To appear in the 12th International Conference on Data Engineering (ICDE'96), New Orleans, Louisiana, February/March 1996.

Knowledge Discovery from Telecommunication Network Alarm Databases

K. Hätönen M. Klemettinen H. Mannila
P. Ronkainen H. Toivonen

University of Helsinki, Department of Computer Science
P.O. Box 26, FIN-00014 Helsinki, Finland
e-mail: Heikki.Mannila@cs.Helsinki.FI

Abstract

A telecommunication network produces daily large amounts of alarm data. The data contains hidden valuable knowledge about the behavior of the network. This knowledge can be used in filtering redundant alarms, locating problems in the network, and possibly in predicting severe faults. We describe the TASA (Telecommunication Network Alarm Sequence Analyzer) system for discovering and browsing knowledge from large alarm databases.

The system is built on the basis of viewing knowledge discovery as an interactive and iterative process, containing data collection, pattern discovery, rule postprocessing, etc. The system uses a novel framework for locating frequently occurring episodes from sequential data.

The TASA system offers a variety of selection and ordering criteria for episodes, and supports iterative retrieval from the discovered knowledge. This means that a large part of the iterative nature of the KDD process can be replaced by iteration in the rule postprocessing stage. The user interface is based on dynamically generated HTML. The system is in experimental use, and the results are encouraging: some of the discovered knowledge is being integrated into the alarm handling software of telecommunication operators.

1 Introduction

Knowledge discovery in databases (KDD) has recently attracted a lot of interest from researchers and users of database systems; see [5, 13] for overviews. KDD combines methods and tools from machine learning, statistics, and databases. It can be loosely defined as the task of obtaining useful and interesting knowledge from large collections of data.

KDD is an iterative and interactive process [3, 4]. In the core of the KDD process are the algorithms for discovering different types of patterns (rules, trends, etc.) from data. However, in the whole problem of obtaining useful knowledge, the inference of patterns is only a small part. Among the other tasks are the following:

1. data collection and cleaning (what types of data can be used, how errors in the data are handled, what is to be done with missing data, etc.); identification of the necessary background knowledge;
2. choice of pattern discovery methods (what types of knowledge are to be discovered, parameter selection, etc.);
3. discovery of patterns (data mining);[1]
4. postprocessing the discovered knowledge (selection of truly interesting patterns, presentation of patterns, etc.);
5. putting the discovered knowledge into use.

In this paper we describe the TASA (Telecommunication Network Alarm Sequence Analyzer) system for discovering knowledge from telecommunication network alarm databases. The system incorporates components for two parts of the KDD process: pattern discovery (or data mining) and postprocessing.

The knowledge discovered in TASA is expressed in terms of rules. The system is based on a novel framework for locating frequently occurring episodes from sequential data. The algorithms we use are based on methods presented in [11], and they have a completeness property: they find from the data all rules having certain properties.

We use these algorithms to find effectively large sets of patterns from the data (typically thousands of rules). By finding a large collection of rules, a large part of the iterative nature of the KDD process can be replaced by iteration in the rule postprocessing stage.

The TASA system offers a variety of selection and ordering criteria, and supports iterative retrieval from the discovered knowledge. The users can manipulate the collection of discovered rules using selection and ordering operations, as well as more complex operations for including or excluding certain classes of rules.

[1] We use here the terminology of Fayyad et al. [4], where "data mining" refers to the pattern extraction part of the KDD process.

Figure 1: Home Page

Figure 2: Single alarm description

Any realistic KDD system produces a lot of information. This wealth of information has to be presented to the users in an accessible form. Our experience shows that the users want to be able to use several different types of views into the data. They want to see the discovered knowledge, but they also want to be able to see how that knowledge is actually supported by the original data. For the user interface system we have adopted the method used by Matheus et al. [12]: the output produced by the TASA system is in the form of HTML documents. This gives several immediate benefits, both in terms of added functionality and in terms of easy implementation.

The TASA system is in prototype use, and the results are encouraging: some of the discovered knowledge is being integrated into the alarm handling software of telecommunication operators.

