Nondeterministic Decision Rules in Classification Process

Piotr Paszek and Barbara Marszał-Paszek

Institute of Computer Science, University of Silesia
Będzińska 39, 41-200 Sosnowiec, Poland
{paszek,bpaszek}@us.edu.pl

Abstract. In the paper, we discuss nondeterministic rules in decision tables, called the truncated nondeterministic rules. These rules have on the right hand side a few decisions. We show that the truncated nondeterministic rules can be used for improving the quality of classification.

We propose a greedy algorithm of polynomial time complexity to construct these rules. We use this type of rules, to build up rule-based classifiers. These classifiers, classification algorithms, are used not only nondeterministic rules but also minimal rules in the sense of rough sets. These rule-based classifiers were tested on the group of decision tables from the UCI Machine Learning Repository. The reported results of the experiment show that the proposed classifiers based on nondeterministic rules improve the classification quality but it requires tuning some of their parameters relative to analyzed data.

Keywords: classification, decision tables, nondeterministic decision rules, rough sets, rule based classifier.

1 Introduction

Over the years many methods based on rule induction and rule-based classification systems were developed [10,17]. Some of them are based on rough sets [2,6,14,15,18] and some of them are based on cluster analysis [7]. In this paper we show that exist possibility for improving the rule-based classification systems.

We discuss a method for rule inducing based on searching for strong rules for a union of a few relevant decision classes – nondeterministic decision rules. Because these rules are created by shortening the deterministic rules they are called truncated nondeterministic rules.

In the paper, the following classification problem is considered: for a given decision table T [11,12] and a new object v generate a value of the decision attribute on v using values of conditional attributes on v.

In [16] Skowron and Suraj shown that there exist information systems $S = (U, A)$ [11], where U is a finite set of objects and A is a finite set of attributes, such that the set U can't be described by deterministic rules. In [9] Moshkov shown that for any information system, the set can be described by nondeterministic

P. Herrero et al. (Eds.): OTM 2012 Workshops, LNCS 7567, pp. 485–494, 2012.

(inhibitory) rules. Inhibitory rules [3] are a special case of nondeterministic rules. These results inspired us to use the nondeterministic rules in a classification process [8].

We present an application of (truncated) nondeterministic rules in construction of rule-based classifiers. We also include the results of experiments that shows that by combining the rule-based classifiers based on the minimal decision rules [11,15] with the nondeterministic rules that have sufficiently large support [1], it is possible to improve the classification quality and reduce the classification error.

The paper consists of six sections. In Section 2, we recall the notions of a decision table and deterministic and nondeterministic decision rules. In Sections 3 and 4 we present a greedy algorithm for nondeterministic decision rule construction and main steps in construction of classifiers enhanced by nondeterministic rules. In Section 5 the results of the experiments with real-life data from the UCI Machine Learning Repository [5] are discussed. Section 6 contains short conclusions.

2 Basic Notations

In 1982 Pawlak proposed the rough set theory as an innovative mathematical tool for describing knowledge, including the uncertain and inexact knowledge [11]. In this theory knowledge is based on possibility (capability) of classifying objects. The objects may be for instance real objects, statements, abstract concepts and processes.

Let $T = (U, A, d)$ be a *decision table*, where $U = \{u_1, \ldots, u_n\}$ is a finite nonempty set of *objects*, $A = \{a_1, \ldots, a_m\}$ is a finite nonempty set of *conditional attributes* (functions defined on U), and d is the *decision attribute* (function defined on U).

We assume that for each $u_i \in U$ and each $a_j \in A$ the value $a_j(u_i)$ belong to $V_{a_j}(T)$ and the value $d(u_i)$ belong to $V_d(T)$, where $V_d(T)$ denotes the set of values of the decision attribute d on objects from U.

2.1 Deterministic Decision Rules

In general, the *deterministic decision rule* in T has the following form:

$$(a_{j_1} \in V_1) \wedge \ldots \wedge (a_{j_k} \in V_k) \to (d = v),$$

where $a_{j_1}, \ldots, a_{j_k} \in A$, $V_j \subseteq V_{a_j}$, for $j \in \{1, \ldots, k\}$ and $v \in V_d(T)$. The predecessor of this rule is a conjunction of generalized descriptors and the successor of this rule is a descriptor.

