Reducing Number of Decision Rules by Joining

Michał Mikołajczyk

Institute of Mathematics, Warsaw University
ul. Banacha 2, 02–097 Warsaw, Poland
M.Mikolajczyk@mimuw.edu.pl

Abstract. Sets of decision rules induced from data can often be very large. Such sets of rules cannot be processed efficiently. Moreover, too many rules may lead to overfitting. The number of rules can be reduced by methods like Quality-Based Filtering [1,10] returning a subset of all rules. However, such methods may produce decision models unable to match many new objects. In this paper we present a solution for reducing the number of rules by joining rules from some clusters. This leads to a smaller number of more general rules.

1 Introduction

Classical decision models based on decision rules induced from data often consist of huge amount of decision rules. However, such large sets of rules cannot be effectively processed, e.g., in matching of new objects. Moreover, too many rules may lead to overfitting. Reducing the number of decision rules is also important for experts analyzing the induced rules. There is a growing research interest in searching for clustering methods or reduction methods of induced from data decision or association rule sets (see, e.g., [1,3,6,7,8,15]).

There is a need to develop methods for inducing set of rules of feasible sizes or methods for decision rule pruning without decreasing the classification quality.

The number of rules can be reduced by methods like Quality-based Filtering returning a subset of all rules. Such methods, however, may produce decision models unable to recognize many new objects.

We present a method for reduction of the number of rules by joining rules from some clusters. This leads to a smaller number of more universal decision rules. We also present experimental results showing that it is possible to induce sets of rules of a feasible size without decreasing the classification quality.

Let us assume data are represented in decision tables [11] of the form presented below.

–	$a_1 \; \ldots \; a_N$	d
x_1		d_1
$\vdots$		$\vdots$
x_k		d_k

where in rows are described objects $x_i \in \mathbb{X}$ by means of attributes $a_1, \ldots, a_N$, and d denotes the decision attribute with values d_i corresponding to decision

J.J. Alpigini et al. (Eds.): RSCTC 2002, LNAI 2475, pp. 425–432, 2002.

classes. From such data table one can generate all (minimal) decision rules [11]. Calculated rules have the form of conjunction, for example: $r : (a_1 = 1 \wedge a_3 = 4 \wedge a_7 = 2) \Rightarrow d^r$. If the conjunction is satisfied by the object x, then the rule classifies x to the decision class d^r $(r(x) = d^r)$. If the conjunction is not satisfied by x, then the rule for x is not applicable what is expressed by the answer $d^?$ $(r(x) = d^?)$.

One can represent decision rules by means of sequences

$$\left\{r^A(i)\right\}_{i=1}^{N} \text{ or by } r : (a_1 = r^A(1) \wedge \cdots \wedge a_N = r^A(N)) \Rightarrow d$$

where: $r^A(i) =$ *attribute value* or "$\star$" if an attribute value can be arbitrary.

New objects can be classified by means of voting strategies [5] using decisions returned by rules matching the objects. Rules can have assigned weights describing their importance in voting. A very important question is how to rate the quality of the rules and how to calculate the weights for voting. There are some partial solutions for these problems [2].

One can induce all (minimal) rules from decision table [11]. Unfortunately, we can get too many rules and then new objects can not be classified efficiently. The other important reason why we want to have fewer rules is their readability [13]. If a large set of rules is generated then we are unable to understand what they describe. Small number of rules is easier to analyze and understand by experts. This is very important when we want to explore an unknown phenomenon from data.

One can reduce the number of rules by selecting a subset of the set of all rules (see, e.g., Quality-Based Filtering [1,10]). In this way one can obtain a small number of rules. However, it is necessary to take care about the quality of classification of new objects [14]. After pruning of many rules from a given set it is highly probable that many new objects will not be recognized because they do not match any decision rule.

One can group (cluster) rules and then use hierarchical classification. This makes possible to reduce the time needed for classification, but it can be still hard to analyze such clusters by human being. Moreover, the process of rule's grouping is also a very difficult problem. The results of clustering give us extra information about the set of rules and dependencies between them. Unfortunately, no universal solution for such task is known.

