Discovery of Fraud Rules for Telecommunications - Challenges and Solutions

Saharon Rosset, Uzi Murad, Einat Neumann, Yizhak Idan, Gadi Pinkas
Amdocs (Israel) Ld.
8 Hapnina St.
Ra'anana 43000, Israel
Email : {saharonr, uzimu, einatn, yizhaki }@amdocs.com

ABSTRACT

Many fraud analysis systems have at their heart a rule-based engine for generating alerts about suspicious behaviors. The rules in the system are usually based on expert knowledge. Automatic rule discovery aims at using past examples of fraudulent and legitimate usage to find new patterns and rules to help distinguish between the two. Some aspects of the problem of finding rules suitable for fraud analysis make this problem unique. Among them are the following: the need to find rules combining both the properties of the customer (e.g., credit rating) and properties of the specific "behavior" which indicates fraud (e.g., number of international calls in one day); and the need for a new definition of accuracy: We need to find rules which do not necessarily classify correctly each individual "usage sample" as either fraudulent or not, but ensure the identification, with a minimum of wasted cost and effort, of most of the fraud "cases" (i.e., defrauded customers).

These aspects require a special-purpose rule discovery system. We present as an example a two-stage system based on adaptation of the C4.5 rule generator, with an additional rule selection mechanism. Our experimental results indicate that this route is very promising.

Keywords

Telecommunications, Fraud, Rule discovery.

1. INTRODUCTION

The two mature applications developed at Amdocs, which make widespread use of data mining techniques and algorithms are Churn Management and Fraud Analysis.

In this paper we first give a brief review of the data mining aspects of our Churn Management application. We devote the bulk of the discussion to the analysis and solution of one of the interesting data-mining problems, which arose within the Fraud Analysis application.

2. CHURN MANAGEMET

In general "churn" refers to the process of customers switching their carrier or service provider. The Churn Management application aims at the dual purposes of understanding the driving forces and reasons behind churn and predicting the likely future churners.

Using "white-box" methods of rule-discovery, the system generates rules or segments that describe the patterns relevant to churn. These rules can help an analyst understand the reasons for churn and devise preventive measures. Thus the analyst can combine the automatically generated knowledge with his domain expertise. An examples of automatically generated patterns (this is a slightly modified "real life" example):

- Customers who make many international calls, and whose overall usage is low, tend to churn. This pattern had an explanation, as it was cheaper to make international calls from one of the competitors.

The second stage of the Churn Management process involves the building of a prediction model, which serves to predict the churn likelihood for current customers in the next month (or few months). Our experience shows that a prediction model enables the operator to find between 20% and 50% of all churners in the top 2% of the list of customers, ranked by their churn prediction scores.

3. FRAUD IN TELECOMMUNCATIONS

The telecommunications industry suffers major losses due to fraud. The various types of fraud may be classified into two categories:

Subscription fraud - fraudsters obtain an account without intention to pay the bill. In such cases, abnormal usage occurs throughout the active period of the account. The account is usually used for call selling or intensive self-usage. Cases of bad debt, where customers who do not necessarily have fraudulent intentions never pay a single bill, also fall into this category. These cases, while not always considered as "fraud", are also interesting and should be identified.

Superimposed fraud - fraudsters "take over" a legitimate account. In such cases, the abnormal usage is superimposed upon the normal usage of the legitimate customers. Examples of such cases include cellular cloning, calling card theft and cellular handset theft.

Call details alone are not enough to establish cases of fraud. A certain call may be perfectly normal in one situation, but indicate fraud in another. For example, a call to a Premium Rate Service may be normal if the customer usually makes such calls, but suspicious otherwise. Usage volume (total number, duration or rated value of calls over a certain period) is also crucial in

KDD-99 San Diego CA USA
Copyright ACM 1999 1-58113-143-7/99/08...$5.00

establishing a fraud case. Finally, customer details, such as price plan and credit rating, are important in fraud analysis, especially in the analysis of subscription fraud. For example, customers with a poor credit rating may need to be monitored with tighter thresholds, and new customers can be monitored with unique fraud patterns. The use of customer data can refine the fraud patterns, and increase the accuracy of these patterns. A fraud detection system should therefore consider the context of the call (customer details and representation of customer's normal behavior) and usage volume, in addition to call details.

