Lazy classification method based on Boolean reasoning approach

Hung Son Nguyen[1] and Sinh Hoa Nguyen[2]

[1] Institute of Mathematics, Warsaw University,
Banacha 2, Warsaw 02-097, Poland
E-mail: son@mimuw.edu.pl
[2] Polish-Japanese Institute of Information Technology
ul. Koszykowa 86, Warsaw 02-008, Poland
E-mail: hoa@mimuw.edu.pl

1 Introduction and motivations

Classification of new unseen objects is a most important task in data mining. There are many classification approaches likes "nearest neighbors", "naive Bayes", "decision tree", "decision rule set", "neural networks" etc. Every classification method has some advantages and disadvantages, hence the choice of classification methods in practical data mining applications depends on different criteria like: accuracy, description clearness, time and memory complexity etc.

Almost all methods based on rough sets use rule set classification approach (see e.g., [3,10,11,12]), which consists of two steps: generalization and specification. In generalization step, some decision rule set is constructed from data as a knowledge base. In specialization step the set of such rules, that match a new object (to be classified) is selected and a conflict resolving mechanism will be employed to make decision for new unseen objects.

Unfortunately, there are opinions that rough set based methods can be used for small data set only. The main reproach is related to their lack of scalability and low efficiency of realization in case of very large data tables. The biggest troubles stick in rule induction step. As we know, the potential number of all rules is exponential. All heuristics for rule induction algorithms have at least $O(n^2)$ time complexity, where n is the number of objects in the data set. Moreover, the existing algorithms require multiple data scanning. In case of large decision table, the data must be held in a database system and the main problem is to minimize the number SQL queries used in the algorithm.

One of most interesting approaches based on Rough set theory is related to *minimal consistent decision rules*. Given a decision table $\mathbb{A} = (U, A \cup \{dec\})$, the decision rule:

$$\mathbf{r} =_{def} (a_{i_1} = v_1) \wedge ... \wedge (a_{i_m} = v_m) \Rightarrow (dec = k)$$

is called minimal consistent decision rule if it is consistent with $\mathbb{A}$ and any decision rule $\mathbf{r}'$ created from $\mathbf{r}$ by removing one of descriptors from left hand side of $\mathbf{r}$ is not consistent with $\mathbb{A}$. The set of all minimal consistent decision rules for a given decision table $\mathbb{A}$ is denoted by $MinConsRules(\mathbb{A})$.

1. Learning phase: generates a set of decision rules $RULES(\mathbb{A})$ (satisfying some predefined conditions) from a given decision table $\mathbb{A}$.
2. Rule selection phase: selects from $RULES(\mathbb{A})$ the set of such rules that can be supported by x. We denote this set by $MatchRules(\mathbb{A}, x)$.
3. Post-processing phase: makes a decision for x using some voting algorithm for decision rules from $MatchRules(\mathbb{A}, x)$

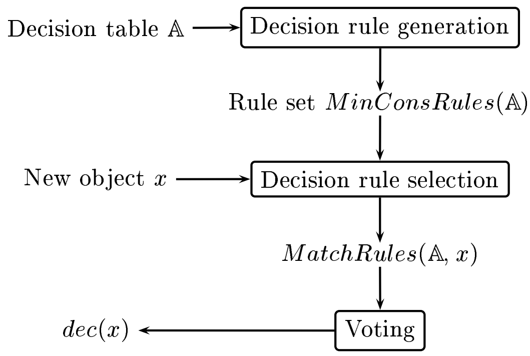

Fig. 1. The Rule base classification system

The set of all minimal consistent decision rules can be found by computing *object oriented reducts* (or local reducts) [4,3,11]. We recall the method based on boolean reasoning approach for computing of such reducts [4,9]. Let $Var = \{\alpha_1, ..., \alpha_k\}$ be a set of boolean variables corresponding to attributes $a_1, ..., a_k$ from A. Let $u, v \in U$ are objects from U. One can define the discernibility function for u, v as follows:

$$disc_{u,v}(\alpha_1, ..., \alpha_k) = \bigvee \{\alpha_i : a_i(u) \neq a_i(v)\}$$

