Scalable Classification Method Based on Rough Sets

Hung Son Nguyen

Institute of Mathematics,
Warsaw University, Banacha 2, Warsaw 02095, Poland
son@mimuw.edu.pl

Abstract. The existing rough set based methods are not applicable for large data set because of the high time and space complexity and the lack of scalability. We present a classification method, which is equivalent to rough set based classification methods, but is scalable and applicable for large data sets. The proposed method is based on lazy learning idea [2] and Apriori algorithm for *sequent item-set approaches* [1]. In this method the set of decision rules matching the new object is generated directly from training set. Accept classification task, this method can be used for adaptive rule generation system where data is growing up in time.

Keywords: Data mining, Scalability, Rough set, Lazy learning.

1 Introduction

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" and many others. Almost all methods based on rough sets use the 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 the new object.

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 (more precisely: there is a lack of proof showing that they can be scalable). The biggest troubles stick in the 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 and require multiple data scanning. In this paper we propose to adopt lazy learning idea to make rough set based methods more scalable. 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 Preliminaries

An *information system* [7] is a pair $\mathbb{A} = (U, A)$, where U is a non-empty, finite set of *objects* and $A = \{a_1, ..., a_k\}$ is a non-empty finite set of *attributes (or features)*, i.e. $a_i : U \to V_{a_i}$ for $i = 1, ..., k$, where V_{a_i} is called *the domain of* a_i. Let $B = \{a_{i_1}, ..., a_{i_j}\}$, where $1 \leq i_1 < ... < i_j \leq k$, be a subset of A, the set $INF_B = V_{a_{i_1}} \times V_{a_{i_2}} \times ... \times V_{a_{i_j}}$ is called *information space defined by* B. Function $inf_B : U \to INF_B$ defined by $inf_B(u) = \langle a_{i_1}(u), ..., a_{i_j}(u)\rangle$ is called "B-information map". The function inf_B defines a projection of objects from U into information space INF_B (or a view of U on features from B).

Using information map one can define the relation $IND(B) = \{(x, y) : inf_B(x) = inf_B(y)\}$ called *indiscernibility relation* (if $(x, y) \in IND(B)$ then we say that they are *indiscernible* by attributes from B. It is easy to show that $IND(B)$ is equivalent relation (see [9]). For any $u \in U$, the set $[u]_B = \{x \in U : (x, u) \in IND(B)\}$ is called equivalent class of u in B. Equivalent classes can be treated as building block to define basic notions of rough set theory.

The main subject of rough set theory is concept description. Let $X \subset U$ be a concept to be describe and $B \subset A$ is a set of accessible attributes. The set X can be described by attributes form B by $(\underline{B}X, \overline{B}X)$ where $\underline{B}X = \{u \in U : [u]_B \subset X\}$ and $\overline{B}X = \{u \in U : [u]_B \cap X \neq \emptyset\}$.

2.1 Classification Problem

Any information system of the form $\mathbb{A} = (U, A \cup \{dec\})$ with a distinguished attribute dec is called *decision table.* The attribute $dec \notin A$ is called *decision attribute.* In this paper we are dealing with the decision rule based approach, which is preferred by many Rough Set based classification methods [3,10,11,12].

Let $\mathbb{A} = (U, A \cup \{dec\})$ be a decision table. Without loss of generality we assume that $V_{dec} = \{1, \ldots, d\}$. Then the set $DEC_k = \{x \in U : dec(x) = k\}$ will be called the k^{th} *decision class of* $\mathbb{A}$ for $1 \leq k \leq d$. Any implication of form

$$(a_{i_1} = v_1) \wedge ... \wedge (a_{i_m} = v_m) \Rightarrow (dec = k) \tag{1}$$

where $a_{i_j} \in A$ and $v_j \in V_{a_{i_j}}$, is called *decision rule* for k^{th} decision class. Let $\mathbf{r}$ be an arbitrary decision rule of the form (1), then $\mathbf{r}$ can be characterized by:

$length(\mathbf{r})$ = the number of descriptor on the assumption of $\mathbf{r}$ (i.e. the left hand side of implication)

$[\mathbf{r}]$ = the carrier of $\mathbf{r}$, i.e. the set of objects from U satisfying the assumption of $\mathbf{r}$

$support(\mathbf{r})$ = the number of objects satisfying the assumption of $\mathbf{r}$: $support(\mathbf{r}) = card([\mathbf{r}])$

$confidence(\mathbf{r})$ = the confidence of $\mathbf{r}$: $confidence(\mathbf{r}) = \frac{|[\mathbf{r}] \cap DEC_k|}{|[\mathbf{r}]|}$

The decision rule $\mathbf{r}$ is called *consistent* with $\mathbb{A}$ if $confidence(\mathbf{r}) = 1$.