Previous work in the field of fraud detection has concentrated mainly on identifying superimposed fraud. Most techniques described in literature use CDR (call detail record) data to create behavior profiles for the customer, and detect deviations from these profiles. [6] describe a rule based system that monitors the average and standard deviation of the daily number and duration of calls of certain characteristics (e.g., international calls), and compares cases against a user-defined threshold in terms of standard deviations. [5] use similar profiles, but learn from known cases a fraud model (combination of monitors and monitor thresholds) using a neural network. [3] do not use predefined monitors, but learn from historical data which monitors are relevant to fraud. Other methods profile customer behavior without using specific predefined patterns ([4], [2]). None of the mentioned techniques uses customer data in the fraud analysis. None of the systems is aimed at or capable of identifying subscription fraud.

4. RULE BASED FRAUD DETECTION

Many commercial fraud analysis applications are based on rules. In a rule-based fraud detection system, fraud patterns are defined as rules. Rules may consist of one or more conditions. When all conditions are met, an alert is raised.

Data of 3 types may participate in rule conditions: call details, customer details and behavior monitors. Behavior monitors are summations of number, duration or rated value of calls over a certain time window (e.g., the daily number of calls to mobile phones at off-peak hours). Any population of calls can be monitored. For identifying superimposed fraud, "normalized" monitors can be used. These monitors denote the measured value in terms of standard deviations from the average value. High value of such monitor indicates an extreme increase in usage, and can be used in a superimposed fraud rule. Example for rules may be:

credit_rating = C AND daily_international_calls_duration > 2hrs => alert

deposit = X AND normalized_daily_duration > 4 standard deviations => alert

The alerts are gathered into cases (a case for each account) together with account data and CDRs. The cases are the starting point of the manual investigation process, where a human analyst determines for each case whether it is actually fraudulent or not.

Traditionally, fraud rules are defined according to expert knowledge of the fraud analyst. However, new fraud techniques emerge constantly, and some patterns are not intuitive. Therefore, it is important to complete the expert knowledge by automatically discovering fraud patterns in historical data. In addition, rule discovery process can help in fine-tuning existing rules or thresholds, in order to minimize alerts. Rule discovery methods, which output lists of fraud rules ("white box"), enable the analyst to easily interpret the results and understand the reasons behind them (contrary to "black box" models, such as neural networks). This way, the analyst can incorporate the discovered knowledge into the existing system. If an appropriate rule discovery procedure is used, the resulting discovered rules may be added to the existing rule-set in the system, and enhance it.

In this paper, we aim at understanding the unique problem of rule-discovery for fraud analysis. We show why standard rule-discovery methodologies, used in the classification context, are inappropriate for this problem, and suggest alternatives for both the rule-discovery methodology and the algorithms used within it.

5. UNIQUENESS OF THE PROBLEM

The problem of finding fraud has some features that make it different from standard classification and rule-discovery problems in other data-mining domains. An attempt to use the standard algorithms and methods will result in unsatisfactory results. Following are some of the main points, which make this problem unique and require either special-purpose algorithms and new methodologies, or at least, significant adjustments to existing ones.

5.1 Two Data Levels

We must first tackle the problem of "where the patterns live". There are at least two separate levels of data, and sometimes more. One level is the customer data, Examples of such attributes are customer's age, ethnicity and family status, price plan and telephone model. The second level is what we have termed "behavior"-level data. This term refers to usage characteristics in a short time frame (typically a single day). Typical behavior-level attributes are the number of international calls in a day and total duration of all calls in a day. They may also include "normalized" behavior monitors detecting changes in behavior relative to the history of usage by this particular customer.

Our goal is to find patterns combining elements from both levels, giving rules such as the following: "People who have a particular price plan that makes international calls expensive and who display a sharp rise in international calls are likely the victims of cloning fraud".

Finding such rules with standard classification-rules generators, such as C4.5 [7] or CART [1] is problematic. If we try to use them, the data should first be arranged with one record per "behavior sample" (which can be, for example, a list of daily behavior monitors). Then, the customer properties are actually "duplicated" for all the records of the same customer. A standard rule-generation algorithm, like C4.5, will then typically find rules using only customer-level attributes that will not represent true patterns but rather results of the duplication effect. For example, a rule can state that a customer named X is likely to be the fraudulent, because the training data contains 100 records (behavior samples) for a single customer named X, all classified as fraudulent due to subscription fraud. Thus this rule will have coverage of 100 "behavior samples" and accuracy of 100%. However, the true coverage of this rule is clearly one (customer) rather than 100 (records) – because it uses only customer attributes. Hence it has no generalization ability.