Let $u \in U$ be a arbitrary object in decision table $\mathbb{A} = (U, A \cup \{dec\})$. We can define a function $f_u(d_1, ..., d_k)$ called *discernibility function for* u as follows:

$$f_u(\alpha_1, ..., \alpha_k) = \bigwedge_{dec(u) \neq dec(v)} disc_{u,v}(\alpha_1, ..., \alpha_k)$$

Every prime implicant of f_u corresponds to "local reduct" for object u and such reducts are associated with a minimal consistent decision rules[4,3]. We denote by $MinRules(u)$ the set of all minimal consistent decision rules created from boolean function f_u. One can show that

$$MinConsRules(\mathbb{A}) = \bigcup_{u \in U} MinRules(u)$$

The set $MinConsRules(\mathbb{A})$ can be used as a knowledge base in classification systems (see Figure 1). In practice, instead of $MinConsRules(\mathbb{A})$, we can use the set of short, strong, and high accuracy decision rules defined by:

$$MinRules(\mathbb{A}, \lambda_{\max}, \sigma_{\min}, \alpha_{\min}) = \left\{ \begin{array}{l} \mathbf{r}\text{: } \mathbf{r} \text{ is minimal } \wedge length(\mathbf{r}) \leq \lambda_{\max} \wedge \\ support(\mathbf{r}) \geq \sigma_{\min} \wedge confidence(\mathbf{r}) \geq \alpha_{\min} \end{array} \right\}$$

All heuristics for object oriented reducts can be modified to extract decision rules from $MinRules(\mathbb{A}, \lambda_{\max}, \sigma_{\min}, \alpha_{\min})$.

We show that classification method presented in Figure 1 based on local reducts can be modified by lazy learning algorithms that make them more scalable. The lazy rule-based classification diagram is presented in Figure 2

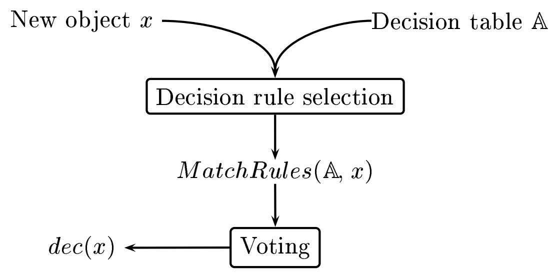

Fig. 2. The lazy rule-based classification system

In other words, we will try to extract the set of decision rules that match object x directly from data without learning process. The large decision table must be held in a data base system and the main problem is to minimize the number SQL queries used by algorithm. We show that this diagram can work for the classification method described above using the set $MinRules(\mathbb{A}, \lambda_{\max}, \sigma_{\min}, \alpha_{\min})$ of decision rules.

The proposed algorithm of searching for decision rules from $MinRules(\mathbb{A}, \lambda_{\max}, \sigma_{\min}, \alpha_{\min})$ that match the new object x is based on Apriori algorithm. We assume that the decision table $\mathbb{A}$ is stored in some database and the access to data is possible by using SQL queries. We will optimize the number of I/O operations to the database.

In this paper we propose to adopt lazy learning idea to make rough set based methods more scalable and more efficient for large data tables. The proposed method does not consist of the generalization step. The main effort is shifted in to rule matching step. We show that the set of such rules, that match a new object (to be classified) can be selected by modification of *Apriori algorithm* proposed in [1] for sequent item set generation from data bases.