2 System of Representatives

We would like to propose another solution based on joining the rules from some clusters. By joining we can produce a smaller set of more general rules. Such rules are able to recognize every object that was classified by the source rules. They can also recognize many more unseen objects. It is important to note that the joining should be performed on rules with similar logical structure. Therefore, first we will group the rules into clusters using some similarity measures and next we join the clusters to more general rules. System of Representatives consists of the set of induced generalized decision rules. Figure 1 presents all stages necessary to build the System of Representatives.

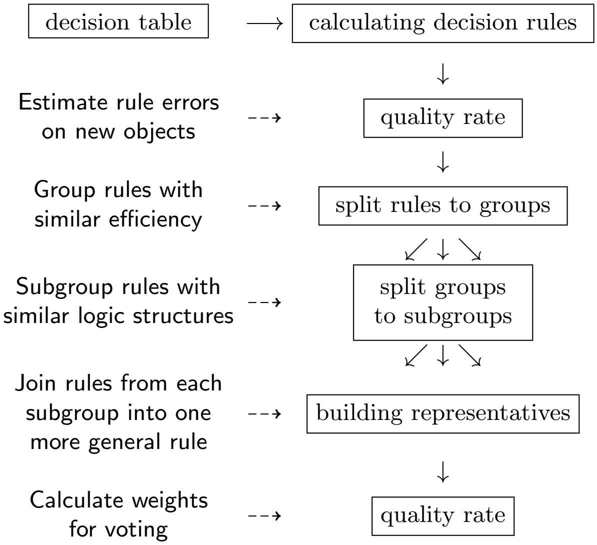
decision table
calculating decision rules
Estimate rule errors on new objects
quality rate
Group rules with similar efficiency
split rules to groups
Subgroup rules with similar logic structures
split groups to subgroups
Join rules from each subgroup into one more general rule
building representatives
Calculate weights for voting
quality rate

Fig. 1. System of Representatives construction

2.1 Rule Induction

We do not have to use the whole decision table for rule induction. A part of the table can be used for rule generation and the remaining part makes possible to estimate the quality of generated rules.

Decision rules, which are too detailed, are difficult to join. One can overcome this drawback by shortening rules (from generated rules we try to drop some conjuncts from the left part of the rule; see, e.g., [5]).

2.2 Splitting the Set of Decision Rules into Groups

Splitting a given set of decision rules into clusters should guarantee that rules joined (into a generalized rule) from the same cluster are of similar quality. It is very important, because a rule of high quality joined with a rule of low quality can give a weak rule, i.e., rule making many mistakes during the classification of objects. Hence we should try to join high quality rules with high quality rules and low quality rules ones with low quality rules. First we split rules into classes of rules with the same decision. Next, each class is split into clusters using some clustering methods like the standard k-means algorithm [4]. Below we present an exemplary set of the parameters used for splitting of a given set of rules $\mathbb{R} = \{r_1, \ldots, r_n\}$ into groups:

- a function which determines a weight for any rule [2,14], for example:

$$Q(r) = \text{card}\left(\{x \in \mathbb{X} \ : \ r(x) = d^r \text{ and the proper decision on } x \text{ is } d^r\}\right) - \text{card}\left(\{x \in \mathbb{X} \ : \ r(x) = d^? \text{ and the proper decision on } x \text{ is } d^r\}\right)$$

- a distance function between rules, for example: $d(r_1, r_2) = |Q(r_1) - Q(r_2)|$.

Similarly to splitting into groups, one can split groups into subgroups using the standard k-means algorithm [4], where the distance function used in our experiments is the following one:

$$d_L(r_1, r_2) = N \cdot |s_\star(r_1) - s_\star(r_2)| + \sum_{i=1}^{N} d_{L0}(r_1^A(i), r_2^A(i))$$

where: $s_\star(r) = \text{card}\left(\{i : r^A(i) = \star\}\right)$ and:

$$d_{L0}(a, b) = \begin{cases} 0 & \text{if } a = b \\ 1 & \text{if } a \neq b \wedge a \neq \star \wedge b \neq \star \\ 2 & \text{if } a \neq b \wedge (a = \star \vee b = \star) \end{cases}$$

Observe that $s_\star(r)$ is equal to the number of free attributes in rule r, and $\sum_{i=1}^{N} d_{L0}(r_1^A(i), r_2^A(i))$ expresses the degree of similarity of logical structures of r_1 and r_2.