The rest of this paper is organized as follows. Section 2 describes the data in the telecommunication network alarm databases. Section 3 points out what types of data cleaning operations have to be done to this data. In Section 4 we study alternative representations for knowledge discovered from network databases, discuss our choice of rule-based representations, and give the precise rule formalism. The algorithm for finding all rules of the formalism that hold in the data is given in Section 5, where we also present some empirical results. Section 6 includes our methodological contribution for the KDD process: the tools for postprocessing the set of discovered rules. Section 7 is a short conclusion.

The *Home Page* of a data set analyzed with the TASA system (Figure 1) contains a brief overview of the most descriptive parameters of the data as a whole. These parameters include, for instance, the time span of the data, number of alarms, average frequency of alarms, and so on. In addition, there are links to histograms that characterize the whole alarm data set as well as links to HTML pages that show the results of the analysis.

2 Telecommunication alarm databases

A telecommunication network can be viewed as consisting of a number of interconnected components: switches, exchanges, transmission equipment, etc. Each component in its turn contains several subcomponents. The number of components depends on the abstraction level used in viewing the system. A network operated by a local telephone company can be considered to contain 10–1000 components.

Each (sub)component and software module can produce *alarms*. They are messages describing some sort of abnormal situations; they do not necessarily indicate that there has been a problem in the network that is visible to the users. Even in a small local telecommunication network there can be thousands of different types of alarms. The number of produced alarms varies greatly, but typically there can be about 200–10000 alarms a day. Figure 2 is a TASA display of statistics of one alarm type recorded during a period of one month.

The operations and maintenance center (OMC) of the telecommunication network management system receives the alarms generated by the nodes in the network. It stores them in an alarm database, may filter them, but most importantly it displays the alarms to an operator, who then decides what has to be done with them.

Processing of the alarm flow is a difficult task for the following reasons.

- The size of the networks and the diversity of alarm types mean that there are a lot of different situations that can occur.
- The alarms occur in bursts, and hence there is only little time for operators to decide what to do with each alarm. However, when a lot of alarms occur within a short time the operators should intervene.
- The hardware and software used in telecommunication networks develop fast. As new nodes are added to the network or old ones are updated, the characteristics of the alarm sequences are changed. Thus the operators do not have time to learn what the appropriate response to each situation is.

To alleviate the problem of processing the alarms, *alarm filtering and correlation* [6] have been used to reduce the number of alarms that actually are shown to the operators and to raise the abstraction level of the information shown. Alarms are filtered at each level of the hierarchical network: a node sends forward only part of the alarms it receives from its subnodes. Alarm correlation means the combination and transformation of alarms so that several alarms are replaced by one alarm of better information value that is sent further.

Alarm filtering and correlation require stored knowledge about the processing of the alarm sequence. Such knowledge can in principle be obtained from the engineers who have designed the individual components, or who have extensive operating experience. However, this process can be tedious.

Filtering and correlation serve to diminish the number of alarms that the operator sees. However, the alarm handling software should ideally also be able to predict the behavior of the network, i.e., to be able to warn the operator beforehand of severe faults. Such faults typically arise or can be located from interconnected failures of components in the network. While prediction of severe faults is a difficult task, the economic benefits that would be obtained from such predictions would be significant.

The TASA system is being built in cooperation with a manufacturer of telecommunication equipment and three telephone operators (two metropolitan-wide fixed phone networks and a nationwide mobile network). The goal of the system is to locate regularities that help to process alarm sequences; these regularities can be used in filtering and transformation of the alarms, as well as in prediction of faults.

3 Collecting and cleaning the data

The first step in the KDD process is collecting and cleaning the data. In some domains this step can take up to 80 % of the total time needed. In our application the information is already collected to the alarm log, and the data is usually of high quality.

Some problems still remain. One is related to the time-dependent nature of the interesting events in the network. Each alarm can be abstractly viewed as a triple (c, a, t), where c is the component that sent the alarm, a is the alarm type, and t is the time when the alarm occurred.[2] The fields c and a are typically reliable, but the time field t is not: there can be differences of up to 3–5 minutes in the synchronization of the clocks. As our goal is to locate regularities that are intimately connected to the temporal aspects of alarms, such errors are problematic. A solution is to use preliminary data analysis and try to locate which components of the network are likely to have erroneous clocks.

The background knowledge needed in our KDD task consists mostly of knowledge about the network topology: how the components are connected and which subcomponents they contain. Additionally, the types of the network components form an inheritance hierarchy that is useful in the KDD process.