2 Lazy learning for rough sets methods

The problem is formulated as follows: *given a decision table* $\mathbb{A} = (U, A \cup \{dec\})$ *and a new object* x*, find all decision rules from the set*

$$MatchRules(\mathbb{A}, x) = \{\mathbf{r} \in MinRules(\mathbb{A}, \lambda_{\max}, \sigma_{\min}, \alpha_{\min}) : x \text{ satisfies } \mathbf{r}\}$$

If the set of such rules is too large, one can modify the problem by searching for as much rules from $MatchRules(\mathbb{A}, x)$ as possible. Let

$$Desc(x) = \{d_1, d_2, ...d_k\}, \text{ where } d_i = (a_i = a_i(x))$$

be a set of all descriptors derived from x. Let $\mathbf{P}_i = \{S \subset Desc(x) : |S| = i\}$ and let $\mathbf{P} = \bigcup_{i=1}^{k} \mathbf{P}_i$. One can see that every decision rule $\mathbf{r} \in MatchRules(\mathbb{A}, x)$ has a form

$$\mathbf{r} \equiv \bigwedge S \Rightarrow (dec = k)$$

for some $S \in \mathbf{P}$. Hence the problem of searching for $MatchRules(\mathbb{A}, x)$ is equivalent to the problem of searching for corresponding families of sets from $\mathbf{P}$.

Let $S \in \mathbf{P}$ be an arbitrary set of descriptors from $Desc(x)$. The vector $(s_1, ..., s_d)$ is called *class distribution of* S if and only if

$$s_i = |\{u \in U : (u \in DEC_i) \wedge (u \text{ satisfies } \bigwedge S)\}|$$

Let $support(S) = s_1 + ... + s_d$. We assume that the function $GetClassDistribution(S)$ returns the class distribution of S. One can see that this function can be computed by using simple SQL query of form

```
SELECT COUNT FROM ... WHERE ... GROUP BY ...
```

The algorithm consists of a number of iterations. In the i^{th} iteration all decision rules containing i descriptors (length $= i$) are extracted. For this purpose we compute three families $\mathbf{C}_i$, $\mathbf{R}_i$ and $\mathbf{F}_i$ of subsets of descriptors in the i^{th} iteration:

– The family $\mathbf{C}_i \subset \mathbf{P}_i$ consists of "candidate sets" of descriptors and it can be generated without any database operation.
– The family $\mathbf{R}_i \subset \mathbf{C}_i$ consists of such candidates which contains descriptors (from left hand side) of some decision rules from $MatchRules(\mathbb{A}, x)$.
– The family $\mathbf{F}_i \subset \mathbf{C}_i$ consists of such candidates which are supported by more than $\sigma_{\min}$ (frequent subsets).

In the algorithm, we apply the function $AprGen(\mathbf{F}_i)$ to generate the family $\mathbf{C}_{i+1}$ of candidate sets from $\mathbf{F}_i$ (see [1]). The main idea is based on the following observations:

1. Let $S \in \mathbf{P}_{i+1}$ and let $S_1, S_2, ..., S_{i+1}$ be subsets formed by removing from S one descriptor we have:

$$support(S) \leq \min\{support(S_1), ..., support(S_{i+1})\}$$

   This means that if $S \in \mathbf{R}_{i+1}$ than $S_j \in \mathbf{F}_i$ for $j = 1, ..., i+1$. Hence if $S_j \in \mathbf{F}_i$ for $j = 1, ..., i+1$, then S can be inserted to $\mathbf{C}_{i+1}$;
2. Let $s(j)_1, ..., s_d^{(j)}$ be the class distribution of S_j and let $s_1, ..., s_d$ be the class distribution of S, we have

$$s_k \leq \min\{s(1)_k, ..., s(i+1)_k\} \text{ for } k = 1, ..., d$$

   This means that if $\max_k\{\min\{s(1)_k, ..., s(i+1)_k\}\} \leq \alpha_{\min} * \sigma_{\min}$, then we can remove S from $\mathbf{C}_{i+1}$;