In the rule-based classifiers, most commonly used are the rules in the form of Horn Clauses

$$(a_{j_1} = b_1) \wedge \ldots \wedge (a_{j_k} = b_k) \ \to \ (d = v)$$

where $k > 0$, $a_{j_1}, \ldots, a_{j_k} \in A$, $b_1, \ldots, b_k \in V_A(T)$, $v \in V_d(T)$ and numbers $j_1, \ldots, j_k$ are pairwise different. The predecessor of this rule (conditional part) is a conjunction of descriptors.

2.2 Nondeterministic Decision Rules

In this paper, we also consider nondeterministic decision rules. A *nondeterministic decision rule* in a given decision table T is of the form:

$$(a_{j_1} \in V_1) \wedge \ldots \wedge (a_{j_k} \in V_k) \rightarrow d = (c_1 \vee \ldots \vee c_s), \tag{1}$$

where $a_{j_1}, \ldots, a_{j_k} \in A$, $V_j \subseteq V_{a_j}$, for $j \in \{1, \ldots, k\}$, numbers $j_1, \ldots, j_k$ are pairwise different, and $\emptyset \neq \{c_1, \ldots, c_s\} \subseteq V_d(T)$.

Some notation about rules of the form (1) are introduced in [4].

Let us introduce some notation.

If r is the nondeterministic rule (1) then by $lh(r)$ we denote its left hand side, i.e., the formula $(a_{j_1} \in V_1) \wedge \ldots \wedge (a_{j_k} \in V_k)$, and by $rh(r)$ its right hand side, i.e., the formula $d = (c_1 \vee \ldots \vee c_s)$.

By $||lh(r)||_T$ (or $||lh(r)||$, for short) we denote all objects from U satisfying $lh(r)$ [12]. To measure the quality of such rules we use coefficients called the support and the confidence [1]. They are defined as follows. If r is a nondeterministic rule of the form (1) then the support of this rule in the decision system T is defined by

$$supp(r) = \frac{|\ ||lh(r)|| \cap ||rh(r)||\ |}{|U|},$$

and the confidence of r in T is defined by

$$conf(r) = \frac{|\ ||lh(r)|| \cap ||rh(r)||\ |}{|\ ||lh(r)||\ |}.$$

We also use a normalized support of r in T defined by

$$norm\_supp(r) = \frac{supp(r)}{\sqrt{|V(r)|}},$$

where $V(r) \subseteq V_d(T)$ is a decision values set from right hand side of the rule $(rh(r))$.

2.3 Truncated Nondeterministic Rules

Now we can define a parameterized set of truncated nondeterministic decision rules that are used in Section 4 for enhancing the quality of classification of rule-based classifiers. This type of nondeterministic rules appears as a result of shortening rules according to the principle MDL (Minimum Description Length) [13].

This parameterized set is defined as the set of all nondeterministic rules r (over attributes in T) such that:

1. On the left hand sides of such rules are only conditions of the form $a \in \{v\}$, where $v \in V_a$. We write $a = v$ instead of $a \in \{v\}$;
2. $conf(r) \geq \alpha$, where $\alpha \in [0.5, 1]$ is a threshold;

3. $|V(r)| \leq k < |V_d(T)|$, where k is a threshold used as an upper bound on the number of decision values on the right hand sides of rules – k is assumed to be small.

Hence, the *truncated nondeterministic decision rules* are of the form:

$$(a_{j_1} = b_1) \wedge \ldots \wedge (a_{j_k} = b_k) \rightarrow d = (c_1 \vee \ldots \vee c_s), \tag{2}$$

where $a_{j_1}, \ldots, a_{j_k} \in A$, for $j \in \{1, \ldots, k\}$, $b_j \in V_{b_j}(T)$, numbers $j_1, \ldots, j_k$ are pairwise different, and $\emptyset \neq \{c_1, \ldots, c_s\} \subseteq V_d(T)$.

