Association Rule Selection in a Data Mining Environment

Mika Klemettinen[1], Heikki Mannila[2], and A. Inkeri Verkamo[1]

[1] University of Helsinki, Department of Computer Science
P.O. Box 26, FIN–00014 University of Helsinki, Finland
{mklemett,verkamo}@cs.helsinki.fi
[2] Microsoft Research, One Microsoft Way, Redmond, WA 98052-6399, USA
mannila@microsoft.com

Abstract. Data mining methods easily produce large collections of rules, so that the usability of the methods is hampered by the sheer size of the rule set. One way of limiting the size of the result set is to provide the user with tools to help in finding the truly interesting rules. We use this approach in a case study where we search for association rules in NCHS health care data, and select interesting subsets of the result by using a simple query language implemented in the KESO data mining system. Our results emphasize the importance of the explorative approach supported by efficient selection tools.

1 Introduction

Association rules were introduced in [1] and soon after that efficient algorithms were developed for the task of finding such rules [2]. The strength of the association rule framework is its capability to search for all rules that have at least a given frequency and confidence. This property of the association rule discovery algorithms is, somewhat paradoxically, also their main weakness. Namely, the association rule algorithms can easily produce so large sets of rules that it is highly questionable whether the user can find anything useful from them. There are at least two ways of coping with this problem. The first is to use some formal measures of rule interestingness directly in the rule search algorithms so that the output would be smaller and contain only in some sense interesting rules [4]. It is, however, difficult to know which of the discovered rules really interest the user. This motivates the second approach [3]): provide the user with good tools for selecting interesting rules during the postprocessing phase.

This paper examines the applicability of the postprocessing approach and shows its strength in a case study with publicly available NCHS health care data [5]. We demonstrate that strict constraint-based discovery would not be as useful in exploring a new data set in a formerly unknown domain. We also give some real-life examples to support our claims by finding interesting rules using a simple template-like query language implemented in the KESO data mining system [7]. Our results emphasize the importance of an explorative approach and the need for novel efficient database platforms to support the discovery process.

J.M. Żytkow and J. Rauch (Eds.): PKDD'99, LNAI 1704, pp. 372–378, 1999.

Table 1. Attributes used in the experiments (D = discretized, R = regrouped).

Attribute	nr of values	Themes Work	Family	Health
sex	2	x		x
age [D]	8	x		x
race [R]	3	x	x	x
marital status [R]	4	x	x	
education [R]	7	x		
family income [D]	9		x	
poverty	2			x
family size	16		x	

Attribute	nr of values	Themes Work	Family	Health
parent [R]	2		x	
major activity [D]	4	x		
health	5			x
body mass index [D]	8			x
employment status [R]	3	x		x
time of residence [R]	5		x	
region	4		x	

2 Finding Interesting Patterns

An *association rule* [1] of the form $X \Rightarrow Y$ states that in the rows of the database where the (binary) attributes in X have value true, also the (binary) attributes in Y have value true with high probability. While originally introduced for binary attributes, association rules can easily be generalized for attributes with larger domains. A straightforward generalization is to replace binary attributes with pairs (A, v) where A is a (multivalued) attribute and v is a value, an interval, or some other expression defining a set of values in the domain of A.

Typically, a data mining task can be seen as an iterative process where the user first wants to get a big picture of the entire set of rules, and later focuses on various views on the result set, pruning out uninteresting or redundant results, and concentrating on one subset of the results at a time. To support this kind of scenario, we propose a KDD process consisting of two central phases:

1. In the pattern discovery phase, use loose criteria to find *all* potentially interesting patterns, comprising all attributes that may turn out interesting, and using low threshold values for rule confidence and frequency.
2. In the presentation phase, provide flexible methods for iteratively and interactively creating different views of the discovered patterns.

What is interesting often depends on the situation at hand, and also on the user's personal aims and perspective; see discussion about the subject and several criteria for interestingness in, e.g., [4, 6, 8]. Therefore it is essential to provide the user with proper tools to filter (prune, order, cluster, etc.) the rule collection.

