations in Nantes, France. Each year, 1500 Nantes Habitat custemers (out of a total of 50000) answer a questionnaire about the quality of their accommodation. In the database, there exists one attribute for each question

Question number and text	
Q8	Neighborhood is quiet
Q18	Cleanness of the floor
Q26	State of the entry hall
Q29	State of the floor
Q60	Building corresponds to expectations
Q65	Technical interventions
Q92	NH services are adequate to needs
Q97	Price
S_boi	Boissiere neighbourhood

Figure 1: Meaning of questions in the example.

– with values of 1 to 4 for the decreasing satisfaction level and one transaction for each questionnaire. A set of questions with their meanings is presented in Figure 1. To obtain binary attributes, items of the form question=answer are formed, for example the item Q35=1 represents an answer of "very satisfied" to the question number 35 – about the lighting of common spaces. Therefore there are a total of 624 items. With classical algorithms and imposing thresholds of 8% for support and 85% for confidence we extracted a total of 1.528.978 rules, very difficult to post-process, even using a tool.

2-Specialization of
$rs([Q97 = 4] \rightarrow [Q26 = 4]\ \ [2\%\ 95\%])$

The analyst is interested on what relates to the implication of the dissatisfaction about price (Question 97) on the dissatisfaction about the state of the entry hall (Question 26) – as a part of the common spaces in the building. The search starts with a Rule Schema containing unsatisfied answers (value 4) to two questions related to the problem: Q97 and Q26. Support must be reasonable (2%) and confidence must be high (95%).

An operation of 2-Specialization has the following output finding items that are related to the implication:

```
17 results.
[Q29=4, Q65=4, Q97=4] -> [Q26=4] [ 2.6% 97.5% ]
[Q29=4, Q32=4, Q97=4] -> [Q26=4] [ 2.4% 97.3% ]
[Q16=4, Q29=4, Q97=4] -> [Q26=4] [ 4.6% 97.1% ]
[Q29=4, Q60=1, Q97=4] -> [Q26=4] [ 2.2% 97.0% ]
[Q29=4, Q37=4, Q97=4] -> [Q26=4] [ 2.0% 96.7% ]
[Q18=4, Q31=4, Q97=4] -> [Q26=4] [ 2.0% 96.7% ]
...
```

The output shows a number of interesting relations: apart from dissatisfaction about the state and cleanness of common spaces (entry hall, building level, etc) and equipment (interphone), there is also an indication about the efficiency with which the technical requests are addressed (Q65). There is also an interesting rule, the forth one: $[Q29 = 4, Q60 = 1, Q97 = 4] -> [Q26 = 4][2.2\%\ 97.0\%]$ there is a strong implication between the state of the building (particularly the level – Q29), the price (Q97) and the respondent's expectations (Q60), in that the expectations actually correspond, although the price is considered too high.

2-Exception of
$rs([Q97 = 4] \rightarrow [Q26 = 4]\ \ [2\%\ 95\%])$

The analyst might also want to check if there are exceptions to the specified Rule Schema. With the support threshold lowered to 1% (exceptions are rare rules), a 2-Exception operation outputs:

```
15 results.
[S_boi, Q18=1, Q97=4] -> [Q26=1] [ 1.5% 100.0%]
[S_boi, Q29=1, Q97=4] -> [Q26=1] [ 1.5% 100.0%]
[S_boi, Q16=1, Q97=4] -> [Q26=1] [ 1.3% 100.0%]
[S_boi, Q47=1, Q97=4] -> [Q26=1] [ 1.2% 100.0%]
[S_boi, Q78=1, Q97=4] -> [Q26=1] [ 1.2% 100.0%]
[S_boi, Q92=1, Q97=4] -> [Q26=1] [ 1.0% 100.0%]
...
```

This result is very interesting. It shows that dissatisfaction in price can actually lead to a better opinion on the building, if it is connected, among with some other items, with a certain neighbourhood (Boissiere), which is also very calm (Q8). This is important to know, because, although clients are quite happy with the conditions in that neighbourhood, they are unhappy with the price.

2-Generalization of
$rs([S\_boi, Q18 = 1, Q97 = 4] \rightarrow [Q26 = 1])$

Considering the last discovery in 2-Exception, the analyst might want to look a bit more into the first rule, so he performs a 2-Generalization, with the following output:

```
3 results.
[S_boi] -> [Q26=1] [ 5.0% 88.1% ]
[Q97=4] -> [Q26=1] [ 22.5% 59.1% ]
[Q18=1] -> [Q26=1] [ 63.2% 76.2% ]
```

Indeed, the Boissiere neighbourhood has a great influence on the satisfaction about state of the hall (Q26).