In section 6 we discuss the rule-discovery techniques that can tackle two-level data and assure the generation of appropriate two-level rules.

5.2 Requirements from "Good" Rules

In the fraud analysis context, the generated rules will be used as alarm-setters for suspected fraud. Therefore, we would like to generate rules that are appropriate for this task, rather than for standard machine learning tasks such as classification or scoring.

In a rule-based fraud management systems, the alarms (or alerts) are usually not treated individually but rather combined at the customer level into "cases" of suspected fraud. Thus, K alerts generated for the same customer result in only one case being created, while K alerts generated for K different customers, result in K different cases being created. If we just count the number of true alarms (i.e., alerts that are actually fraudulent) and false alarms, the two situations would be identical. Thus, it is generally true that accuracy should be computed at the customer (case) level – the "higher" of the two levels mentioned above. The success of a fraud rule is determined by how many really fraudulent cases were identified and how many cases were false alarms.

The coverage of a rule should similarly be computed at the customer level, to indicate the variety of fraudulent cases covered by this pattern. However, the coverage in "records" (the total number of alerts generated by this rule) is also of interest, as it indicates the number of alerts per case. A case with many alerts is likely to both appear earlier (i.e., the first alert for the case will be generated soon after the fraud starts) and be treated more quickly.

Thus the criteria for the quality of a rule are:

- High accuracy in cases (=> specificity - most cases found are really fraudulent).
- High coverage of true fraud cases (=> sensitivity - most fraudulent cases are found).
- High coverage of true fraud alerts (=> fraud cases are detected quickly).

5.3 Requirements from "Good" Rule-Sets

Within a rule-based system the performance of each individual rule is secondary in importance. The main issue is, of course, the performance of the rule-set selected for use in the system. Our ultimate goal in the rule-discovery process should be to select a rule-set that maximizes the three criteria mentioned above, rather than individual rules with desirable properties.

Another criterion of quality for a rule-set is its ability to reflect different patterns related to fraud. We would like our rules to be different from each other in the sense that they reflect different kinds of "behavior" patterns. This difference should be reflected in a relatively small overlap between the sets of "behaviors" belonging to the rules within our sample of fraudulent and legitimate customer behaviors. In other words, difference in patterns should be measured at the "behavior" level (the lower of the two). So, two rules can describe completely different patterns, with no overlap in the behavior samples belonging to the rules, but give alerts for the same customers, because all these customers have both patterns. In such a case, we would consider these rules as different, and would like to find both rules.

We would also like to have a relatively small number of rules in the selected rule-set. The rule-discovery results are supposed to serve only as an increment to the hand crafted rules, which are the main set of rules in the rule-based system. The analysts managing the system should be able to understand the rule-discovery results and integrate them into the system, while retaining their control over the whole process. Therefore, generating small rule-sets is essential for using rule-discovery in this context.

Thus our qualitative demands from a good rule-set are:

- High specificity and sensitivity at the customer (case) level.
- Large number of true alerts at the behavior (record) level.
- The rule-set should contain rules that are not "similar", i.e. rules that capture different behavior patterns.
- The number of rules in the rule-set should be small.

6. BI-LEVEL RULE GENERATION

There are several possible approaches to constructing correct bi-level rules. One is to abandon standard rule-generation procedures completely in favor of simpler ad-hoc methods. For example, we could use a standard procedure to build rules on customer attributes only, using a database with one record per customer, then run a separate second stage with one record per "behavior sample" to add behavior attributes to the rules. This naive approach is unlikely to give good results, as it would be limited in its ability to find "interactions" between customer-level and behavior-level attributes (e.g., that customers in a certain area are likely to be fraudulent if they make many international calls).

Another approach is to modify the existing algorithms to ensure that they count the records correctly, taking into account the issue of bi-level data. We have taken this approach, and have built a rule generator based on a modification of the C4.5 algorithm. Section 6 presents an example of rules generated by the system prior to the modification, compared to the results with the bi-level modification. We assume that the reader is familiar with the use of C4.5 as a rule generation program and with its basic algorithms.