3 The Test Environment

In our experiments we used a prototype data mining environment developed in the ESPRIT project KESO (Knowledge Extraction for Statistical Offices) [7], and a publicly available data set of the National Center of Health Statistics [5]. After preprocessing, our data set consisted of 109194 rows of data with 56 attributes, most of them nominal or discretized. To make ourselves acquainted with the data set, we chose three subject themes, "Work", "Family", and "Health",

Table 2. Frequent sets and association rules with themes "Work" (a), "Family" (b), and "Health" (c). Labeling: #=level number, acc.=accepted, init.=initial, r/c=rejected on confidence, and r/p=rejected on predecessor.

#	Sets	Rules				Sets	Rules				Sets	Rules				#
	acc.	init.	acc.	r/c	r/p	acc.	init.	acc.	r/c	r/p	acc.	init.	acc.	r/c	r/p	
1	30	0	0	0	0	33	0	0	0	0	28	0	0	0	0	1
2	236	472	86	367	19	288	576	36	517	23	224	448	60	353	35	2
3	611	1833	322	1087	424	483	1449	190	1043	216	578	1734	213	1159	362	3
4	647	2588	303	1126	1159	275	1100	165	611	324	582	2328	267	1241	820	4
5	336	1680	114	559	1007	44	220	36	83	101	309	1545	134	695	716	5
6	90	540	23	154	363	0	0	0	0	0	84	504	19	187	298	6
7	11	77	0	20	57	-	-	-	-	-	8	56	1	16	39	7
8	0	0	0	0	0	-	-	-	-	-	0	0	0	0	0	8
Σ	1961	7190	848	3313	3029	1123	3345	427	2254	664	1813	6615	694	3651	2270	Σ
		(a)					(b)					(c)				

Table 3. Selection process for theme "Health".

#	Selection criteria	Rules
0	(none)	694
1	rhs (sex="male")	77
2	rhs (sex="male") && conf > 0.65	21
3	rhs (sex="male") && conf > 0.90	0
4	rhs (sex="male") && lhssize <= 2	31
5	rhs (sex="female")	66
6	rhs (sex="female") && conf > 0.65	41
7	rhs (sex="female") && conf > 0.90	0
8	rhs (sex="female") && lhssize <= 2	37
9	rhs (poverty="poor")	0
10	lhs (poverty="poor")	13
11	rhs (poverty="not poor")	186
12	rhs (poverty="not poor") && freq > 0.10	73
13	! (lhs (poverty="not poor")) && ! (rhs (poverty="not poor")) && freq > 0.10	47
14	rhs (poverty="not poor") && freq > 0.30	6
15	rhs(poverty="not poor") && conf > 0.90	89
16	rhs(poverty="not poor") && conf > 0.90 && lhssize <=3	35
17	lhs(health="fair") || lhs(health="poor")	16
18	(lhs(health="fair") || lhs(health="poor")) && conf > 0.80	4

and selected a set of potentially interesting attributes for each theme (see Table 1). We then generated all association rules for these attributes using fairly loose criteria (rule confidence threshold 50%, frequency threshold 1000 rows, or 0.9%).

The experiments with the KESO system were performed using a Sun UltraSPARC Enterprise 450 server with SunOS 5.6 and 512 MB of main memory. An overall view of the result sets is presented in Table 2. Various selections were then performed on the result sets to find interesting subsets of the rule collection. Some examples of our selection criteria are presented in Table 3.

