MINING THE STRONGEST PATTERNS IN MEDICAL SEQUENTIAL DATA

J. Kléma*, T. Holas*, F. Železný* and F. Karel*

* Gerstner Laboratory, Department of Cybernetics,
Czech Technical University, Technická 2,
166 27 Prague, Czech Republic

{klema,zelezny}@labe.felk.cvut.cz, {holas,karel}@fel.cvut.cz

Abstract: Sequential data represent an important source of automatically mined and potentially new medical knowledge. They can originate in various ways. Within the presented domain they come from a longitudinal preventive study of atherosclerosis – the data consist of series of long-term observations recording the development of risk factors and associated conditions. The intention is to identify frequent sequential patterns having any relation to an onset of any of the observed cardiovascular diseases. This paper focuses on application of inductive logic programming. The prospective patterns are based on first-order features automatically extracted from the sequential data. The features are further grouped in order to reach final complex patterns expressed as rules. The presented approach is also compared with the approaches published earlier (windowing, episode rules).

Introduction

Medical databases have accumulated large quantities of information about patients and their clinical conditions. Relationships and patterns hidden within this data can provide new medical knowledge, which has been proven in a number of medical data mining applications. However, the data are rarely provided in a format suitable for immediate application of conventional attribute-valued learning (AVL). In some tasks, a domain-independent preprocessing methodology (e.g., feature selection) is sufficient. In other tasks, domain-specific preprocessing shows vital and may strongly increase mining performance. But, the domain-specific algorithms are frequently conducted by the trial-and-error method, which is often time consuming and demands both for experienced researcher and medical expert.

The presented paper focuses on mining temporal and sequential medical data which usually ask for complex and sophisticated preprocessing. A sequence is understood as a sequence of events where each event is described by its value altogether with a time stamp. Event types can also be distinguished. Whole dataset can either contain a single sequence or it can be composed of an arbitrary number of shorter and independent sequences. The ultimate goal is to identify strong sequential patterns, i.e., such event chains (sub-sequences) that appear frequently in the dataset and optionally study their interaction with the target event. The typical target event in a medical application can be a disease manifestation or a change of the state of health.

In particular, the paper centers on the study Stulong [1], a longitudinal primary preventive study of middle-aged men lasting twenty years. The study contains data resulting from observation of approximately 1400 men, the main intention of the project was to identify risk factors of atherosclerosis. The data is inherently multi-relational. The main attention is paid to the table of checkups including results of a series of long-term observations recording the development of risk factors and associated conditions. A single man represents a single sequence, i.e., the task deals with a base of sequences. Since men were followed for different time periods - some of them underwent 20 checkups while others many fewer - the sequences vary heavily in their length. There are several risk factors followed (BMI, blood pressure, biochemical explorations) - the task has to also consider different event types. Finally, the immediate measurements represent the event values. Disease appearance is also recorded and time-stamped.

The intention of the study mentioned-above can be rephrased in terms of sequential data mining as follows. The intention is to identify frequent sequential patterns having any relation to an onset of any of the observed cardiovascular diseases (CVDs). Examples of such patterns can be following: (1) when BMI goes down and then it increases again while blood pressure increases then CVD is more likely to appear, (2) when BMI increases and HDL cholesterol is low then CVD is more likely to appear.

The Stulong study made a subject of ECML/PKDD data mining challenge in past years. A great majority of contributions took no account of the issue of sequential mining, nevertheless several papers relevant to this issue appeared. Our former paper [3] presents a windowing, the domain-specific approach based on trend features generated through aggregation windows. Incidentally, the examples of patterns mentioned above represent two of true outcomes of this approach. The second approach [5] mines for episode rules with a universal tool WinMiner. Besides the domain independence, the added value consists in supplying the optimal window sizes of the discovered relations.

This paper applies another paradigm becoming popular in domains structured as the Stulong study - inductive logic programming. The paper applies a general tool RSD [4] for relational subgroup discovery in individual-centered domains. The prospective patterns are expressed in a form of first-order features automatically extracted from the sequential data. Relevance of these features can be then studied in terms of AVL - the features can even be grouped in order to reach final complex patterns.

The main contribution of the paper lies in the RSD application as well as in general comparison with the approaches published earlier. The paper compares the reached results simultaneously discussing issues of simplicity, comprehensibility and reusability.