Using the above distance function we guarantee that rules with similar logical structures are joined into the same clusters. The joining of rules with different logical structures into one more general rule could cause many errors in classification of new objects.

2.3 Joining Rules

After joining rules from any constructed cluster we obtain one rule called a *representative*:

$$R\text{: } R^A(1) \ \ R^A(2) \ \ R^A(3) \ \ \ldots \ \ R^A(N) \Rightarrow d$$

where $R^A(i) = \{r_1^A(i), \cdots, r_n^A(i)\}$ for $i = 1, \cdots, N$, N is the number of attributes, and $r_1, \ldots, r_n$ are joined rules from a given cluster.

We use one more parameter for the representative R, i.e., the maximal number W_{max}^R of free attributes ($s_\star(r)$) in joined rules: $W_{max}^R = \max_{i=1,\ldots,n}\{s_\star(r_i)\}$. This number is used in the matching of objects by generalized rules. Without this parameter the rules are too general, which decreases the quality of classification by such rules.

Matching Objects by Representatives - R(x)

```
1. Let $W_A := W^R_{max}$.
2. Let $i = 1$.
3. If $a_i(x) \in R^A(i)$ then go to Step 5.
4. If $\star \in R^A(i)$ then $W_A := W_A - 1$,
   otherwise STOP and return $R(x) = d^?$.
5. If $i < N$ then $i := i + 1$ and go back to Step 3.
6. STOP and return $R(x) = \begin{cases} d^R & \text{if } W_A \geqslant 0 \\ d^? & \text{if } W_A < 0 \end{cases}$.
```

Let us consider an example. Rules to be joined are the following ones:

$$r_1\colon 1\ 3 \star 1 \star \star 2 \Rightarrow d \qquad r_2\colon 2\ 3 \star \star 1 \star 2 \Rightarrow d \qquad r_3\colon 5\ 3 \star 1\ 1 \star 4 \Rightarrow d$$

After joining we obtain a representative:

$$\text{R: } \{1,2,5\}\ 3 \star \{1,\star\}\ \{\star,1\} \star \{2,4\} \Rightarrow d\ \ ; W^R_{max} = 3$$

Now let us take some objects:

x_1: 1 3 3 1 2 3 2
x_2: 2 3 1 2 1 5 2
x_3: 2 3 6 1 1 3 4
x_4: 5 3 4 1 5 3 2
x_5: 6 3 4 1 5 3 2
x_6: 5 3 4 1 5 3 3
x_7: 5 1 4 1 5 3 2
x_8: 5 3 9 9 9 9 2

Objects x_1, x_2, x_3, x_4 are classified by representative R to the decision class corresponding to d^R. Objects x_5, x_6, x_7, x_8 are not recognized by R and the representative R will return for them the answer $d^?$.

By bounding to W^R_{max} the maximal number of free attributes we do not make the rules too general. This makes possible to obtain the high classification quality of the induced representatives.

2.4 Algorithm Parameters and Complexity

The following parameters are used in tuning of the System of Representatives: (i) a parameter describing a fraction of the number of objects from training set used for inducing rules to the number of those used for the estimation of classification quality of induced rules, (ii) a parameter describing the acceptable error in shortening rules, (iii) all parameters used for splitting into groups, (iv) all parameters used for splitting into subgroups.

Experiments have shown that the System of Representatives can be tuned for any tested data.

Observe that the computational complexity of the k-means algorithm is quadratic. However, the System of Representatives is built only once and next it can be used many times.

