Lazy Classification Algorithms Based on Deterministic and Inhibitory Rules

Pawel Delimata
Chair of Comp. Sci.
Univ. of Rzeszów
Rejtana 16A
35-310 Rzeszów
Poland
pdelimata@wp.pl

Mikhail Moshkov
Inst. of Comp. Sci.
Univ. of Silesia
Będzińska 39
41-200 Sosnowiec
Poland
moshkov@us.edu.pl

Andrzej Skowron
Inst. of Math.
Warsaw Univ.
Banacha 2
02-097 Warsaw
Poland
skowron@mimuw.edu.pl

Zbigniew Suraj
Chair of Comp. Sci.
Univ. of Rzeszów
Rejtana 16A
35-310 Rzeszów
Poland
zsuraj@univ.rzeszow.pl

Abstract

In the paper, two families of lazy classification algorithms of polynomial time complexity are considered. These algorithms are based on deterministic (with a relation "attribute = value" on the right hand side) and inhibitory (with a relation "attribute ≠ value" on the right hand side) rules, but the direct generation of rules is not required. Instead of this, the considered algorithms extract efficiently for a new object some information about the set of rules which is next used by a decision-making procedure. Results of experiments show that the performance of algorithms based on inhibitory rules is in many cases better than the performance of algorithms based on deterministic rules.

Keywords: Rough sets, Decision tables, Information systems, Deterministic rules, Inhibitory rules.

1 Introduction

Let $S = (U, A)$ be an information system [9, 10], where U is a finite set of objects and A is a finite set of attributes (functions defined on U). We consider both *deterministic* and *inhibitory* rules of the following form:

$$a_1 = b_1 \wedge \ldots \wedge a_t = b_t \Rightarrow a_{t+1} = b_{t+1},$$
$$a_1 = b_1 \wedge \ldots \wedge a_t = b_t \Rightarrow a_{t+1} \neq b_{t+1},$$

respectively, where $a_1, \ldots, a_{t+1}$ are attributes from A and $b_1, \ldots, b_{t+1}$ are values of these attributes. We consider only true and realizable rules. *True* means that the rule is true for any object from U. *Realizable* means that the left hand side of the rule is true for at least one object from U.

We identify objects from U with tuples of values of attributes from A on these objects. Let V be the Cartesian product of ranges of attributes from A. We say that the set U can be described by deterministic (inhibitory) rules if there exists a set Q of true and realizable deterministic (inhibitory) rules such that the set of objects from V, for which all rules from Q are true, coincides with U. In [12, 14] it was shown that there exist information systems $S = (U, A)$ such that the set U can not be described by deterministic rules. In [7] it was shown that for any information system $S = (U, A)$ the set U can be described by inhibitory rules. It means that the inhibitory rules can express essentially more information encoded in information systems than the deterministic rules. This fact is a motivation for more wide use of inhibitory rules, in particular, in classification algorithms and in algorithms for synthesis of concurrent systems [12, 13]. There is an additional (intuitive) motivation for the use of inhibitory rules in classification algorithms: the support of inhibitory rules is very often larger than the support of deterministic rules.

To compare the "classification power" of inhibitory and deterministic rules we developed two analogous families of lazy classification algorithms based on deterministic and inhibitory rules respectively [6]. Results of experiments from [6] show that the algorithms based on inhibitory rules are noticeably better than the algorithms based on deterministic rules. In this paper, results of additional experiments with these lazy classification algorithms are reported.

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

To this end, we extract from the decision table T a number of information systems S_i, for $i \in V_d$, where V_d is the set of values of the decision attribute d in T. For $i \in V_d$, the information system S_i contains only objects (rows) of T with the value of the decision attribute equal to i.

For each information system S_i and a given object v, it is constructed (using polynomial-time algorithm) the so called characteristic table. For any object u from S_i and for any attribute a from S_i, the characteristic table contains the entry encoding information if there exist a rule which (i) is true for each object from S_i; (ii) is realizable for u, (iii) is not true for v, and (iv) has the attribute a on the right hand side. Based on the characteristic table the decision on the "degree" to which v belongs to S_i is computed, for any i, and a decision i with the maximal "degree" is selected.

Note that in [13] for classifying new objects it was proposed to use deterministic rules defined by conditional attributes in different decision classes.

In this paper, we consider both deterministic and inhibitory rules. Using these two kinds of rules and different evaluation functions a "degree" to which v belongs to S_i is computed by two families of classification algorithms.

In the literature, one can find a number of papers which are based on the analogous ideas: instead of construction of huge sets of rules it is possible to extract some information on such sets using algorithms having polynomial time complexity.