RSD: Relational Subgroup Discovery

Relational rule learning is typically used in solving classification and prediction tasks. The former research within the Stulong domain has proven that the discovered patterns (and undoubtedly hidden ones too) do not show the strength to reliably distinguish between diseased and healthy individuals a priory. The task should rather be defined as subgroup discovery. The input is a population of individuals (middle-aged men) and a property of those individuals we are interested in (a CVD onset), and the output are population subgroups that are statistically "most interesting": are as large as possible, have the most unusual statistical (distributional) characteristics with respect to the property of interest and are sufficiently distinct for detecting most of the target population. The definition of subgroups arises out of the sequential patterns reflecting temporal development of risk factors.

Relational rule learning can be adapted also to subgroup discovery. A relational subgroup discovery system RSD has been devised [4]. It is based on principles that employ the following main ingredients: exhaustive first-order feature construction, elimination of irrelevant features, implementation of a relational rule learner, use of the weighted covering algorithm and incorporation of example weights into the weighted relative accuracy heuristic.

The whole process can be simplified as follows. The system tries to construct features first, i.e., conjunctions of literals available in the domain. Their critical property is potential to form subgroups as defined above. Then, the features are grouped into rules, whose critical property is very similar. They only stress the coverage issue, i.e., they try to cover as many target individuals that have not been covered yet as possible (for details see [4, 6]).

Mining The Stulong Data

Feasibility, complexity, resolution

When mining the Stulong data, the most general and natural approach seems to be to allow arbitrary sequential features. Those features capture sequences of arbitrary length and they are inherently *inter-transactional*, i.e., each sequence may contain *events* from different *risk factors*. Two examples of such sequences/features that emphasize time relations are shown in Figure 1. Time relations are modelled by binary predicates $after_1$, $after_2$, ..., $after_n$ – meaning that the second event occurred 1, 2 or n checkups after the first event – and $\mathit{simultaneous}$ – meaning that the events occurred in the same checkup. Of course, there could also have been defined various generalizations of $after$ predicate, e.g., the second event occurred at an arbitrary checkup following the checkup of the first event. Let us point out that the checkups are slightly irregular in time but for the sake of simplicity we consider the checkups being annual in this text.

In order to minimize preprocessing work, the continuous risk factors can be discretized by another set of predicates (e.g., weight_cat(X, small) :- X < 64.). This approach can also bring a higher variability in definition of events as the event can be understood as an immediate feature value ($weight(checkup_i, 71)$) or a category ($weight(checkup_i, X)$, $weight\_cat(X, xsmall)$). Then, the (simplified) textual representation of a feature can be as follows:

```
feature(ID,PAT):-checkup(PAT,Time1),
checkup(PAT,Time2), after1(Time1,Time2),
syst(Time1,V1), syst_cat(V1,low),
syst(Time2,V2), syst_cat(V2,high).
```

The feature holds for each individual/patient having two consecutive checkups, whose systolic blood pressure value changes from "low" category to "high" category. It can be seen that each event corresponds to three predicates (defining patient/time, risk factor/value and category), moreover, the events have to be associated with a time predicate.

Although the variability of candidate sequences is desirable from the point of view of the final practical knowledge, it can hinder the feasibility of sequence space search. The number of generated features can become exceedingly high and disable to generate the rules in a reasonable time. Suppose we have a number of attributes a, a number of values of each attribute v, and a length of a sequence l. Then, the amount of possible single-transactional sequences is $O(n_s) = v^l$ bounded, while the number of inter-transactional sequences is $O(n_i) = (av)^l$. The amount of sequences grows exponentially with the maximal allowed sequence length. Computation is even more cumbersome when considering features. As mentioned above, the feature length multiply exceeds the sequence length (as each event corresponds to more predicates and the events have to be mutually organized), while computational burden grows exponentially with the maximal allowed feature length again. In some sense, the feature space exceeds the originally intended sequence space since the system cannot distinguish between meaningful and pointless features (that do not correspond to any sequence) [1].

Therefore the feature and consequently the sequence length have to be limited as well as the number of values being distinguished as events and the number of risk factors. As a result, sequences which are long and consisting of many attributes with many different values cannot be generated.