ALGORITHM: Rule selection

Input: The object x, the maximal length $\lambda_{\max}$, the minimal support $\sigma_{\min}$, and the minimal confidence $\alpha_{\min}$.
Output: The set $MatchRules(\mathbb{A}, x)$ of decision rules from $MinRules(\mathbb{A}, \lambda_{\max}, \sigma_{\min}, \alpha_{\min})$ matching x.

```
BEGIN
    C_1 := P_1; i := 1;
    WHILE (i ≤ λ_max) AND (C_i IS NOT EMPTY)) DO
    BEGIN
        F_i := ∅; R_i := ∅;
        FOR C ∈ C_i DO
        BEGIN
            (s_1, ..., s_d) := GetClassDistribution(C);
            support = s_1 + ... + s_d;
            IF support ≥ σ_min THEN
                IF (max{s_1, ..., s_d} ≥ α_min * support) THEN
                    R_i := R_i ∪ {C};
                ELSE
                    F_i := F_i ∪ {C};
        END
        C_{i+1} := AprGen(F_i); i := i + 1;
    END
    RETURN ⋃_i R_i
END
```

Fig. 3. The rule selection method based on Apriori algorithm

3 Example

We illustrate our concept for *whether* decision table in Figure 4 .

Using system ROSETTA [6] one can see that $MatchRules(\mathbb{A}, x)$ consists of two decision rules:

`(outlook = sunny) AND (humidity = high)` $\Rightarrow play = no$
`(outlook = sunny) AND (temperature = mild) AND (windy = TRUE)` $\Rightarrow play = yes$

We show that this set can be found using our algorithm.

Let us define a new decision table $\mathbb{A}|_x = (U, A|_x \cup \{dec\})$ where $A|_x = \{a_1|_x, ..., a_k|_x\}$ is a new set of binary attributes defined as follows:

$$a_i|_x(u) = \begin{cases} 1 \text{ if } a_i(u) = a_i(x) \\ 0 \text{ otherwise;} \end{cases}$$

It is easy to see that every decision rule from $MatchRules(\mathbb{A}, x)$ can be derived from $\mathbb{A}|_x$. The table $\mathbb{A}|_x$ will be used to illustrate our approach. In Figure 4 we present the decision table $\mathbb{A}|_x$.

$\mathbb{A}$	a_1	a_2	a_3	a_4	dec
ID	outlook	temperature	humidity	windy	play
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no
x	sunny	mild	high	TRUE	?

Fig. 4. A decision table $\mathbb{A}$ and new object x

4 Concluding remarks

We presented the rough set based classification method which is scalable for large data set. The method is based on lazy learning idea and Apriori algorithm.

One can see that if $x \in U$ then the presented algorithm can generate the object oriented reducts for x. Hence the proposed method can be applied also for eager learning. This method can be used for adaptive rule generation system where data is growing up in time. In the next paper we will describe more details about this observation.

$\lambda_{max} = 3; \sigma_{\min} = 1; \alpha_{\min} = 1$

$i = 1$				$i = 2$				$i = 3$			
$\mathbf{C}_1$	check	$\mathbf{R}_1$	$\mathbf{F}_1$	$\mathbf{C}_2$	check	$\mathbf{R}_2$	$\mathbf{F}_2$	$\mathbf{C}_3$	check	$\mathbf{R}_3$	$\mathbf{F}_3$
$\{d_1\}$	(3,2)		$\{d_1\}$	$\{d_1, d_2\}$	(1,1)		$\{d_1, d_2\}$	$\{d_1, d_3, d_4\}$	(0,1)	$\{d_1, d_3, d_4\}$	
$\{d_2\}$	(4,2)		$\{d_2\}$	$\{d_1, d_3\}$	(3,0)	$\{d_1, d_3\}$					
$\{d_3\}$	(4,3)		$\{d_3\}$	$\{d_1, d_4\}$	(1,1)		$\{d_1, d_4\}$	$\{d_2, d_3, d_4\}$	(1,1)		$\{d_2, d_3, d_4\}$
$\{d_4\}$	(3,3)		$\{d_4\}$	$\{d_2, d_3\}$	(2,2)		$\{d_2, d_3\}$				
				$\{d_2, d_4\}$	(1,1)		$\{d_2, d_4\}$				
				$\{d_3, d_4\}$	(2,1)		$\{d_3, d_4\}$				