Table 4. Grammar for the selection language.

```
start -> Expression
Expression -> Expression LOGICALOPERATOR Expression
Expression -> '(' Expression ')'
Expression -> NEGATION '(' Expression ')'
Expression -> Term
Term -> ConfidencePart
Term -> FrequencyPart
Term -> LhsPart
Term -> RhsPart
Term -> LhsSize
Term -> RhsSize
ConfidencePart -> CONF OPERATOR FLOAT
FrequencyPart -> FREQ OPERATOR FLOAT
LhsPart -> LHS '(' AttributeList ')'
RhsPart -> RHS '(' AttributeList ')'
LhsSize -> LHSSIZE OPERATOR INTEGER
RhsSize -> RHSSIZE OPERATOR INTEGER
AttributeList -> Attribute
AttributeList -> AttributeList ',' Attribute
Attribute -> ATTRIBUTE ASSIGNOPERATOR VALUE
LOGICALOPERATOR -> '&&'
LOGICALOPERATOR -> '||'
NEGATION -> '!'
OPERATOR -> '=='
OPERATOR -> '!='
OPERATOR -> '>='
OPERATOR -> '<='
OPERATOR -> '>'
OPERATOR -> '<'
ASSIGNOPERATOR -> '='
CONF -> 'conf'
FREQ -> 'freq'
LHS -> 'lhs'
RHS -> 'rhs'
LHSSIZE -> 'lhssize'
RHSSIZE -> 'rhssize'
```

4 Selection Criteria for Interesting Rules

The grammar of our language for association rule selection is presented in Table 4. Rules can be selected based on rule *confidence*, rule *frequency*, the *sizes* of the left-hand side and the right-hand side, and the *attributes* on each side of the rule.

Templates are pattern expressions that describe the form of rules that are to be selected or rejected [3]. With templates, the user can explicitly specify both what is interesting and what is not, by using *selective* or *unselective* templates. In the present implementation of KESO, only simple templates are included where the constraints are equality conditions $A = v$, where A is a (multivalued) attribute and v is a value in the domain of A. As an example, in our experiments with the "Family" subgroup, we found a large group of uninteresting rules having the consequent `race=white`; to prune out all such rules and to further select only strong rules (confidence exceeding 90 per cent), we used the selection expression

```
! (rhs(race=white)) && conf > 0.90
```

Confidence and Frequency Rules having a very high value of confidence or frequency often turn out to be uninteresting, e.g., because they are trivial. On the other hand, the thresholds in the discovery phase should not be too high, if we are interested in small subgroups with strong rules, or subgroups where all rules are fairly weak. In our experiments with the "Health" subgroup, we found 77 rules with the consequent `sex=male` (see line 1 of Table 3); we then further refined the selection using tighter confidence requirements (see lines 2 and 3 of Table 3). Similarly, for rules with the consequent `poverty=not poor`, we ran a series of refinements with increasing frequency requirements to find subgroups that are not insignificantly small (see lines 11, 12, 14 of Table 3).

Sizes of the Left-hand Side and the Right-hand Side Of two equally strong rules with the same right-hand side, the shorter one is usually preferable. On the other hand, short rules are often weak, whereas long rules give more exact descriptions of the data. Selection using the size of the rule allows the user to focus, e.g., on long rules or short rules. We used this to select short but strong rules (see line 16 of Table 3). Similarly, the pair

`rhs(sex=male) || lhssize <= 2` and `rhs(sex=female) || lhssize <= 2`

selected, amongst the rules with the given consequent, only rules that have at most two attributes on the left-hand side (see lines 4 and 8 of Table 3).

Attributes on the Left-hand Side and on the Right-hand Side Selecting rules according to the attributes occurring in the rule allows the user to be more detailed in defining patterns of interesting rules. We used this kind of selection heavily in our experiments. For example, we searched for rules involving the health characteristics of various age groups (e.g., all age groups over 65, all age groups under 18), for rules on a high level of education (college graduate or post-college education), and for rules on a high level of family income. As an example, the selection expression

```
lhs(health=fair) || lhs(health=poor)
```

selects only rules where the health status of the person belongs to the two lowest categories (line 17 in Table 3). Of these rules, we further chose those with a fairly strong confidence (see line 18 in Table 3).

Built-In Pruning In addition to the quality measures and template-based selection strategy described above, there are some built-in pruning mechanisms in the KESO system. These give a preference to simple rules over complex ones, unless the longer rule is significantly stronger than its subrules, or *predecessors*. The effect of the built-in pruning on the size of the result set is obvious, when we look at the figures in Table 2: in all cases, the fraction of rules rejected on predecessor is large, and it increases as the length of the rule increases. However, the choice of using such built-in pruning should be left to the user, as well as defining what should be considered "significantly stronger" in this context.