It follows that searching for inter-transactional sequences is computationally very demanding. Let us estimate the number of candidate sequences in the Stulong domain. The number of checkups varies from 1 to 21, around 80% of men were measured for 5 and more times – it seems to be reasonable to allow for the sequences limited by 5 events. The group of the most significant variables consists of 5 risk factors (the systolic and diastolic blood pressure (SYST, DIAST), level of cholesterol in mg%(CHLSTMG), triglyceride level in mg%(TRIGLMG) and body mass index (BMI), there were tens of different values measured. Obviously, there are tens of billions of candidate sequences.

Consequently, the number of attributes has to be cut down, which causes loss of the information about the relationships between attributes. At the same time the length of sequences needs to be cut down, which reduces resolution in the time domain. The number of possible attribute values has to be lowered (a reasonable amount of discretization categories can only be used), which reduces the resolution in the data domain. The inter-transactional nature of sequences may therefore be seen by some rather a problem than a feature, but we have to keep in mind, that albeit it's computational intensity it is a new way of handling information and, as proposed in [2], new and more effective algorithms of inter-transactional rules processing are being developed. We have spent some time trying to find an equilibrium between the number of attributes and the length of sequences and then we decided to take kind of a "third way" and we divided the data to 3 disjunct windows as described thereunder.

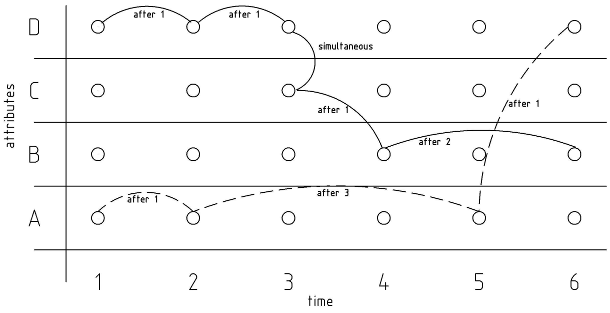

Figure 1: Inter-transactional sequences in Prolog

Data preprocessing

One of the first tasks we have to cope with in order to use RSD effectively is preprocessing. RSD loads and interprets language declarations and data in a predicate logic format [6]. Converting the dataset from simple tables to Prolog code involves a lot of work, and it would be almost impossible to do this by hand. To address this problem, a Java conversion program has been developed. It is a simple console program run from the command line, which reads the data in the comma separated values (CSV) format and outputs .pl (Prolog code) and .b (background knowledge) files. The Java program is in early alpha version, it will be made public later. The final relational representation covers patients, times of examinations and examination data themselves. The background knowledge defines how to work with time, what is a time sequence, and what elements can individual features consist of. The time sequences vary in their length, CVDs may appear at their end only. Apparently, the most recent measurements at the end of the sequence are also the most important as they are most likely to affect the current state of the patient.

The main preprocessing tasks are: (1) to generate the Prolog code for feature template construction, (2) to carry out attribute discretization and (3) to perform trend construction. While the first task is necessary formatting, the other two tasks address effectiveness. As outlined in the previous section, immediate utilization of Prolog predicates for preprocessing turned out to be quite ineffective, because an extra predicate is needed for each discretization made, which effectively doubles the length of the features and decreases computational effectiveness of feature generation. Thus, the features were discretized in advance in terms of preprocessing. The following discretized attributes were generated: NORMBMI, NORMSYST (NORMDIAST), NORMCHLSTMG, NORMTRIGLMG. All

[1] RSD by no means generates arbitrary features, i.e., arbitrary conjunctions of literals. The feature space is implicitly reduced as every variable has to be used as the input variable at least once, features cannot be decomposable, predicates can be defined as antisymmetric, etc. The real computation also depends on background knowledge design that can introduce high-level predicates further reducing the feature space.

those attributes are derived from appropriate Stulong risk factors BMI, SYST, DIAST, CHLSTMG and TRIGLMG mentioned earlier. The discretized attributes were transformed from the original attributes by equidistant discretization into 3 intervals referred to as "low", "medium" and "high" [2].

Another way that helps to simplify feature construction and that makes it more effective is introduction of short-time trends. The attributes TRENDBMI, TRENDSYST (TRENDDIAST), TRENDCHLSTMG, TRENDTRIGLMG represent transformations of original sequences, which are reflecting the speed of change of the attribute value in time. Possible values of the "trend" attributes are "down2", "down", "flat", "up", and "up2", meaning "big decrease", "decrease", "no change", "increase", and "big increase" of the attribute value respectively. When dealing with trend attributes simplification is obvious. The feature that holds for each patient having two consecutive checkups, whose systolic blood pressure value changes from "low" category to "high" category introduced in the previous section can be expressed as follows:

```
feature(ID,PAC):-checkup(PAC,Time1),
trendsyst(Time1,big_increase).
```

The target (class) attribute CVD is a binary attribute signalling the presence of an cardiovascular disease at the end sequence corresponding to the given individual (0 – non diseased, 1 – diseased).