$MatchRules(\mathbb{A}, x) = \mathbf{R}_2 \cup \mathbf{R}_3$

(outlook = sunny) AND (humidity = high) $\Rightarrow$ $play = no$

(outlook = sunny) AND (temperature = mild) AND (windy = TRUE) $\Rightarrow$ $play = yes$

Fig. 5. The illustration of algorithm for $\lambda_{max} = 3; \sigma_{\min} = 1; \alpha_{\min} = 1$.

Acknowledgement

This paper has been partially supported by Polish State Committee of Research (KBN) grant No 8T11C02519 and grant of the Wallenberg Foundation.

References

1. Agrawal R., Mannila H., Srikant R., Toivonen H., Verkamo A.I.: Fast discovery of assocation rules. In V.M. Fayyad, G.Piatetsky Shapiro, P. Smyth, R. Uthurusamy (eds): *Advanced in Knowledge Discovery and Data Mining*, AAAI/MIT Press, 1996, pp. 307-328.
2. Aha D.W. (Editorial): "Special Issue on Lazy Learning", Artificial Intelligence Review, 11(1-5), 1997, pp. 1-6.
3. Bazan J.: A comparison of dynamic non-dynamic rough set methods for extracting laws from decision tables. In: L. Polkowski and A. Skowron (Eds.), *Rough Sets in Knowledge Discovery* **1**, Physica-Verlag, Heidelberg, 1998, pp. 321–365.
4. Komorowski J., Pawlak Z., Polkowski L. and Skowron A.: Rough sets: A tutorial. In: S.K. Pal and A. Skowron (eds.), Rough - fuzzy hybridization: A new trend in decision making, Springer-Verlag, Singapore, 1999, pp. 3-98.
5. Manish Mehta, Rakesh Agrawal, and Jorma Rissanen. SLIQ: A fast scalable classifier for data mining. In Proc. of the Fifth Int'l Conference on Extending Database Technology (EDBT), Avignon, France, March 1996. pp. 18-32
6. Ohrn A., Komorowski J., Skowron A., Synak P.: The ROSETTA Software System.In Polkowski, L., Skowron, A. (Eds.): *Rough Sets in Knowledge Discovery* **Vol. 1,2**, Springer Physica-Verlag, Heidelberg, 1998, pp. 572-576.
7. Pawlak Z.: *Rough sets: Theoretical aspects of reasoning about data*, Kluwer Dordrecht, 1991.
8. J. Shafer, R. Agrawal, and M. Mehta. SPRINT: A scalable parallel classifier for data mining. In Proc. 1996 Int. Conf. Very Large Data Bases, Bombay, India, Sept. 1996, pp. 544–555.
9. Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems. In. R. Słowiński (ed.). Intelligent Decision Support – Handbook of Applications and Advances of the Rough Sets Theory, Kluwer Academic Publishers, Dordrecht, 1992, pp. 311–362
10. Stefanowski J.: On rough set based approaches to induction of decision rules. In: A. Skowron, L. Polkowski (red.), Rough Sets in Knowledge Discovery Vol 1, Physica Verlag, Heidelberg, 1998, 500-529.
11. Wróblewski J., 1998. Covering with reducts - a fast algorithm for rule generation. In L. Polkowski and A. Skowron (Eds.): Proc. of RSCTC'98, Warsaw, Poland. Springer-Verlag, Berlin Heidelberg, pp. 402–407.
12. Ziarko, W.: Rough set as a methodology in Data Mining. In Polkowski, L., Skowron, A. (Eds.): *Rough Sets in Knowledge Discovery* **Vol. 1,2**, Springer Physica-Verlag, Heidelberg, 1998, pp. 554–576.