The algorithm presented in Section 3 is searching for truncated nondeterministic rules with sufficiently large support and relatively small (in comparison to the set of all possible decisions), sets of decisions defined by the right hand sides of such rules for the decision table T.

3 Algorithm for Nondeterministic Decision Rule Construction

Let us describe the algorithm with threshold $\alpha \in [0.5, 1]$ which constructs truncated nondeterministic decision rules for T. This algorithm is based on greedy strategy which is used to minimize the length of rules.

The algorithm consists of two main steps.

In the first step, the deterministic decision rules are constructed for a given decision table T. The different algorithms may be used to build up the set of deterministic rules. In Section 5 we present the influence of the choice of the deterministic rules that will be shorten in the second step of the algorithm on the classification quality.

In the second step, of the algorithm, the set of deterministic decision rules are shortened (truncated). *TruNDeR* (*Tru*ncated *N*ondeterministic *De*cision *R*ules) is the name of the algorithm implementing the second step. Algorithm 1 contains pseudo-code of algorithm *TruNDeR*.

Algorithm *TruNDeR*, for truncated nondeterministic rules construction, has polynomial computational complexity, which depends on number of deterministic rules, and number of objects and number of attributes in the decision table.

4 Classifiers

In this section, we present an application of nondeterministic rules for the classification process. We constructed two classifier *MDR* and *TNDR*.

The set of minimal rules, generated using *RSESlib* library (Rough Set Exploration System library) [2], and standard voting procedure to resolve conflicts between rules, were used to induce *MDR* classifier.

The set of truncated nondeterministic rules, generated by the *TruNDeR* algorithm and the set of minimal rules, generated using *RSESlib* library, are used to induce our second classifier (*TNDR*).

Algorithm 1. *TruNDeR* – greedy algorithm for truncated nondeterministic decision rule construction

Input: T – decision table, R_d – a set of deterministic decision rules of T, $\alpha \in [0.5, 1]$, k – upper bound on the number of decision values;
Output: $R_{nd}(\alpha)$ – a set of nondeterministic decision rules for T.

$R_{nd} \leftarrow \emptyset$;
for all $r \in R_d$ **do**
  $\{r : L \rightarrow (d = v);\ L = D_1 \wedge \ldots \wedge D_m;\ v \in V_d;\}$
  $STOP \leftarrow false$;
  $\lambda_L \leftarrow norm\_supp(L)$;
  repeat
    for all condition attributes from r **do**
      $L^i = D_1 \wedge \ldots \wedge D_{i-1} \wedge D_{i+1} \wedge \ldots \wedge D_m$;
      $\{L^i$ is obtained by dropping i-th attribute from the left hand side of rule $r\}$
      $||L^i||_T$; $\theta = \{v \in V_d : \exists_{x \in U_{L^i}} d(x) = v\}$;
      Sorting in decreasing order θ;
      $\theta_i \subset \theta$: $conf(L^i \rightarrow (d = \theta_i)) = \frac{|||L^i|| \cap ||\theta_i|||}{|||L^i|||} \geq \alpha$; $\{\theta_i$ greedy selection$\}$
      $\lambda_{L^i} \leftarrow norm\_supp(L^i \rightarrow \theta_i)$;
    end for
    $\lambda^i_{max} \leftarrow argmax\{\lambda_{L^i}\}$;
    if $\lambda^i_{max} \geq \lambda_L$ **then**
      $L \leftarrow L^i;\ \lambda_L \leftarrow \lambda^i_{max};\ \{r_{nd} : L \rightarrow (d = \theta_i);\ \lambda_L\}$
    else
      $STOP \leftarrow true$;
    end if
  until STOP
  if $|\theta_i| \leq k$ **then**
    $R_{nd} \leftarrow R_{nd} \cup \{r_{nd}\}$;
  end if
end for
return R_{nd};

Because we have two groups of rules in the classification process, in the *TNDR* classifier, we should negotiate between them.

For any new object the decision value set is generated as follows.