The final set-up

Preprocessing proposed and implemented in the previous section reasonably reduces the feature length while preserving the complexity of the underlying sequence. To finish the final set-up, proximity of CVD onset has to be also quantified. The length of the original sequences varies from 1 to 21 checkups, the average is around 8. The individual sequences (SYST, BMI, etc.) were divided into 3 disjunct windows called *begin, middle, end*, where *end* covers last 4 events, *middle* covers another 4 *events* before the *end*, and *begin* covers the rest - all the events from the beginning of the sequence to the middle window. Each generated feature is located in one of those windows and it may contain one sequence of a maximum length of 2. The time predicates $after_i$ were replaced by the binary predicates $after\_beg$, $after\_mid$ and $after\_end$ defining that the second event occurred an arbitrary time after the first event and both of the events are located in the same window (*beg* stands for the beginning window etc.).

When combining features into *rules*, each *rule* consists of a maximum of 3 *features*. This gives us an opportunity to describe a sequence with a maximum length of 6. Those numbers may of course vary in future applications, but the principle will be essentially the same. Examples of the final rules can be seen in the next section.

Results

In this section, selected generated rules and their interpretations are presented. Let us take a look at the following rule:

```
class:0, conf:0.968, cov:0.156, lift:1.308
f(7369,A):-checkup(A,B), normsyst(B,medium),
trendbmi(B,flat), trendsyst(B,up).
f(3068,A):-checkup(A,B), checkup(A,C),
after_mid(C,B), trendbmi(C,flat).
f(1158,A):-checkup(A,B), checkup(A,C),
after_beg(C,B), normtriglmg(B,low),
trendtriglmg(C,up2).
```

The rules have the same form as the classical decision rules $Cond \Rightarrow Class$, where Cond (premise) is "object satisfies all the listed features" and Class (result) is "object is assigned the listed class". However, the rules are not used to classify the individuals but to distinguish interesting subgroups. Thus they can also or rather better be viewed and treated as association rules $Ant \Rightarrow Suc$. As a matter of fact, classical association rule characteristics serve for the purpose of their evaluation – they can be viewed at the first row of the rule. Class 0 suggests that the rule concerns non diseased individuals. *Coverage* $cov = n(Ant)/n$, where $n(Ant)$ is the number of instances covered by the rule's antecedent, n is the number of all patients. Coverage is the fraction of patients covered by the rule. Rules with low coverage (5% or less) are usually considered useless. *Confidence* $conf = n(Ant \cap Suc)/n(Ant)$ is the accuracy of the rule. It expresses how many instances that satisfy the premise also satisfy the result. *Lift* is defined as $lift = conf/p_a$, where $p_a = n(Suc)/n$ is the prior probability of the rule's class. It conveys how much better is the rule's performance compared to a trivial classifier, which assigns all instances into one class and its performance is the same as the prior probability of the class.

Remaining rows present the antecedent, namely 3 features, which have to be satisfied simultaneously. The meaning of the first feature is that the patient had an examination, in which he had medium systolic pressure, a steady trend of BMI and a rising trend of systolic pressure. The meaning of the second feature is, that the patient had two examinations (B and C) in the middle of the time sequence and examination C happened before examination B. In the examination C, he had a steady trend of BMI. The third feature says that the patient had two examinations (B and C) in the beginning of the time sequence and examination C happened before examination B. In examination C he had a steeply rising

[2] There are many alternate ways to discretize – a finer partitioning, equi-depth discretization or local approaches defining interval boundaries for every single patient separately could have also been applied.