4 Types of knowledge to be discovered

From a sequence of alarms different types of knowledge can be discovered, for example neural networks, hazard models, or rule-based representations.

If the goal would be just to obtain good predictive performance, a neural network could be useful. There is impressive evidence on the wide applicability of neural networks. However, in the current application one important goal is the comprehensibility of the discovered knowledge: the telecommunication companies do not wish to install any “black boxes” into their systems. This rules out the simple-minded use of neural networks.

From the statistical point of view, an alarm sequence can be viewed as a marked point process, and each event (i.e., an alarm) can be considered as a failure of a component. Therefore the hazard rate based methods of analysis of failure-time data for the data can be used [7]. Combined with the Bayesian paradigm, the statistical machinery for these types of models is very powerful, and elaborate simulation mechanisms exist for their analysis (e.g., the EM and Gibbs methods, see [15]). We are, in fact, applying such techniques for small subproblems of the alarm analysis problem. However, these methods require a lot of human effort in the building of the statistical models, as well as enormous amounts of computational resources. Currently they are not feasible for the analysis of hundreds of different types of alarms and their potential relations.

[2] Of course, the actual alarms contain much more information (typically around 15 fields).

We use rule-based formalisms for knowledge discovery in the TASA system. The general form is as follows: "if a certain combination of alarms occurs within a time period, then an alarm of a given type will occur within a time period". The reasons for choosing this type of knowledge to be discovered are the following.

1. Comprehensibility: such rules are easy to understand for humans, and the operators currently handling the alarm sequences like to express their knowledge about the alarms using something close to these types of rules.

2. Characteristics of the application domain: such rules can be representations of simple small causal relationships within the domain. It can be argued that such type of knowledge is suitable for telecommunication networks (as compared to, say, a large causal network describing the whole structure of the domain all at once).

3. Existence of efficient algorithms: rules of the above form can be discovered efficiently.

Examples of the rule types discovered in the TASA system are:

1. If an alarm of type A occurs, then an alarm of type B occurs within 30 seconds with probability 0.8.

2. If alarms of types A and B occur within 5 seconds, then an alarm of type C occurs within 60 seconds with probability 0.7.

3. If an alarm of type A precedes an alarm of type B, and C precedes D, all within 15 seconds, then E will follow within 4 minutes with probability 0.6.

The general framework for expressing such rules is the following. An *alarm predicate* is an expression that can be evaluated from a single occurrence of an alarm, e.g., "alarm type is A", or "alarm type is A or B, and alarm severity class is 3". An *episode* is a pair $\alpha = (V, \leq)$, where V is a collection of alarm predicates, and $\leq$ is a partial order on V. Given a sequence of alarms S, an episode $\alpha = (V, \leq)$ occurs within S if there is a way of satisfying the alarm predicates in V using the alarms of S so that the partial order $\leq$ is respected. Figure 3 presents, as an example, the predicates and their partial order from the third episode of the above example.[3]

Figure 4 presents a snapshot of some simple episode rules. The first field on the row contains the left-hand side of the rule while the right-hand side is in the second field. After that there are the probability ("conf"), the frequency ("supp"), and the statistical significance ("sign") of the rule.

Next we describe how the number of occurrences of a given episode in a sequence is defined. Given an alarm sequence S and a width W for a time window, we form all subsequences of S consisting of alarms happening within an interval of W time units. That is, if the first event in the sequence happens at time T_0, we consider the subsequences $S_i, i \geq 0$, consisting of the alarms that take place in the half-open interval $[T_0+iv, T_0+iv+W)$, where v is the window movement. Note that subsequent windows are typically overlapping, if $v < W$. The *frequency* of an episode α is the number of windows S_i in which α occurs.

[3] TASA also searches for association rules between alarm predicates that are satisfied in one alarm. See [10] for the algorithm.

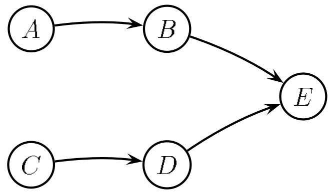

Figure 3: An episode presented as predicates and their partial order

A rule $r = (\alpha, e, W, W')$ consists of an episode $\alpha = (V, \leq)$, a distinguished predicate e of V, and two window widths W and W'. The interpretation of r is the following: the predicates in $V' = V \setminus \{e\}$ and the restriction of $\leq$ to V' define the left-hand side of the rule, and e is the right-hand side of the rule. The rule states: if alarms satisfying V' occur in the right order within W time units, then also e occurs in the location described by $\leq$, within W' time units (i.e., the whole episode α occurs.) The time interval W may not be larger than W'.