First, for any new object, all (truncated) nondeterministic rules matching the object are extracted. Next, from these matched rules, a rule with the largest normalized support is selected. In the case when several rules have the same support, the decision value set $V(r)$ of the nondeterministic rule r with the smallest set of decision value set ($|V(r)|$) is selected. If still several nondeterministic rules with the above property exist then first of them is selected.

Next, for this object, all minimal rules matching the object are extracted. We obtain a single decision value using standard voting procedure.

In this way, for any new object we obtain a decision value $v \in V_d(T)$ and a decision value set $V(r)$, where r is the rule selected from the set of nondeterministic rules.

The final decision for a given new object is obtained from the decision value v and decision value set $V(r)$ by the following strategy for resolving conflicts [8].

1. If for a given new object the standard voting based on minimal rules predicts the decision value v and $v \in V(r)$, (i.e., no conflict arises) then we take as the final decision the single decision v.
2. If for a given new object the standard voting based on minimal rules predicts the decision value v and $v \notin V(r)$ (i.e., conflict arises) then we take as the final decision value the single decision value v if support of the minimal (deterministic) rule is larger than the normalized support of nondeterministic decision rule r and selected for the given new object. In the opposite case, we take as the final decision a single decision value from the set $V(r)$, with the largest support in T among decisions from $V(r)$.
3. If for a new object, the standard voting based on minimal rules predicts the decision value v and this object does not match any nondeterministic rule then we assign the decision v as the final decision.
4. If a given new object does not match any of the minimal rules then we assign as the final decision the single decision from $V(r)$ with the largest support among decisions from $V(r)$, where r is the rule selected by voting on nondeterministic rules.
5. In the remaining cases, a given new object is not classified.

5 Experiments

We have performed the experiments on decision tables from the UCI Machine Learning Repository [5] using proposed *MDR* and *TNDR* classification algorithms.

The data sets selected for the experiments included the following: Balance Scale, Iris, Lymphography, Postoperative and Zoo.

Decision table *BalanceScale* was generated in 1976 to model psychological experimental results. *Iris* is the best known database to be found in the pattern recognition literature. *Lymphography* data is one of three domains provided by the University Medical Center, Institute of Oncology from Ljubljana. The classification task of decision table *Postoperative* is to determine when patients in a postoperative recovery area should be sent to the next one. Decision table *Zoo* is a simple database containing information about animals from the zoo.

The *MDR* classification algorithm is based on all minimal decision rules. The *TNDR* classification algorithm is based on all minimal decision rules and truncated nondeterministic rules.

On the input of the *TruNDeR* algorithm, which constructs truncated nondeterministic decision rules, we can use different sets of deterministic rules. Therefore, we can check how the choice of the deterministic decision rules in the *TruNDeR* algorithm influence the classification quality in the *TNDR* algorithm.

We used two types of deterministic rules. The first type of deterministic rules are minimal rules. In the Table 1, in the rows marked by (a) truncated rules are created from minimal rules.

The second type of deterministic rules are complete rules (rows from the decision table). In the Table 1, in the rows marked by (b) truncated rules are created from full rows from decision table.

Table 1. Accuracy of classifiers based on nondeterministic decision rules - cross-validation method