In [2, 3, 4] it is considered an approach based on decision rules (with decision attribute on the right hand side). These rules are obtained from the whole decision table T. The considered algorithms find, for a new object v and any decision i, the number of objects u from the information system S_i such that there exists a decision rule r satisfying the following conditions: (i) r is true for the decision table T, (ii) r is realizable for u and v, and (iii) r has the equality $d = i$ on the right hand side, where d is the decision attribute.

This approach was generalized by A. Wojna [15] to the case of decision tables with not only nominal but also numerical attributes.

Note that such algorithms can be considered as a kind of lazy learning algorithms [1].

The paper consists of five sections. In Sect. 2 the notions of deterministic and inhibitory characteristic tables, as well as the notion of evaluation function are introduced. Definitions of two families of lazy classification algorithms are included in Sect. 3. In Sect. 4 results of experiments are reported. Sect. 5 contains short conclusions.

2 Characteristic Tables

2.1 Information Systems

Let $S = (U, A)$ be an *information system*, where $U = \{u_1, \dots, u_n\}$ is a finite nonempty set of *objects* and $A = \{a_1, \dots, a_m\}$ is a finite nonempty set of *attributes* (functions defined on U). We assume that for each $u_i \in U$ and each $a_j \in A$ the value $a_j(u_i)$ belongs to ω, where $\omega = \{0, 1, 2, \dots\}$ is the set of nonnegative integers.

The information system $S = (U, A)$ can be represented by a *tabular representation*, i.e., a table with m columns and n rows. Columns of the table are labeled by attributes $a_1, \dots, a_m$. At the intersection of i-th row and j-th column the value $a_j(u_i)$ is included.

The set $\mathcal{U}(S) = \omega^m$ is called the *universe* for the information system S. Besides objects from U we consider also objects from $\mathcal{U}(S)$.

For any object (tuple) $v \in \mathcal{U}(S)$ and any attribute $a_j \in A$ the value $a_j(v)$ is equal to j-th integer in v.

2.2 Deterministic Characteristic Tables

Let us consider a rule

$$a_{j_1} = b_1 \wedge \ldots \wedge a_{j_t} = b_t \Rightarrow a_k = b_k, \qquad (1)$$

where $t \geq 0$, $a_{j_1}, \ldots, a_{j_t}, a_k \in A$, $b_1, \ldots, b_t, b_k \in \omega$, and numbers $j_1, \ldots, j_t, k$ are pairwise different. Such rules are called *deterministic* rules. The rule (1) is *realizable for an object* $u \in \mathcal{U}(S)$ if $a_{j_1}(u) = b_1, \ldots, a_{j_t}(u) = b_t$ or $t = 0$. The rule (1) is *true for an object* $u \in \mathcal{U}(S)$ if $a_k(u) = b_k$ or (1) is not realizable for u. The rule (1) is *true for* S if it is true for any object from U. The rule (1) is *realizable for* S if it is realizable for at least one object from U. By $Det(S)$ we denote the set of all deterministic rules each of which is true for S and realizable for S.

Let $u_i \in U$, $v \in \mathcal{U}(S)$, $a_k \in A$ and $a_k(u_i) \neq a_k(v)$. We say that a rule (1) from $Det(S)$ *contradicts* v *relative to* u_i *and* a_k (or, (u_i, a_k)-*contradicts* v, for short) if (1) is realizable for u_i but is not true for v. Our aim is to recognize for given objects $u_i \in U$ and $v \in \mathcal{U}(S)$, and given attribute a_k such that $a_k(u_i) \neq a_k(v)$ if there exist a rule from $Det(S)$ which (u_i, a_k)-contradicts v.

Let $M(u_i, v) = \{a_j : a_j \in A, a_j(u_i) = a_j(v)\}$ and $P(u_i, v, a_k) = \{a_k(u) : u \in U, a_j(u) = a_j(v)$ for any $a_j \in M(u_i, v)\}$. Note that $|P(u_i, v, a_k)| \geq 1$.

In [6] it is shown that in $Det(S)$ there exists a rule (u_i, a_k)-contradicting v if and only if $|P(u_i, v, a_k)| = 1$.

Hence, it follows that there exists polynomial algorithm recognizing, for a given information system $S = (U, A)$, given objects $u_i \in U$ and $v \in \mathcal{U}(S)$, and a given attribute $a_k \in A$ such that $a_k(u_i) \neq a_k(v)$, if there exist a rule from $Det(S)$, (u_i, a_k)-contradicting v.

This algorithm constructs the set $M(u_i, v)$ and the set $P(u_i, v, a_k)$. The considered rule exists if and only if $|P(u_i, v, a_k)| = 1$.