In data mining philosophy, we are interested on *short, strong* decision rules with *high confidence.* The linguistic features like "short", "strong" or "high confidence" of decision rules can be formulated by term of their length, support and confidence. Such rules can be treated as interesting, valuable and useful patterns in data. Any rule based classification method works in three phases (Figure 1):

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 $MatchRules(\mathbb{A}, x)$

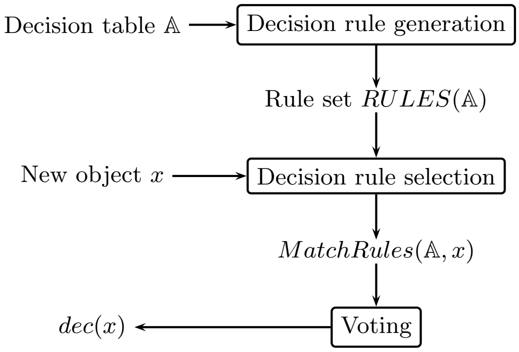
Decision table $\mathbb{A}$
Decision rule generation
Rule set $RULES(\mathbb{A})$
New object x
Decision rule selection
$MatchRules(\mathbb{A}, x)$
$dec(x)$
Voting

Fig. 1. The Rule base classification system

2.2 Rough Sets and Classification Problems

Unfortunately, the number of all decision rules can be exponential with regard to the size of the given decision table [3,4,9,11]. In practice, we apply some heuristics to generate a subset of "interesting" decision rules. Many decision rule generation methods have been developed by using Rough set theory. One of the most interesting approaches 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 $\mathbb{A}$, denoted by $MinConsRules(\mathbb{A})$, can be found by computing *object oriented reducts* (or local reducts) [4,3,11]. 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 $MinRules(\mathbb{A}, \lambda_{\max}, \sigma_{\min}, \alpha_{\min})$.

2.3 Lazy Learning

The classification methods based on learning schema presented in Figure 1 are called *eager (or laborious) methods.* In *lazy learning* methods new objects are classified without generalization step.

Lazy learning methods need more time complexity for the classification step, i.e., the answer time for the question about decision of a new object is longer than in eager classification methods. But lazy classification methods are *well scalable*, i.e. it can be realized for larger decision table using distributed computer system [5,8]. The scalability property is also very advisable in data mining. Unfortunately, the eager classification methods are weakly scalable. As we recall before, the time and memory complexity of existing algorithms does not make possible to apply rule base classification methods for very large decision table[1].

3 Lazy Learning for Rough Sets Methods

The most often reproach, which is placed for Rough set based methods, relates to the lack of scalability. In this paper we try to defend rough set methods again such reproaches. We show that some classification methods based on rough set theory can be modified by using lazy learning algorithms that make them more scalable. The lazy rule-based classification diagram is presented in Figure 2

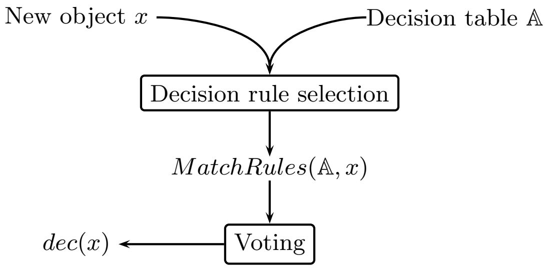

Fig. 2. The lazy rule-based classification system

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 in the algorithm. We show that this diagram can work for the classification method described in Section 2.3 using the set $MinRules(\mathbb{A}, \lambda_{\max}, \sigma_{\min}, \alpha_{\min})$ of decision rules. The problem is formulated as follows: *given a decision table* $\mathbb{A} = (U, A \cup \{dec\})$ *and a new object* x*, find all (or almost all) decision rules of the set*

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

[1] i.e., such tables containing more than 10^6 objects and 10^2 attributes.

Let $Desc(x) = \{d_1, d_2, ...d_k\}$, where $d_i \equiv (a_i = a_i(x))$, 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 $[\bigwedge S \Rightarrow (dec = k)]$ 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 subsets from $\mathbf{P}$ using minimal number of I/O operations to the database. We will show that the set $MatchRules(\mathbb{A}, x)$ can be found by modifying Apriori algorithm (see [1]).