The *confidence* of a rule $r = (\alpha, e, W, W')$ is now t'/t, where t is the number of windows of width W in which $\alpha \setminus \{e\}$ occurs, and t' is the number of windows of width W' in which α occurs.

In the TASA system, the users specify the *class of interesting rules* by defining what types of partial orders are allowed in episodes, what types of alarm predicates are used, and what are the values for the two window widths W and W'. The most common types of partial orders used are (1) total orders, i.e., the predicates of each episode are totally ordered; such episodes are called *serial* or *ordered*, and (2) trivial partial orders, where no two disjoint predicates are ordered by $\leq$; such episodes are called *parallel* or *unordered*. It is not quite clear how useful truly partially ordered (i.e., non-serial and non-parallel) episodes are in the applications.

Typical alarm predicates involve the type and severity of the alarm, and possibly also the component of the network that sent the alarm. In our experiments, the application experts have preferred time windows ranging from 5 seconds to half an hour.

Figure 4: Simple episode rules

W (s)	Serial episodes		Parallel episodes	
	$\Sigma\lvert L_i\rvert$	time (s)	$\Sigma\lvert L_i\rvert$	time (s)
10	16	31	10	8
20	31	63	17	9
40	57	117	33	14
60	87	186	56	15
80	145	271	95	21
100	245	372	139	21
120	359	478	189	22

Table 1: Characteristics of episode discoveries with $c = 0.03$.

	Serial episodes		Parallel episodes	
c	$\Sigma\lvert L_i\rvert$	time (s)	$\Sigma\lvert L_i\rvert$	time (s)
0.1	0	7	0	5
0.05	1	12	1	5
0.008	30	62	19	14
0.004	60	100	40	15
0.002	150	407	93	22
0.001	357	490	185	22

Table 2: Characteristics of episode discoveries with $W = 60$ s.

5 Pattern extraction from alarms

In this section we present the algorithm that is used in TASA to discover all rules satisfying certain conditions. The algorithm is presented in more detail in [11] for a slightly simpler setting. Related ideas have been expressed in [1, 2, 10], where the idea of obtaining *all* rules is pursued.

The pattern extraction task in the TASA system can be formalized as follows. Given a sequence S of alarms, a set E of alarm predicates, a class $\mathcal{E}$ of episodes built from the predicates of E, a frequency threshold c, and window widths W and W', find the confidences of all rules (α, e, W, W) whose frequency is at least c. We first discuss how the frequently occurring episodes are computed, and then outline how rules are obtained from this information.

5.1 Computing frequent episodes

The problem we consider in this subsection is the following. Given S, E, $\mathcal{E}$, W, and c as above, find the frequencies of all episodes from $\mathcal{E}$ such that the frequency is at least c. The following algorithm will output all such episodes [11].

1. $C_1 := \{\{e\} \mid e \in E\}$;
2. $i := 1$;
3. **while** $C_i \neq \emptyset$ **do**
4. recognition: read the sequence S, and let L_i be the episodes of C_i that occur often enough with respect to W and c;
5. building: compute C_{i+1} to contain those episodes of $i+1$ alarm predicates whose all subepisodes occur often enough;
6. $i := i + 1$;
7. **od**;
8. **for** all i, output L_i;

The algorithm starts with simple (i.e., general) episodes and proceeds to larger episodes. The algorithm works iteratively: first a candidate collection C_i is generated, candidates are counted in the alarm sequence, those occurring often enough are saved in the collection L_i, and finally new candidates are generated again. Essential for the efficiency of the algorithm is the following observation. If an episode does not occur often enough, then its superepisodes — which are more specific — cannot occur often enough. Therefore, the candidate collection C_i of episodes is built (step 5) to contain only episodes whose all subepisodes occur often enough. See [11] for more details.

Characteristics of experiments on a real sequence of 73679 alarms of 287 types, covering a time period of 50 days, are presented in Tables 1 and 2. By $\Sigma|L_i|$ we denote the total number of frequent episodes. These experiments show that a large number of episodes can be found efficiently. The experiments have been run on a PC with 90 MHz Pentium processor and 16 MB main memory, under the Linux operating system. The alarm data resided in a flat text file. For more experimental results, see [11].