Table 1. Decision tables used in all experiments

Decision table name	Number of examples	Number of conditional attributes	Number of possible decisions	Average number of attributes value
Austra0	690	14	2	83.4
Austra1	345	14	2	55.0
Austra2	345	14	2	56.8
Diab0	768	8	2	156.4
Diab1	384	8	2	111.9
Diab2	384	8	2	111.5
Heart0	270	13	2	29.5
Heart1	135	13	2	22.6
Heart2	135	13	2	22.8
Irys	120	4	3	29.0
Kan 1	84	16	17	8.1
Kan 12	84	17	15	8.0
Kan 2	84	16	14	7.5
Kan 21	84	17	15	8.1
Lymn0	148	18	4	3.3
Lymn1	74	18	4	3.2
Lymn2	74	18	4	3.3
Monk1dat	124	6	2	2.8
Monk1tes	432	6	2	2.8
Monk2dat	169	6	2	2.8
Monk2tes	432	6	2	2.8
Monk3dat	122	6	2	2.8
Monk3tes	432	6	2	2.8
Tttt	615	8	2	140.1

3 Results of Experiments

3.1 Tested Data

In this section we present the results of experiments. All tests were made using the CV5 (Cross Validation) method [9]. In experiments we used twenty-four decision tables presented in Table 1. Most of the data tables are from UCI Machine Learning Repository [9] http://www.ics.uci.edu/ mlearn/MLRepository.html.

3.2 System of Representatives

Now we are ready to present the results of experiments obtained using our parameterized System of Representatives. For each table all parameters have been tuned experimentally. The results are summarized in Table 2.

3.3 The Discussion of the Results

Let us look at the averaged results presented in Table 3. From Table 3 one can observe that the strategy shortening of rules improves classification quality and reduces the number of rules. Therefore, we used this strategy in the System of Representatives. Additional experiments showed that this improves significantly the results.

Table 2. Results obtained by means of the proposed System of Representatives

Decision	Objects identified			Number	Average time
table	correctly	wrongly	not recognized	of rules	of classification
Austra0	85.51%	14.49%	0.00%	24.80	0.01 s
Austra1	80.00%	20.00%	0.00%	30.60	< 0.01 s
Austra2	87.25%	12.75%	0.00%	26.40	< 0.01 s
Diab0	66.02%	33.98%	0.00%	146.60	< 0.01 s
Diab1	62.76%	37.24%	0.00%	90.40	< 0.01 s
Diab2	67.45%	32.55%	0.00%	77.80	0.02 s
Heart0	83.70%	16.30%	0.00%	13.40	< 0.01 s
Heart1	80.00%	20.00%	0.00%	15.40	< 0.01 s
Heart2	75.56%	24.44%	0.00%	11.00	< 0.01 s
Irys	92.50%	7.50%	0.00%	24.80	0.07 s
Kan 1	14.29%	83.33%	2.38%	150.00	< 0.01 s
Kan 12	39.29%	60.71%	0.00%	116.60	0.01 s
Kan 2	40.48%	59.52%	0.00%	100.00	< 0.01 s
Kan 21	33.33%	66.67%	0.00%	127.40	< 0.01 s
Lymn0	81.76%	18.24%	0.00%	49.40	0.16 s
Lymn1	81.08%	18.92%	0.00%	44.00	< 0.01 s
Lymn2	83.78%	16.22%	0.00%	30.00	0.06 s
Monk1dat	90.32%	9.68%	0.00%	59.00	< 0.01 s
Monk1tes	100.00%	0.00%	0.00%	49.40	0.01 s
Monk2dat	68.64%	28.40%	2.96%	67.20	< 0.01 s
Monk2tes	66.20%	33.80%	0.00%	22.60	< 0.01 s
Monk3dat	93.44%	6.56%	0.00%	27.60	< 0.01 s
Monk3tes	97.22%	2.78%	0.00%	13.20	< 0.01 s
Tttt	65.69%	34.31%	0.00%	45.40	< 0.01 s
Average	72.34%	27.43%	0.22%	56.79	0.02 s