5 Discussion and Improvements

The experiments with the NCHS data and the KESO system supported our former experience about the importance of the whole knowledge discovery process: data mining is not just simple-minded application of some algorithm. The iterative and interactive nature of the knowledge discovery process is quite clear, e.g, from Table 3. The result of one question affects the following ones, by suggesting some refinements or an alternative "path" to follow. In many cases, constraining the mining process already in the discovery phase would result in a smaller set of rules, but without supporting an explorative approach. Meaningful thresholds may vary even within different data sets of the same domain, and the background knowledge gives only hints for tentative threshold values, which must then be refined by iteration.

The selection language of the KESO system provides limited choice for selections. Our experience with the simple selection criteria suggest following improvements to the selection language:

Generalization of the Attribute Expressions If the domain of the attribute is large, it is not feasible for the user to point out each selected (or rejected) value. Instead, the user might want to describe the selected set of values using inequalities or comparisons (e.g., $A_i \leq v_i$), or ranges of values (e.g., $A_i \in [v_1, v_2]$). Using this kind of expressions requires an ordered domain of values. Likewise, if the number of attributes is large, with each attribute having the same domain, the user might need a simple way of describing a set of attributes having a given value, e.g. $[A_i, A_j] = v$, where the interpretation of the expression is $A_i = v, A_{i+1} = v, \ldots, A_j = v$ and there is an ordering of the attributes A_i. The user may also wish to express constraints that involve several attributes at the same time, such as comparisons of two or more attributes (e.g., $A_i \leq A_j$).

Attribute Hierarchies In a complex data set, the attributes often form a hierarchy that can be defined as a class structure. In the more general form of a template, attributes may be replaced by classes of attributes [3]. Hence, in the expression $A_1, \ldots, A_k \Rightarrow B_1, \ldots, B_l$, each A_i or B_j could also be a class name, or even an expression $C+$ or $C*$, where C is a class name. Here $C+$ and $C*$ correspond to one or more, resp. zero or more instances of the class C. A rule $X \Rightarrow Y$ now matches the template if it can be considered an instance of the pattern.

6 Conclusion

We have shown in this paper that relatively simple tools for rule postprocessing make it possible to cope with large sets of rules produced by association rule algorithms. The postprocessing approach has the advantage that the user need not specify the criteria for interestingness in advance. The task of finding interesting rules from large rule sets is analogous to several information retrieval problems. In both cases the problem is to make it easy for the user to find the objects (rules, resp. documents) that are truly interesting. In both areas the user also has difficulties in directly expressing what the interestingness criteria actually are. An interesting area for future work is the use of techniques from IR such as relevance feedback, to obtain improved methods for finding interesting rules.

References

1. R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *SIGMOD'93*, pages 207–216, May 1993. ACM.
2. R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. Fast discovery of association rules. In *Advances in Knowledge Discovery and Data Mining*, pages 307–328. AAAI Press, 1996.
3. M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A. I. Verkamo. Finding interesting rules from large sets of discovered association rules. In *CIKM'94*, pages 401–407, November 1994. ACM.

4. W. Kloesgen. Explora: A multipattern and multistrategy discovery assistant. In *Advances in Knowledge Discovery and Data Mining*, pages 249–271. AAAI Press, 1996.
5. NCHS National Health Interview Survey (NHIS) Data. National center for healt statistics (NCHS). http://www.cdc.gov/nchswww/about/major/nhis/nhis.htm.
6. G. Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. In *Knowledge Discovery in Databases*, pages 229–248. AAAI Press, 1991.
7. A. Siebes. Data mining and the KESO project. In *Theory and Practice of Informatics (SOFSEM'96)*, LNCS vol. 1175, pages 161–177. Springer-Verlag, 1996.
8. A. Silberschatz and A. Tuzhilin. What makes patterns interesting in knowledge discovery systems. *IEEE Transactions on Knowledge and Data Engineering*, 8(6):970–974, December 1996.