5.2 Finding rules from episodes

Once we know how often each episode from a class of episodes occurs in an alarm sequence, finding the confidences of rules is easy. For a potential rule (α, e, W, W') we have that the frequencies of the episode $\alpha \setminus \{e\}$ for time window W and of the episode α for time window W' exceed the threshold. I.e., the potentially holding rules can be generated efficiently from the frequent episodes. Then the task of computing rule confidences can be done simply in a single pass through the data.

6 Postprocessing of rule sets

6.1 Background and functionality

We now move to the next step in the KDD process, the handling of discovered rules. Our thesis is that this part of the process needs to be supported by powerful information retrieval tools, and that the use of such tools can decrease the amount of iteration in other parts of the process.

The aim of knowledge discovery in databases is to extract *interesting* knowledge from large collections of data. What is interesting varies from one situation to another. The interestingness criteria are in many KDD systems given as inputs to the pattern extraction process [9, 14]. If the users change their views, the data has to be analyzed anew. This can in the worst case require a lot of computational effort, and it can cause delays in the KDD process. In the TASA system we aim at a KDD process where going back to the actual data for a new pattern extraction process has to be done only seldom.

The algorithms presented in the previous section can discover thousands of rules reasonably quickly. Our methodological starting point is that interactive and iterative postprocessing of a large rule set can be useful in helping the users of a KDD system in locating the truly interesting information.

The TASA system offers a variety of selection and ordering criteria for rules and supports iterative retrieval from the discovered knowledge. We have identified the following types of operations the users want to do with the rule sets:

1. *Pruning*: removal of uninteresting rules or selection of interesting ones.

2. *Ordering* of rules according to some criteria.

3. *Grouping*: clustering of rules into groups of rules that have similar effects in the analyzed data sets.

Pruning and ordering need to be done according to the values of various attributes of the rules, e.g., types or severities of the alarms, or confidence, frequency, or statistical significance of the rules. Additionally, rules can be selected or removed by giving templates [8], i.e., simple regular expressions that describe the form of rules that are to be selected (inclusive template) or removed (restrictive template). This technique is surprisingly powerful. For example, some background knowledge about the connections in the network can fairly easily be taken into account by using templates that reject all rules where the nodes are not connected to each other. Figure 5 presents the selection form for pruning and ordering rules. On the form the user can define templates for both sides of rules, and give minimum and maximum values for rule confidence and frequency. Simple ordering criteria for the resulting rules can be given.

Figure 5: Fill-out form for rule pruning and selection

Grouping of rules is based on the following. In the data there are often various explanations for the occurrence of a particular alarm type, say B. The behavior of the alarm sequence can be better understood by clustering rules so that two rules with right-hand side B belong to the same cluster if they often explain B in the same situations [16].

6.2 User interface issues

The view of KDD as a process shows immediately the importance of the user interface of the KDD system. Any realistic KDD system produces a lot of information. This wealth of information has to be presented to the users in an accessible form. Our experience shows that in addition to looking and manipulating the discovered knowledge, the users want to be able to use several types of views into the data. They want to see the discovered knowledge, but they also want to be able to see how that knowledge is actually supported by the original data.

Figure 6: Distance histogram

Visualization of information is obviously an important part of KDD applications. For this the TASA system offers currently only simple facilities, implemented using an attached statistical package. As an example, consider the confidence of a rule. It is a crude measure of how well the rule manages to predict the occurrence of the right-hand side. A more complete picture of the interaction between the left and right-hand sides of a rule can be obtained by simply drawing a histogram showing the distance from each occurrence of the left-hand side to the nearest occurrence of the right-hand side. Such histograms are valuable guides for locating possible periodic relationships between the left and right-hand sides, as is demonstrated by Figure 6.

For the user interface system we have adopted the method used in [14]: we have based the implementation of the rule browsing system on the use of the HTML language and the browsers for HTML documents. This gives several immediate benefits, both in terms of added functionality and in terms of easy implementation. This architecture also means that it is extremely simple to port the rule browsing system to different platforms.