The relevant changes in the algorithm are concentrated in three areas – splitting criterion and stopping rule for tree construction and pruning significance tests. The splitting criterion is used to select the "best" greedy split in each stage during tree construction. It is based on calculating the "information content" of each of the suggested splits with regard to the class distribution and choosing the one with the highest content. The stopping rule dictates the size of groups we are willing to accept as "leaves" in the tree. The goal of using a stopping rule is to prevent the system from creating rules representing small samples with no statistical generalization ability.

For both of these areas the key to working on bi-level data is that the "size of groups" concept has to be defined with respect to the level at which the attribute being split belongs. So, when splitting on a customer-level attribute, the amount of customers of each class found in each "leaf" is counted. When splitting on a behavior-level attribute we should count the number of instances of behavior (i.e. the number of "records") of each class in each "leaf".

The idea of the necessary changes for the pruning significance tests is similar. The example in section 6, in fact, includes only the changes in the splitting criterion and stopping rule, which in most experiments have been sufficient for creating good bi-level rules. Future development of special-purpose algorithms for bi-level rule generation are discussed in section 9.

7. SOLUTION: TWO-STAGE PROCESS

We have developed an approach to handle the unique problem of rule-discovery for fraud based on a partition of the rule-discovery process into two independent components. This gives the system the maximum flexibility to adapt to the various challenges it faces. We have completely separated the rule-generation stage from the rule-selection (or rule-set selection) stage in the system. The general architecture of the system can be seen in figure 1.

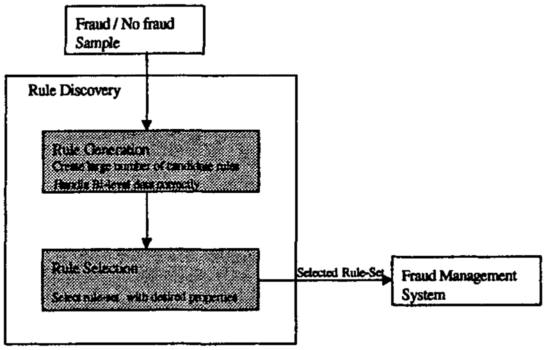
Fraud / No fraud
Sample
Rule Discovery
Rule Generation
Create large number of candidate rules
Handle Bi-level data correctly
Rule Selection
Select rule-set with desired properties
Selected Rule-Set
Fraud Management
System

Figure 1: Architecture of 2-stage Solution

The rule-generation step is used to create a large number of candidate rules, and could include multiple rule-generation applications, each of them generating a set of rules. All rule generators should be able to handle the bi-level problem discussed in section 6, as was done for our version of C4.5.

7.1 Rule Selection Methodology

The rule selection stage receives as input all the candidate rules, generated in the pervious stage, as well as classified customer and behavior data. It applies a selection procedure that produces as output a rule-set with the desirable quality properties. The procedure designed here is a greedy algorithm, which is divided into two main sub-procedures.

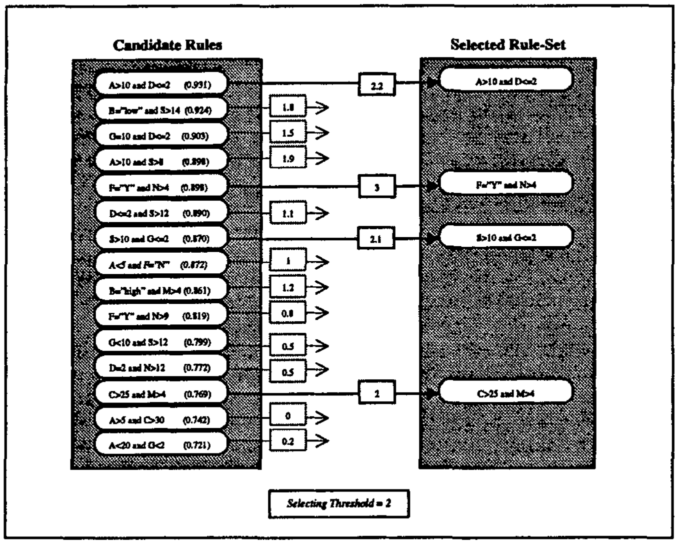

Figure 2: Illustration of rule-selection methodology

The first sub-procedure sorts all candidate rules according to a single rule quality criterion, while the second performs the actual selection. It scans all the sorted rules sequentially, beginning with the "best" rule. Each rule is selected (or not) and added to the rule-set according to rule-set quality criteria, defined in terms of thresholds on various quality measures. Thus, the selection decision is affected not only by a rule's individual performance, but also by its incremental performance relative to the previously selected rules. The predefined thresholds can be adjusted according to specific aims, e.g., high accuracy, small rule-set, etc. Figure 2 illustrates the selection procedure.