Let $S \in \mathbf{P}$ be an arbitrary set of descriptors from $Desc(x)$. The support of S can be defined by $support(S) = |\{u \in U : u \text{ satisfies } \bigwedge S\}|$. Let $s_i = |\{u \in U : (u \in DEC_i) \wedge (u \text{ satisfies } \bigwedge S)\}|$, then the vector $(s_1, ..., s_d)$ is called *class distribution of* S. Obviously $support(S) = s_1 + ... + s_d$. 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 ...`".

```
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
     $\mathbf{C}_1 := \mathbf{P}_1$; $i := 1$;
     WHILE ($i \leq \lambda_{\max}$) AND ($\mathbf{C}_i$ IS NOT EMPTY)) DO
        $\mathbf{F}_i := \emptyset$; $\mathbf{R}_i := \emptyset$;
        FOR $C \in \mathbf{C}_i$ DO
           $(s_1, \ldots, s_d) := GetClassDistribution(C)$;
           $support = s_1 + \ldots + s_d$;
           IF $support \geq \sigma_{\min}$ THEN
              IF ($\max\{s_1, \ldots, s_d\} \geq \alpha_{\min} * support$) THEN
                 $\mathbf{R}_i := \mathbf{R}_i \cup \{C\}$;
              ELSE
                 $\mathbf{F}_i := \mathbf{F}_i \cup \{C\}$;
        ENDFOR
        $\mathbf{C}_{i+1} := AprGen(\mathbf{F}_i)$; $i := i + 1$;
     ENDWHILE
     RETURN $\bigcup_i \mathbf{R}_i$
  END
```

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

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]) using 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 $support(S) \leq \min\{support(S_j)$, for any $j = 1, \ldots, j+1$. This means that if $S \in \mathbf{R}_{i+1}$ then $S_j \in \mathbf{F}_i$ 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 $s_1^{(j)}, ..., s_d^{(j)}$ be the class distribution of S_j and let $s_1, ..., s_d$ be the class distribution of S, we have $s_k \leq \min\{s_k^{(1)}, ..., s_k^{(i+1)}\}$, 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 $\mathbf{C}_{i+1}$;

4 Example

In Figure 4, we illustrate the *weather* decision table and in Figure 5 we present the set $MinConsRules(\mathbb{A})$ generated by system ROSETTA [6].

$\mathbb{A}$	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

One can see that $MatchRules(\mathbb{A}, x)$ consists of two rules:

rules	**support**
outlook(overcast)⇒play(yes)	4
humidity(normal) AND windy(FALSE)⇒play(yes)	4
outlook(sunny) AND humidity(high)⇒play(no)	3
outlook(rainy) AND windy(FALSE)⇒play(yes)	3
outlook(sunny) AND temperature(hot)⇒play(no)	2
outlook(rainy) AND windy(TRUE)⇒play(no)	2
outlook(sunny) AND humidity(normal)⇒play(yes)	2
temperature(cool) AND windy(FALSE)⇒play(yes)	2
temperature(mild) AND humidity(normal)⇒play(yes)	2
temperature(hot) AND windy(TRUE)⇒play(no)	1
outlook(sunny) AND temperature(mild) AND windy(FALSE)⇒play(no)	1
outlook(sunny) AND temperature(cool)⇒play(yes)	1
outlook(sunny) AND temperature(mild) AND windy(TRUE)⇒play(yes)	1
temperature(hot) AND humidity(normal)⇒play(yes)	1

Fig. 5. The set of all minimal decision rules generated by ROSETTA

(outlook = sunny) AND (humidity = high) ⇒ *play* = *no* (rule nr 3)
(outlook = sunny) AND (temperature = mild) AND (windy = TRUE) ⇒ *play* = *yes* (rule nr 13)

Figure 6 shows that this set can be found using our algorithm.

$i = 1$				$i = 2$				$i = 3$			
$\mathbf{C}_1$	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)		$\{d_3, d_4\}$				

$\mathit{MatchRules}(\mathbb{A}, x) = \mathbf{R}_2 \cup \mathbf{R}_3$:
(outlook = sunny) AND (humidity = high) ⇒ *play* = *no*
(outlook = sunny) AND (temperature = mild) AND (windy = TRUE) ⇒ *play* = *yes*

Fig. 6. The illustration of algorithm for $\lambda_{max} = 3; \sigma_{\min} = 1; \alpha_{\min} = 1$.

5 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.

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. Fayad, 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.