We also use the notion of *deterministic characteristic table* $D(S, v)$, where $v \in \mathcal{U}(S)$. This is a table with m columns and n rows. The entries of this table are binary (i.e., from $\{0, 1\}$). The number 0 is at the intersection of i-th row and k-th column if and only if $a_k(u_i) \neq a_k(v)$ and there exists a rule from $Det(S)$, (u_i, a_k)-contradicting v.

It is clear that there exists a polynomial algorithm which for a given information system $S = (U, A)$ and a given object $v \in \mathcal{U}(S)$ constructs the deterministic characteristic table $D(S, v)$.

2.3 Inhibitory Characteristic Tables

Let us consider a rule

$$a_{j_1} = b_1 \wedge \ldots \wedge a_{j_t} = b_t \Rightarrow a_k \neq b_k, \qquad (2)$$

where $t \geq 0$, $a_{j_1}, \ldots, a_{j_t}, a_k \in A$, $b_1, , \ldots, b_t, b_k \in \omega$, and numbers $j_1, \ldots, j_t, k$ are pairwise different. Such rules are called *inhibitory* rules. The rule (2) is *realizable for an object* $u \in \mathcal{U}(S)$ if $a_{j_1}(u) = b_1, \ldots, a_{j_t}(u) = b_t$ or $t = 0$. The rule (2) is *true for an object* $u \in \mathcal{U}(S)$ if $a_k(u) \neq b_k$ or (2) is not realizable for u. The rule (2) is *true for* S if it is true for any object from U. The rule (2) is *realizable for* S if it is realizable for at least one object from U. By $Inh(S)$ we denote the set of all inhibitory rules each of which is true for S and realizable for S.

Let $u_i \in U$, $v \in \mathcal{U}(S)$, $a_k \in A$ and $a_k(u_i) \neq a_k(v)$. We say that a rule (2) from $Inh(S)$ *contradicts* v *relative to the object* u_i *and the attribute* a_k (or (u_i, a_k)-contradicts v, for short) if (2) is realizable for u_i but is not true for v. Our aim is to recognize for given objects $u_i \in U$ and $v \in \mathcal{U}(S)$, and given attribute a_k such that $a_k(u_i) \neq a_k(v)$ if there exist a rule from $Inh(S)$, (u_i, a_k)-contradicting v.

In [6] it is shown that in $Inh(S)$ there is a rule (u_i, a_k)-contradicting v if and only if $a_k(v) \notin P(u_i, v, a_k)$.

Hence, it follows that there exists polynomial algorithm recognizing for a given information system $S = (U, A)$, given objects $u_i \in U$ and $v \in \mathcal{U}(S)$, and a given attribute $a_k \in A$ such that $a_k(u_i) \neq a_k(v)$ if there exist a rule from $Inh(S)$, (u_i, a_k)-contradicting v.

This algorithm constructs the set $M(u_i, v)$ and the set $P(u_i, v, a_k)$. The considered rule exists if and only if $a_k(v) \notin P(u_i, v, a_k)$.

In the sequel, we use the notion of *inhibitory characteristic table* $I(S, v)$, where $v \in \mathcal{U}(S)$. This is a table with m columns and n rows. The entries of this table are binary. The number 0 is at the intersection of i-th row and k-th column if and only if $a_k(u_i) \neq a_k(v)$ and there exists a rule from $Inh(S)$, (u_i, a_k)-contradicting v.

It is clear that there exists a polynomial algorithm which for a given information system $S = (U, A)$ and a given object $v \in \mathcal{U}(S)$ constructs the inhibitory characteristic table $I(S, v)$.

2.4 Evaluation Functions

Let us denote by $\mathcal{T}$ the set of binary tables, i.e., tables with entries from $\{0, 1\}$ and let us consider a partial order $\preceq$ on $\mathcal{T}$. Let $Q_1, Q_2 \in \mathcal{T}$. Then $Q_1 \preceq Q_2$ if and only if $Q_1 = Q_2$ or Q_1 can be obtained from Q_2 by changing some entries from 1 to 0.

An *evaluation function* is an arbitrary function $W : \mathcal{T} \to [0, 1]$ such that $W(Q_1) \leq W(Q_2)$ for any $Q_1, Q_2 \in \mathcal{T}$, $Q_1 \preceq Q_2$. Let us consider five examples of evaluation functions W_1, W_2, W_3^α, W_4, and W_5^α, $0 < \alpha \leq 1$. Let Q be a table from $\mathcal{T}$ with m columns and n rows. Let $L_1(Q)$ be equal to the number of 1 in Q, $L_2(Q)$ be equal to the number of columns in Q filled by 1 only, $L_3^\alpha(Q)$ be equal to the number of columns in Q with at least $\alpha \cdot 100\%$ entries equal to 1, $L_4(Q)$ be equal to the number of rows in Q filled by 1 only, and $L_5^\alpha(Q)$ be equal to the number of rows in Q with at least $\alpha \cdot 100\%$ entries equal to 1. Then $W_1(Q) = \frac{L_1(Q)}{mn}$, $W_2(Q) = \frac{L_2(Q)}{m}$, $W_3^\alpha(Q) = \frac{L_3^\alpha(Q)}{m}$, $W_4(Q) = \frac{L_4(Q)}{n}$, and $W_5^\alpha(Q) = \frac{L_5^\alpha(Q)}{n}$. Note that $W_2 = W_3^1$ and $W_4 = W_5^1$.