Decision table		Clas.	Classification algorithm							
			$MDR^{(1)}$	$TNDR^{(1)}$, $\alpha^{(2)}$						
name	#dec.	factor		(3)	1.0	0.9	0.8	0.7	0.6	0.5
Balance Scale	3	a×c	78.54	(a)	80.00	82.13	82.10	80.91	79.97	77.09
		mrd	1.78		1.44	2.13	2.10	2.35	2.37	1.79
		a×c		(b)	79.46	**82.18**	82.16	80.96	79.84	77.10
		mrd			1.82	2.34	2.32	2.56	2.56	1.90
Iris	3	a×c	88.40	(a)	87.07	86.33	83.87	81.60	80.80	80.80
		mrd	7.07		7.07	11.67	10.53	9.07	9.87	11.47
		a×c		(b)	**88.73**	87.87	87.00	85.53	85.33	85.20
		mrd			7.40	10.53	9.67	9.53	9.33	10.53
Iris (discret.)	3	a×c	94.13	(a)	94.13	93.20	92.40	88.53	87.87	86.33
		mrd	4.80		4.80	5.47	12.40	11.20	10.53	10.33
		a×c		(b)	**94.20**	94.00	93.40	88.87	88.20	87.80
		mrd			4.87	4.67	10.73	11.53	10.87	9.13
Lympho-graphy	8	a×c	37.36	(a)	37.36	37.36	37.36	37.36	37.36	37.36
		mrd	4.26		4.26	4.26	4.26	4.26	4.26	4.26
		a×c		(b)	37.36	37.36	37.36	37.36	37.36	37.36
		mrd			4.26	4.26	4.26	4.26	4.26	4.26
Post-Operative	3	a×c	65.44	(a)	65.67	65.67	65.67	67.67	68.78	**69.22**
		mrd	3.44		3.44	3.44	3.44	5.44	3.22	2.56
		a×c		(b)	65.67	65.67	65.67	66.89	69.11	69.11
		mrd			3.44	3.44	3.44	4.67	2.44	2.44
Zoo	7	a×c	**93.37**	(a)	83.07	83.07	84.26	85.45	86.34	86.44
		mrd	5.25		3.07	4.06	2.87	2.67	3.17	3.27
		a×c		(b)	83.07	83.37	84.36	85.64	86.34	86.53
		mrd			3.07	3.76	2.77	2.48	3.17	3.37

(1) In the column marked by *MDR* the classification is defined by the classification algorithm based on deterministic rules. In the columns marked by *TNDR* the classification is defined by the classification algorithm based on nondeterministic and deterministic rules.
(2) Confidence of nondeterministic rules generated by the algorithm is not smaller than the parameter α.
(3) In the rows marked by (a) nondeterministic rules are created from minimal rules. In the rows marked by (b) nondeterministic rules are created from rows from decision table (complete rules).

In evaluation of the accuracy of classification algorithms on a decision tables (i.e., the percentage of correctly classified objects) the 5–fold cross-validation method was used. For any considered data table, we used the classification algorithms for different values of parameter α. On testing sets the accuracy and the coverage factor were calculated. Also the maximal relative deviation (mrd) was calculated.

Table 1 contain the results of our experiments. The *TNDR* classifier was compared with *MDR* classifier.

For three decision tables – *Balance Scale, Iris, Iris (discretization)* – the classification quality measured by *accuracy* $\times$ *coverage* (marked in the table as the a$\times$c) was better for the *TNDR* classification algorithm, when truncated rules are created from complete rules, than in the case of the *MDR* classification algorithm.

For the decision table *Post-Operative* the classification quality measured by *accuracy* $\times$ *coverage* was better for the *TNDR* classification algorithm, when truncated rules are created from minimal rules, than in the case of the *MDR* classification algorithm.

For one data set (*Lymphography*), the classification quality for both classifiers *TNDR* and *MDR* was equal.

For one data set (*Zoo*), using only deterministic rules in the classification process, the result was better than in case of the *TNDR* classification algorithm.

For obtaining those results it was necessary to optimize the threshold α for each data table. This means that the parameter α should be tuned for each data set.

6 Conclusions

The results of experiments with truncated nondeterministic rules are showing that these rules can improve the classification quality. We have demonstrated this by using classification algorithms based on minimal decision rules and truncated nondeterministic rules. The experiments have shown that proposed classifiers can improve classification accuracy, in our experiments the improvement was for the most decision tables.

The proposed *TNDR* (a) and *TNDR* (b) classifiers are comparable in case of the classification quality. For decision tables *Balance Scale, Iris, Iris (discretization)* and *Zoo* we got better result for the *TNDR* (b) classifier then for the *TNDR* (a) classifier. For Post-Operative data set we got better result for the *TNDR* (a) classifier then for the *TNDR* (b) classifier. For the data set *Lymphography*, the classification quality for both *TNDR* (a) and *TNDR* (a) classifiers is the same.

These results show that we can replace the minimal rules with the complete rules in the *TruNDeR* algorithm. This is important for this reason that the *TruNDeR* algorithm, for truncated nondeterministic rules generation, has polynomial computational complexity, which depends on number of objects and number of attributes.