Table 3. Averaged results of presented systems calculated on twenty-four tables, where: *kNN* – k Nearest Neighbour system with $k = 1$, *Classical std.* – classical rule system [12], *Classical ext.* – classical rule system with rule shortening [12] , *QbF std.* – Quality-based Filtering system, *QbF ext.* – Quality-based Filtering system with rule shortening, *Rep. System* – proposed System of Representatives

Decision	Objects identified			Number	Average time
table	correctly	wrongly	not recognized	of rules	of classification
kNN	70.04%	29.86%	0.10%	0.00	0.06 s
Classical std.	69.56%	30.36%	0.08%	2772.55	0.06 s
Classical ext.	71.08%	28.91%	0.01%	2135.53	0.05 s
QbF std.	60.16%	17.38%	22.47%	347.37	0.10 s
QbF ext.	64.08%	18.33%	17.58%	23.51	0.08 s
Rep. System	72.34%	27.43%	0.22%	56.79	0.02 s

System of Representatives, like classical rule systems [12], makes more mistakes classifying objects than Quality-based Filtering systems. However, it recognizes correctly more objects. System of Representatives leads to better classification quality than classical systems and to smaller number of decision rules. Time needed for the classification of new objects is very short.

4 Conclusions

The results presented in the paper show that it is possible to reduce the number of decision rules without decreasing the classification quality. Moreover, we obtain less, more general rules. Hence, the time needed for classification of new objects is very short.

Unfortunately, it takes time to build and tune the System of Representatives. However, by tuning of parameters the high quality System of Representatives can be induced for various data. Once tuned and trained, the system exhibits excellent efficiency and the very good classification quality.

System of Representatives consists of relatively small number of rules and in consequence the size of decision model built using such rules is substantially smaller that in case of not generalized rules. Hence, using the Minimal Description Length Principle [13] one can expect that such models will be characterized by higher quality of classification than the traditional decision models.

Acknowledgments. I would like to thank professor Andrzej Skowron for his support while writing this paper. This work was partially supported by the Polish State Committee for Scientific Research grant No. 8T11C02519.

References

1. T. Ågotnes, Filtering large propositional rule sets while retaining classifier performance. Department of Computer and Information Science, Norwegian University of Science and Technology, 1999
2. I. Bruha, Quality of decision rules. Machine Learning and Statistics. The Interface, chapter 5, 1997
3. P. Gago, C. Bento, A metric for selection of the most promising rules. In PKDD, pages 19–27, 1998
4. A. K. Jain and R. C. Dubes, Algorithms for Clustering Data. Englewood Cliffs, New Jersey: Prentice Hall, 1988
5. Jan Komorowski and Zdzisław Pawlak and Lech Polkowski and Andrzej Skowron, Rough Fuzzy Hybridization. A New Trend in Decision Making. Springer-Verlag, pages 3–98, 1999
6. B. Lent, A. N. Swami, J. Widom, Clustering association rules, In ICDE, pages 220–231, 1997
7. B. Liu, W. Hsu, Y. Ma, Prunig and summarization of descovered associations, In SIGKDD, pages 125–134, 1999
8. B. Liu, M. Hu, W. Hsu, Multi-level organization and summarization of discovered rules, In SIGKDD, pages 208–217, 2000
9. Michell T.M.: Machine Learning. Mc Graw-Hill, Portland, 1997.
10. A. Øhrn, L. Ohno-Machado, T. Rowland, Building manageable rough set classifiers. AMIA Annual Fall Symposium, pages 543–547, Orlando, USA, 1998
11. Z. Pawlak, Rough sets – Theoretical aspects of reasoning about data. Kluwer Academic Publishers, Dordrecht, 1991
12. RSES system: alfa.mimuw.edu.pl
13. J. Rissanen, Modeling by the shortest data description. Authomatica 14, pages 465–471, 1978
14. J. A. Swets, Measuring the accuracy of diagnostic systems. Science, 240: 1285–1293, 1988
15. H. Toivonen, M. Klementinen, P. Ronkainen, K. Hätönen, H. Manila, Pruning and grouping discovered association rules. In ML Net Familiarization Workshop on Statistics, Ml and KDD, pages 47–52, 1995