Rules mining as a next step of knowledge extraction from data

Agnieszka Nowak-Brzezińska and Roman Simiński

Department of Computer Science, Institute of Computer Science, Silesian University, Poland http://ii.us.edu.pl

Knowledge bases (KBs) are still one of the most popular methods of knowledge representation. Searching within a KB that consists of a large number of rules is an important data-mining task. There are various methods of inducing rules from data: association rules, decision trees algorithms or algorithms based on the rough set theory (RST)[2]. Unfortunately, the number of rules generated automatically from a large data set is too big, i.e. a data set[1] that contains only 195 of instances described by 24 attributes generates 4266 rules in the RSES software. Decreasing the number of instances to 50% of the original data set reduces the number of rules to 2136, while reducing the number of attributes to 50% produces a set with 1996 rules. This simply shows that quite a small data set usually generates a large number of rules and it is not possible for human (even if he/she is an expert in a given domain) to analyse such a rule set in a short time effectively. In such case, rules should be further analysed and preprocessed. There is a growing research interest in searching for methods that manage large sets of rules. Most of them use the clustering approach as well as joining and reducing rules[1, 3, 5]. The authors introduce a different idea, in which rules are divided into a number of groups based on a condition (similar premises and/or conclusion). The process is called *partition of rules*[2] and is conducted by so-called *partition strategies* PS. At the beginning, the knowledge base $\mathcal{R}$ that consists of n rules: $r_1, \ldots, r_i, \ldots, r_n$ is a single set without any order of rules. Each rule $r_i \in \mathcal{R}$ is stored as a Horn's clause defined as: $r_i : p_1 \wedge p_2 \wedge \ldots \wedge p_m \rightarrow c$, where m is the number of literals (pairs of attribute and its value) $a = v_i^a$. Attribute $a \in A$ may be a conclusion of rule r_i as well as a part of the premises. For every KB $\mathcal{R}$ with n rules the number of possible subsets is 2^n. Any arbitrarily created subset of rules $R \in 2^{\mathcal{R}}$ is called *partition of rules* and it can be generated by one of the many possible partition strategies. It is defined as follows $PR = \{R_1, R_2, \ldots, R_k\}$, where: k is the number of groups of rules formed by the partition PR, and R_j is j-th group of rules, $R \in 2^{\mathcal{R}}$ and $j = 1, \ldots, k$. The most general division of partition strategies talks about two types of strategies: *simple*[3] and *complex*[4].

[1] Parkinson data set from ML Repository

[2] The idea is new but it is based on the authors' previous research, where the idea of *clustering rules* as well as creating so-called *decision units* was introduced[3, 4].

[3] *Simple strategies* allocate every rule r_i to the proper group R_j, according to the value of the function $mc(r_i, R_j)$, i.e. the strategy of creating *decision partition.*

[4] *Complex strategies* usually do not generate the final partition in single step. It is defined by *a sequence of simple strategies* or *a combination of them*, or by *iteration*

In the authors' opinion, *partition of rules* leads to the improvement of the inference process efficiency [6]. The modification of the KB structure reduces the time of inference. It is crucial to assume that the user wants to get an answer from the system as soon as possible. Instead of searching within the full structure of rules (in case of traditional inference processes), only representatives ($Profiles$) of groups are compared with the set of facts F (and/or hypothesis to be proven). The most relevant group of rules is selected and the exhaustive searching is done only within a given group[5].

The idea of *rules partition* is implemented in the `kbExplorer` system[6], which is not limited to the inference optimization. The practical goal of the project is to create an expert system shell that allows for flexible switching between different inference methods based on knowledge engineer preferences[7].

Ack. This work is a part of the project ,,Exploration of rule knowledge bases" founded by the Polish National Science Centre (NCN: 2011/03/D/ST6/03027)

References

1. Mikołajczyk M., Reducing Number of Decision Rules by Joining, RSCTC2002, LNCS, vol. 2475, Springer Berlin Heidelberg, p.425-432, (2002).
2. Skowron A., Komorowski J., Pawlak Z., Polkowski L.: Rough Sets Perspective on Data and Knowledge, Handbook of Data Mining and Knowledge Discovery, Oxford University Press, Inc., New York, USA, (134-149), (2002).
3. Nowak-Brzezińska A., Wakulicz-Deja A.:The way of rules representation in composited knowledge bases, Advanced In Intelligent and Soft Computing, p. 175-182, Springer-Verlag, (2009).
4. Nowak-Brzezińska A., Simiński R.: New inference algorithms based on rules partition. CS&P Informatik-Berichte, vol.245, Humboldt-University. Germany, (2014).
5. Nalepa G., Ligeza A., Kaczor K.: Overview of Knowledge Formalization with XTT2 Rules, LNCS, vol. 6826, p. 329-336, Springer Verlag, (2011).
6. Grzymała-Busse J.W. :Managing Uncertainty in Expert Systems, Springer Science & Business Media, Vol. 143, (1991).

---

of a single simple strategy. An example of a complex strategy is the strategy of creating *centroids based partition*, which stems from the method of *cluster analysis* used for rules clustering.

[5] Experiments show that a KB that contains about 150 rules, the partition strategy based on clustering the conditional part of the rules led to a significant reduction of the percentage of the KB search during the inference process to about 4%. In case of partition strategies based on conclusions of the rules, in the optimistic case, we have to search only 2% of the whole KB (in the pessimistic case - 74%).

[6] http://kbexplorer.ii.us.edu.pl/

[7] The user has the possibility of creating KBs using a special creator or by importing a KB from a given data source. The format of the KBs enables working with a rule set generated automatically based on the RST theory as well as with rules given apriori by the domain expert. The KB can have one of the following file formats: XML, RSES, TXT. It is possible to define attributes of any type: nominal, discrete or continuous. There are no limits for the number of rules, attributes, facts or the length of the rule.