In our implementation, the criterion for sorting the rules was their accuracy in terms of cases, measured at the customer level. The selection criteria used for deciding whether to add a new rule included incremental fraud coverage in terms of cases, measured at the customer level, as well as the difference between rules, measured at the behavior level. The difference between two rules is indicated by their correlation (measured at the behavior or record level).

So, when considering selecting a new rule (NR) we compute its additional fraud coverage (AFC) and its maximum correlation (MC) with the rules in the selected group (SR) (FC refers to the fraud coverage of a rule-set in cases):

- $AFC = FC_{(SR \cup NR)} - FC_{(SR)}$
- $MC = MAX\{correlation\ (NR, SR_i)\}$ $\quad SR_i \in SR$

If the rule's additional fraud coverage is at least T_{AFC} and its maximum correlation with a previously selected rule is at most T_{MC} then the rule is selected. We can emphasize various aspects of the desired properties for resulting rule-sets by tuning these threshold levels (T_{AFC}, T_{MC}).

8. EXAMPLE: IDENIFYING BAD DEBT

The data for these experiments came from a cellular carrier with a considerable number of bad debt ("never paid") customers, who use the network and never pay a single bill. Our data set included a few hundred "legitimate" customers and a few hundred bad debt customers. We used as "behavior samples" daily summaries of usage of the different kinds, such as total calls, local calls, calls to mobile, and international calls. As we were dealing with subscription fraud starting from day one, there was no need for monitoring changes in behavior. We also had customer-level data such as the customer's type, his credit limit, residence area, etc. Thus we had in our data a record for each day of each customer, with customer data replicated for all the daily records of the same customer, and with daily behavior monitors.

We first ran the data through the standard C4.5 rule-discovery system, trying to assess the impact of not handling bi-level data correctly. Not surprisingly, the generated rules, used customer features almost exclusively, and some of them were as silly as "If a customer's name is X and he lives in area Y he is likely to be fraudulent". The program failed in finding rules successfully combining customer-level and behavior-level attributes.

Next we ran the same data through our "bi-level compliant" version of the c4.5 engine, making sure that the coverage of tree-leaves, hence of the generated rules, was calculated correctly. The result was now numerous sensible patterns describing both single indicators and combinations of indicators of likely fraud. Some of the rules described new, unexpected patterns. For confidentiality reasons we cannot display the actual rules discovered. As an example, however, we give the structure of some of the rules we got:

credit_limit < X AND total_usage_type1 > Y AND total_usage_type2 < Z => alert

For the purpose of evaluating the performance of the rule selection stage, the set of cases (customers) was divided randomly into a training set and a test set. The selection procedure received 47 candidate rules and was implemented solely on the training group. A total of 35 threshold configurations were made, using T_{AFC} values in [0.01, 0.05] and T_{MC} values in [0.3, 0.9]. Some threshold combinations produced the same rule-set. The results, i.e., the selected rule-set, were tested against the test group. For each selected set, four performance attributes were measured:

Set Size = Number of selected rules

$$\text{Accuracy} = \frac{\text{Number of detected fraud customers}}{\text{Number of customers classified fraud}}$$

$$\text{Fraud Coverage} = \frac{\text{Number of detected fraud customers}}{\text{Total number of fraud customers}}$$

$$\text{Maximum Correlation} = \text{MAX}\{\text{correlation (SRi, SRj)}\}\ i \neq j$$

Table 1 presents the performance attributes of the selected rule-set using different thresholds. For example, in iteration #1, loose thresholds were used, demanding an additional coverage of only 1% and allowing correlation between selected rules as high as 0.9. The thresholds produced a rule set containing 5 rules, in which 90.1% of the cases classified as fraud were indeed fraudulent. This rule-set detected 90.1% of the fraud cases but at least two rules in the set represent the same pattern, since they are highly correlated (0.89). It is clear that the thresholds used in iteration #9 produced a better rule-set. This set, with fewer rules, has the same fraud coverage with higher accuracy and lower maximum correlation. However, the comparison is not always that obvious. For example, set #2 detects a higher fraud rate than set #1 with lower maximum correlation but is less accurate. If we want to enable comparability between all rule-sets, the performance measures should be prioritized or assigned weights.