The Home Page of an analyzed data set (Figure 1) gives a brief overview of the data and contains links to the results of the analysis. Each rule set is an HTML document (e.g., Figure 4). Hypertext links lead from alarms in the rules to descriptions of single alarms (e.g., Figure 2). Selection, ordering, and grouping criteria for a rule set are given with an HTML fill-out form (Figure 5). The operations are applied on the rule set, and a new HTML document is created "on-the-fly". The user can move freely in the history path and reformulate the selections from any of the former pages. If the results are interesting, the user can save the on-screen page for further examination.

7 Concluding remarks

We have described the TASA system for analyzing sequences of alarms from telecommunication networks. Actually, the system can also be used for several other types of sequential data, e.g., in user interface studies: there the input consists of the key presses and mouse events, and the task is to find repeating groups of interconnected actions.

The first version of the TASA system has been tested with real alarm data from telecommunication network operators. The experiences are encouraging: rules discovered by the system have been deemed interesting, and telecommunication operators are integrating some of the discovered knowledge into their alarm handling systems. We are currently running more extensive tests.

References

[1] Rakesh Agrawal, Heikki Mannila, Ramakrishnan Srikant, Hannu Toivonen, and A. Inkeri Verkamo. Fast discovery of association rules. In Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*. AAAI Press, Menlo Park, CA, 1996. To appear.

[2] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the Twentieth International Conference on Very Large Data Bases (VLDB'94)*, September 1994.

[3] R. Brachman and T. Anand. Data archeology. In *Knowledge Discovery in Databases, Papers from the 1993 AAAI Workshop (KDD'93)*, July 1993.

[4] Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery: An overview. In Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*. AAAI Press, Menlo Park, CA, 1996. To appear.

[5] Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy, editors. *Advances in Knowledge Discovery and Data Mining*. AAAI Press, Menlo Park, CA, 1996. To appear.

[6] Gabriel Jakobson and Mark D. Weissman. Alarm correlation. *IEEE Network*, 7(6):52 – 59, November 1993.

[7] J. D. Kalbfleisch and R. L. Prentice. *The Statistical Analysis of Failure Time Data*. John Wiley Inc., New York, NY, 1980.

[8] Mika Klemettinen, Heikki Mannila, Pirjo Ronkainen, Hannu Toivonen, and A. Inkeri Verkamo. Finding interesting rules from large sets of discovered association rules. In *Proceedings of the Third International Conference on Information and Knowledge Management (CIKM'94)*, pages 401 – 407, New York, NY, November 1994. ACM.

[9] Willi Kloesgen. Efficient discovery of interesting statements in databases. *Journal of Intelligent Information Systems*, 4(1):53 – 69, 1995.

[10] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo. Efficient algorithms for discovering association rules. In Usama M. Fayyad and Ramasamy Uthurusamy, editors, *Knowledge Discovery in Databases, Papers from the 1994 AAAI Workshop (KDD'94)*, pages 181 – 192, Seattle, Washington, July 1994.

[11] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo. Discovering frequent episodes in sequences. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD'95)*, Montreal, Canada, August 1995. 210 – 215.

[12] Christopher J. Matheus, Gregory Piatetsky-Shapiro, and Dwight McNeill. Selecting and reporting what is interesting: An application of KEFIR to healthcare data. In Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*. AAAI Press, Menlo Park, CA, 1996. To appear.

[13] Gregory Piatetsky-Shapiro and William J. Frawley, editors. *Knowledge Discovery in Databases*. AAAI Press, Menlo Park, CA, 1991.

[14] Gregory Piatetsky-Shapiro and Christopher J. Matheus. The interestingness of deviations. In Usama M. Fayyad and Ramasamy Uthurusamy, editors, *Knowledge Discovery in Databases, Papers from the 1994 AAAI Workshop (KDD'94)*, pages 25 – 36, Seattle, Washington, July 1994.

[15] Martin A. Tanner. *Tools for Statistical Inference — Methods for the Exploration of Posterior Distributions and Likelihood Functions*. Springer-Verlag, Berlin, 1993.

[16] Hannu Toivonen, Mika Klemettinen, Pirjo Ronkainen, Kimmo Hätönen, and Heikki Mannila. Pruning and grouping of discovered association rules. In *Workshop Notes of the ECML-95 Workshop on Statistics, Machine Learning, and Knowledge Discovery in Databases*, pages 47 – 52, Heraklion, Crete, Greece, April 1995.