At this moment the proposed *TNDR* classification algorithm uses nondeterministic rules (from the *TruNDeR*) and deterministic rules – minimal rules (from the *RSESlib*). The algorithm for constructing nondeterministic rules has polynomial computational complexity, and the algorithm for constructing minimal rules has exponential computational complexity. To decrease computational complexity of the *TNDR* algorithm, we plan to use others algorithms for constructing deterministic rules (e.g. based on subsets of minimal decision rules or decision trees).

In the future, we plan to compare the *TNDR* algorithm with other classifiers (e.g. decision trees, SVM).

References

1. Agrawal, R., Imielinski, T., Swami, A: Mining Association Rules Between Sets of Items in Large Databases. In: Buneman, P., Jajodia, S. (eds.) Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pp. 207–216. ACM Press, New York (1993)
2. Bazan, J.G., Szczuka, M.S., Wojna, A., Wojnarski, M.: On the Evolution of Rough Set Exploration System. In: Tsumoto, S. et al (eds.) RSCTC 2004. LNCS, vol. 3066, pp. 592–601. Springer-Verlag, Heidelberg (2004)
3. Delimata, P., Moshkov, P., Skowron, A., Suraj, Z.: Inhibitory Rules in Data Analysis: A Rough Set Approach. Studies in Computational Intelligence, vol. 163, Springer. Heidelberg (2009)
4. Delimata, P. et al.: Comparison of Some Classification Algorithms Based on Deterministic and Nondeterministic Decision Rules. In: Peters, J.F. et al. (eds.) Transactions on Rough Sets XII. LNCS, vol. 6190, pp. 90–105, Springer, Heidelberg (2010)
5. Frank, A., Asuncion, A.: UCI Machine Learning Repository, University of California, Irvine (2010), `http://archive.ics.uci.edu/ml/`
6. Grzymala-Busse, J.W.: LERS - A Data Mining System. In: Maimon, O., Rokach, L. (eds.) The Data Mining and Knowledge Discovery Handbook, pp. 1347-1351, Springer, New York (2005)
7. Jach, T., Nowak-Brzezińska, A., Simiński, R., Xięski, T.: Towards a practical approach to discover internal dependencies in rule-based knowledge bases. RSKT 2011. LNCS, vol. 6954, pp. 232–237. Springer, Heidelberg (2011)
8. Marszał-Paszek, B., Paszek. P.: Minimal Templates and Knowledge Discovery. In: Kryszkiewicz, M. et al. (eds.) RSEISP 2007. LNCS, vol. 4585, pp. 411–416. Springer, Heidelberg (2007)
9. Moshkov, M., Skowron, A., Suraj, Z.: Maximal consistent extensions of information systems relative to their theories. Information Sciences 178 (12), 2600–2620 (2008)
10. Ryszard Michalski, `http://www.mli.gmu.edu/michalski/`
11. Pawlak, Z.: Rough Sets – Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
12. Pawlak, Z., Skowron, A.: Rudiments of Rough Sets. Information Sciences 177, 3–27 (2007); Rough Sets: Some Extensions. Information Sciences 177, 28–40 (2007); Rough Sets and Boolean Reasoning. Information Sciences 177, 41–73 (2007)
13. Rissanen, J.: Modeling by Shortest Data Description. Automatica 14, 465–471 (1978)

14. Rosetta, http://www.lcb.uu.se/tools/rosetta/
15. Rough Set Exploration System, http://logic.mimuw.edu.pl/~rses
16. Skowron, A., Suraj, Z.: Rough Sets and Concurrency. Bulletin of the Polish Academy of Sciences 41, pp. 237–254 (1993)
17. Triantaphyllou, E., Felici, G. (eds.): Data Mining and Knowledge Discovery Approaches Based on Rule Induction Techniques. Springer Science and Business Media, LLC, New York (2006)
18. Tsumoto, S.: Modelling Medical Diagnostic Rules Based on Rough Sets. RSCTC 1998. LNCS, vol. 1424, pp. 475–482, Springer-Verlag, Berlin (1998)