trend of triglycerides and in examination B he had a low level of triglycerides. To summarize up all the three features: Our patient had long time ago a steep rise of triglycerides followed by a low level of triglycerides. Short time ago, he had a steady trend of BMI. At any time in history, he had a medium level of systolic pressure, steady trend of BMI and a rising trend of systolic pressure. Patient, who satisfies those conditions, has 30.8%[3] more chance of not having a cardiovascular disease than the average. Let's take a look at another rule:

```
class:1, conf:0.615, cov:0.049, lift:2.367
f(4380,A):-checkup(A,B), checkup(A,C),
after_end(C,B),normsyst(B,high),trendbmi(C,flat).
f(4124,A):-checkup(A,B),checkup(A,C),
after_end(C,B),normbmi(B,medium),trendchlstmg(C,up2).
f(4439,A):-checkup(A,B),checkup(A,C),
after_end(C,B),normsyst(B,high),trendchlstmg(C,up2).
```

This rule has a very good lift, but its coverage is on the edge of usefulness. So the rule is very strong, but valid only for a small fraction of instances. All the events are happening at the end of the sequence, very short time before the cardiovascular disease was found. This patient had flat trend of BMI followed by high systolic pressure, steeply rising trend of cholesterol level followed by medium level of BMI and high systolic pressure. To summarize a bit again, those features mean that the patient had normal BMI with steady trend, and after that he had a steep rise of cholesterol level followed by high systolic pressure. Patients, who satisfy those conditions, have a 137% more risk of cardiovascular disease than the average.

But we shall keep in mind, that this rule was induced from quite a small number of examples, so its predictive/descriptive value is limited. The value of the rule can be assumed from real group sizes. The coverage 0.049 implies that the rule covers 39 individuals. As the prior probability of the diseased class is around 26%, there are 10 diseased individuals expected in a randomly chosen group of 39 individuals while the rule covers 24 diseased individuals. Considering the binomial probability formula, there is only the probability $2.6e^{-6}$ that a rule covering 24 and more diseased out of 39 individuals occurs at random. Nevertheless, repeated trials have to also be taken into account as we have searched through a large number of potential rules.

Relational subgroup discovery can also be utilized for non-sequential data. In such a case, the application is still more straightforward resulting in rules as follows:

```
class:0, conf:0.910, cov:0.084, lift:1.230
f(9745,A):-liquors(A,none).
f(9737,A):-beer(A,more_than_1_liter).
```

This rule means that strong beer drinkers who do not drink liquors are 23% less likely to have a cardiovascular disease. When compared with traditional association rule mining or statistical analysis, the relational method outputs a comparable set of non-sequential rules (actually, the same rule as this was already found before). Sequential and non-sequential predicates/features can be naturally combined as demonstrated in the following rule:

```
class:1 conf:0.568, cov:0.055, lift:2.185
f(9738,A):-beer(A,occasionally).
f(8453,A):-checkup(A,B),normchlstmg(B,medium),
trendchlstmg(B,flat).
f(3787,A):-checkup(A,B),checkup(A,C),after_mid(C,B),
trendtriglmg(B,down2),trendtriglmg(C,flat).
```

The rule can be interpreted such that occasional beer drinkers with a normal cholesterol level with a steep drop of triglycerides level in blood have a 118% more chance of developing CVD. Of course, the coverage characteristic has to be considered again.

When putting those two rules together, one might infer that a good prevention of CVD is not to drink liquors, and to stop smoking (which is a common knowledge), but the interesting part is, that it also helps to drink a lot of beer, but not for people with dropping triglycerides in blood.

Table 1: Characteristics of the strongest found rules

Class	Confidence	Coverage	Lift
0	0.9	0.32	1.22
0	0.95	0.2	1.28
0	0.97	0.16	1.31
0	0.90	0.15	1.22
0	0.91	0.08	1.23
0	0.97	0.13	1.31
0	0.95	0.05	1.29
0	1.0	0.07	1.35
1	0.45	0.17	1.73
1	0.47	0.13	1.81
1	0.47	0.1	1.8
1	0.57	0.06	2.19
1	0.62	0.05	2.37
1	0.7	0.03	2.68

The presented examples demonstrate the structure and interpretation of the inferred rules. It is impossible to list all the meaningful rules and their interpretations. Table 1 gathers quantitative characteristics of the most promising rules which can express their real strength. Generally speaking, the rules having better coverage mostly have a lower lift. As the non-disease group is larger, the rules aimed at this group show better coverage, the diseased group is just the opposite. When confronted with a common medical knowledge, the majority of the rules seems to be sound, others may be considered as in-

[3] Taken from the lift characteristics, $p = (lift - 1) \cdot 100\%$

teresting, surprising or even contradictory.