Confirmation of
$rs([S\_boi] \rightarrow [Q26 = 1]\ \ [Q92])\ \ [1\%\ 60\%]$

Last, the analyst will want to investigate the relation of the first rule in the last result with the satisfaction about adequacy of Nantes Habitat services (Q92). The analyst performs a Confirmation operation on the Rule Schema.

There are 2 results:

```
[S_boi, Q92=1] -> [Q26=1]        [ 3.7% 88.7% ]
[S_boi]        -> [Q26=1, Q92=1] [ 3.7% 65.4% ]
```

It appears that the rules relate more to the satisfaction answer to Question 92 (Q92$\bar{1}$) and it is usually perceived by the customers in this neighbourhood that a good adequacy of the agency's services leads, among others, to a good state of the building.

This is only an example of actions that an analyst may perform. Using Rule Schemas is easy and allows focus on the most interesting rules. Also, the operations are executed very quickly. For comparison, using apriori and rule filtering would require extracting all the rules in the database and filtering all of them every time (1.528.978 rules for 8% of support and 85% of confidence), in order to obtain the desired results. Moreover, if the database is more dynamic, the rule extraction must be done again, which can take a considerable amount of time.

5 CONCLUSIONS

In this paper we have presented a new solution for local association rule mining that integrates user beliefs and expectations. The solution has two important components. The Rule Schema formalism, based on the concepts introduced by Liu (Liu et al., 1999), helps the user focus the search for interesting rules, by means of a flexible and unitary manner of representation. The local mining algorithm that was developed does not extract all rules and then post-process them, but, instead, searches interesting rules in the vicinity of what the user believes or expects. This way, the user can explore the rule space in a local and incremental manner, global processing being avoided.

The proposed algorithm was tested on a real-life example, showing that the presented solution is valid and leads to good practical results.

REFERENCES

Agrawal, R., Imielinski, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. *ACM SIGMOD Record*, 22(2):207–216.

Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., and Verkamo, A. I. (1996). Fast discovery of association rules. *Advances in knowledge discovery and data mining*, 1:307–328.

Anand, S. S., Bell, D. A., and Hughes, J. G. (1995). The role of domain knowledge in data mining. *Proceedings of the fourth international conference on Information and knowledge management*, 1:37–43.

Bayardo, R. J., Bayardo, R. J., Agrawal, R., Agrawal, R., Gunopulos, D., and Gunopulos, D. (1999). Constraint-based rule mining in large, dense databases. *In Proceedings of the 15th International Conference on Data Engineering*, pages 188–197.

Blanchard, J., Guillet, F., and Briand, H. (2007). Interactive visual exploration of association rules with rule-focusing methodology. *Knowledge and Information Systems*, 13:43–75.

Ceglar, A. and Roddick, J. F. (2006). Association mining. *ACM Comput. Surv.*, 38(2):5.

Duval, B., Salleb, A., and Vrain, C. (2007). On the discovery of exception rules: A survey. *Quality Measures in Data Mining*, pages 77–98.

Fayyad, U. M., Piatetsky-Shapiro, G., and Smyth, P. (1996). From data mining to knowledge discovery: An overview. *Advances in Knowledge Discovery and Data Mining*, 1:1 –34.

Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H., and Verkamo, A. I. (1994). Finding interesting rules from large sets of discovered association rules. *Proceedings of the third international conference on Information and knowledge management*, pages 401–407.

Li, J. (2006). On optimal rule discovery. *IEEE Transactions on Knowledge and Data Engineering*, 18(4):460–471.

Liu, B., Hsu, W., and Chen, S. (1997). Using general impressions to analyze discovered classification rules. *Proc. 3rd Int. Conf. Knowledge Discovery & Data Mining*, 1:31–36.

Liu, B., Hsu, W., Wang, K., and Chen, S. (1999). Visually aided exploration of interesting association rules. *PAKDD '99: Proceedings of the Third Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining*, pages 380–389.

Padmanabhan, B. and Tuzhilin, A. (1998). A belief-driven method for discovering unexpected patterns. *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, 1:94–100.

Pei, J. and Han, J. (2000). Mining frequent patterns by pattern-growth: methodology and implications. *ACM SIGKDD Explorations*, 2:14–20.

Phillips, J. and Buchanan, B. G. (2001). Ontology-guided knowledge discovery in databases. *Proceedings of the international conference on Knowledge capture*, pages 123–130.

Piatetsky-Shapiro, G. and Matheus, C. J. (1994). The interestingness of deviations. *Proceedings of the AAAI-94 Workshop on Knowledge Discovery in Databases*, 1:25–36.

Silberschatz, A. and Tuzhilin, A. (1995). On subjective measures of interestingness in knowledge discovery. *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, pages 275–281.

Srikant, R. and Agrawal, R. (1995). Mining generalized association rules. *Future Generation Computer Systems*, 13:161–180.