#	T_{AFC}	T_{MC}	Set Size	Accuracy	Fraud Coverage	Max Cor.
1	1%	0.9	5	90.1%	90.1%	0.89
2	1%	0.7	5	89.0%	92.7%	0.65
3	1%	0.5	4	89.1%	89.1%	0.47
4	1%	0.3	4	87.8%	89.6%	0.4
5	2%	0.6	4	90.1%	90.1%	0.5
6	2%	0.5	4	89.2%	90.1%	0.44
7	2%	0.4	4	89.0%	92.7%	0.41
8	3%	0.3	3	92.7%	85.4%	0.4
9	4%	0.6	3	91.5%	90.1%	0.5
10	5%	0.4	3	90.4%	92.7%	0.41
11	5%	0.3	2	94.3%	85.4%	0.17

Table 1: Results of the rule-selection procedure

9. CONCLUSION AND FUTURE WORK

This paper illustrates the need for "vertical" applications, analyzing a particular business problem in a particular industry, and tailoring a specific solution for a specific problem, rather than attempting to utilize standard tools and algorithms. In the case of rule discovery for fraud, we believe that understanding the unique features and identifying the points at which the standard tools were falling short were the key steps to suggesting a successful alternative approach.

In general, the contribution of this paper can be divided into two parts: description of the unique features of rule-discovery for telecommunications fraud (bi-level data, special rule and rule-set quality criteria) and suggestion of solutions for the particular problems and an appropriate framework for the whole process.

We believe that there is room for further research, especially concerning the design and implementation of new algorithms for rule-discovery for fraud. Among the directions we are pursuing:

Exhaustive rule generation: Use the sparseness of fraud examples within the data to build an exhaustive mechanism for finding all possible rules containing a substantial number of fraud examples. As the two-level problem is similar in structure to the hierarchy problem which [8] discuss, it seems especially appropriate to utilize some of the ideas from their domain of Association Rules discovery. The advantage here is that the rule-generation step will not lose any of the possible candidates for "good" rules, and will not require complicated mechanisms. This would create a very large number of candidate rules, out of which select a "good" subset of rules will be selected by an efficient rule-selection mechanism.

Non-greedy rule-selection procedures: Optimization methods such as Simulated Annealing and Genetic Programming can be utilized to create a more robust selection process, with a better chance of finding the "best" rule-set.

10. REFERENCES

[1] Breiman, L., J. H. Friedman, R. A. Olshen and C. J. Stone (1984). Classification and Regression Trees. Chapman Hall.

[2] Burge, P. and J. Shawe-Taylor (1997). Detecting Cellular Fraud Using Adaptive Prototypes. Proceedings of AAAI-97 Workshop on AI Approaches to Fraud Detection and Risk Management, Providence, RI, 9-13.

[3] Fawcett, T. and F. Provost (1997). Adaptive Fraud Detection. Data Mining and Knowledge Discovery. U. Fayyad, H. Mannila and G. Piatetsky-Shapiro (Eds.), Kluwer Academic Publishers, Boston, CA. vol 1, 291-316.

[4] Kokkinaki, A. I. (1997). On Atypical Database Transactions: Identification of Probable Fraud using Machine Learning for User Profiling. Proceedings of IEEE Knowledge and Data Engineering Exchange Workshop, 107-113.

[5] Moreau, Y. and J. Vandewalle (1997). Detection of Mobile Phone Fraud using Supervised Neural Networks: A First Prototype. Available via ftp://ftp.esat.kuleuven.ac.be/pub/SISTA/moreau/reports/icann97_TR97-44.ps.

[6] Moreau, Y., B. Preneel, P. Burge, J. Shawe-Taylor, C. Stoermann and C. Cook (1997). Novel Techniques for Fraud Detection in Mobile Telecommunication Networks. ACTS Mobile Summit, Grenada, Spain.

[7] Quinlan, J. R. (1993). C4.5 – Programs for Machine Learning. Morgan Kaufmann.

[8] Srikant, R. and R. Aggrawal (1995). Mining Generalized Association Rules. In Proceedings of the 21st International Conference on Very Large Data Bases, 417-419.