Discussion

The generated rules are able to describe detailed interconnections between attributes in time, and are quite immune to errors coming from having too many different sequences because of minor changes in attribute values or time placement. On the other hand, the proposed method is not effective when used on systems, where those minor changes may have a major influence on the property of interest. The time axis is abruptly split while physiological nature of the modelled phenomenon ask for smooth treatment. The method is also better suited for finding local patterns than global models. When used directly for classification (i.e., intended for prediction), its performance is the same or worse than standard learning techniques (i.e., Decision Table, J48 Decision Tree, Bayesian network, etc.).

On the other hand, the proposed relational method is able to find patterns, which might be omitted by standard association rule learning algorithms and systems for mining non-intertransactional episode rules from sequential data. Generally speaking, the method can be considered as fully general. However, its performance is highly dependent on the data mining goal, the nature of the dataset and subsequent design of preprocessing and/or background knowledge.

Let us compare the presented method with its alternatives. Windowing is a simple and often used method to transform sequential data. The sequence of data can be either decomposed into several disjunctive windows or a sliding window approach can be applied. In both cases, the windows are subsequently replaced by aggregate attributes (linear trends are mostly used) and analyzed by traditional AVL. [3] applies the method of the fixed length sliding window to Stulong data. Although this method brought very good results in the particular task, it has to be tailored to the analyzed domain. Questions such as "what is the optimal window length?" or "is the linearization a proper generalization of the prospective patterns?" have to always be considered.

WinMiner [5] presents a general tool allowing to search for episode rules – patterns that can be extracted from a large event sequence. When dealing with the Stulong data, similar problems as discussed in this paper have to be solved first. In particular, a proper discretization has to be proposed in order to distinguish event types, inter-transactional patterns can be searched for if and only if the event types are distinguished also for distinct risk factors. During runtime the maximum time gap between the first and the last prospective events has to also be limited in order to cope with the exponential growth of the search space. The sequential patterns found by RSD and WinMiner are similar. In principal, RSD allows the user to better pre-specify the searched patterns via background knowledge and thus restrict the searched space which may, on the other hand, turn out to be time consuming and ask for expert knowledge in relational learning.

Acknowledgements

This research work was supported within the programme MSM 6840770012 "Transdisciplinary Biomedical Engineering Research II." funded by the Czech Ministry of Education and the grant 1ET101210513 "Relational Machine Learning for Analysis of Biomedical Data" funded by the Czech Academy of Sciences.

The STULONG study has been running at 2nd Dept. of Internal Medicine, 1st Faculty of Medicine of Charles University and University Hospital in Prague (head Prof. M. Aschermann), under the supervision of Prof. F. Boudik, with the collaboration of M. Tomeckova, PhD, and J. Bultas, PhD. Collected data was transferred to the electronic form by the European Centre of Medical Informatics, Statistics and Epidemiology of Charles University and Czech Academy of Sciences (head Prof. J.Zvarova).

References

[1] EUROMISE, Stulong – epidemiological study of atherosclerosis. Internet site address: http://euromise.vse.cz/challenge2004/index.html.

[2] Guil, F., Bosch, A., and Marin, R. TSET: Algorithm for mining frequent temporal patterns. In *Proc. of ECML/PKDD'04 Workshop on Knowledge Discovery in Data Streams - A Collaborative Effort in Knowledge Discovery*, pages 65–74, 2004.

[3] Kléma, J., Nováková, L., Karel, F., and Štěpánková, O. Trend analysis in stulong data. In *Proc. of ECML/PKDD'04 Discovery Challenge - A Collaborative Effort in Knowledge Discovery.* Prague: Univ. of Economics, 2004.

[4] Lavrac, N., Železný, F., and Flach, P. RSD: Relational subgroup discovery through first-order feature construction. In Matwin and Sammut, editors, *Proc. 12th Int. Conf. on Inductive Logic Programming*, 2002.

[5] Meger, N., Leschi, C., Lucas, N., and Rigotti, C. Mining episode rules in stulong dataset. In *Proc. of ECML/PKDD'04 Discovery Challenge - A Collaborative Effort in Knowledge Discovery.* Prague: Univ. of Economics, 2004.

[6] Železný, F. RSD user's manual. Available at: http://labe.felk.cvut.cz/ zelezny/rsd/rsd.pdf.