3 Algorithms of Classification

A decision table T is a finite table filled by nonnegative integers. Each column of this table is labeled by a conditional attribute. Rows of the table are interpreted as tuples of values of conditional attributes on some objects. Each row is labeled by a nonnegative integer, which is interpreted as the value of decision attribute. Let T contain m columns labeled by conditional attributes $a_1, \ldots, a_m$. The set $\mathcal{U}(T) = \omega^m$ will be called the *universe* for the decision table T. For each object (tuple) $v \in \mathcal{U}(T)$ integers in v are interpreted as values of attributes $a_1, \ldots, a_m$ for this object.

We consider the following classification problem: for any object $v \in \mathcal{U}(T)$ it is required to generate a value of decision attribute on v. To this end, we use D-classification algorithms and I-classification algorithms based on the deterministic characteristic table and the inhibitory characteristic table.

Let V_d be the set of values of decision attribute. For each $i \in V_d$, let us denote by S_i the information system which tabular representation consists of all rows of T, that are labeled by the decision i. Let W be an evaluation function.

D-algorithm. For a given object v and $i \in V_d$ we construct the deterministic characteristic table $D(S_i, v)$. Next, for each $i \in V_d$ we find the value of the evaluation function W for $D(S_i, v)$. For each $i \in V_d$ the value $W(D(S_i, v))$ is interpreted as the "degree" to which v belongs to S_i. As the value of decision attribute for v we choose $i \in V_d$ such that $W(D(S_i, v))$ has the maximal value. If more than one such i exists then we choose the minimal i for which $W(D(S_i, v))$ has the maximal value.

I-algorithm. For a given object v and $i \in V_d$ we construct the inhibitory characteristic table $I(S_i, v)$. Next, for each $i \in V_d$ we find the value of the evaluation function W for $I(S_i, v)$. For each $i \in V_d$ the value $W(I(S_i, v))$ is interpreted as the "degree" to which v belongs to S_i. As the value of decision attribute for v we choose $i \in V_d$ such that $W(I(S_i, v))$

has the maximal value. If more than one such i exists then we choose the minimal i for which $W(I(S_i, v))$ has the maximal value.

4 Results of Experiments

We have performed experiments with following algorithms: D-algorithm with the evaluation functions W_1, W_2, W_3^α, W_4 and W_5^α, and I-algorithm with the evaluation functions W_1, W_2, W_3^α, W_4 and W_5^α, $\alpha \in \{0.05, 0.10, \ldots, 0.90, 0.95\}$. To evaluate error rate of an algorithm on a decision table, we use either train-and-test method or cross-validation method. For evaluation functions W_3^α and W_5^α, we choose the minimal error rate among all considered α.

In our experiments, we use decision tables from [8] which were not considered in [6]. For each initial table, we choose a number of many-valued (with at least three values) attributes different from the decision attribute, and consider each such attribute as a new decision attribute. In this way, we obtain the same number of new decision tables as the number of chosen attributes.

The following decision tables from [8] were used in our experiments: balance-scale (5 attributes, 625 objects, 10-fold cross-validation, 4 new decision attributes), soybean-large (35 attributes, 307 objects in training set, 376 objects in testing set, 5 new decision attributes), post-operative (9 attributes, 90 objects, 10-fold cross-validation, 8 new decision attributes), hayes-roth (5 attributes, 132 objects in training set, 28 objects in testing set, 4 new decision attributes), lung-cancer (57 attributes, 32 objects, 10-fold cross-validation, 7 new decision attributes), solar-flare (13 attributes, 1066 objects in training set, 323 objects in testing set, 7 new decision attributes). Missing values in decision tables were filled by an algorithm from RSES2 [11]. We removed attributes of the kind "name" that are injections on the set of objects. As a result we obtained 41 decision tables (initial and new).

Table 1 includes the results of experiments. The rows of this table correspond to the evaluation functions W_1, W_2, W_3^α, W_4 and W_5^α. The column labeled by "$D < I$" contains the number of decision tables for which the error rate of D-algorithm is less than the error rate of I-algorithm. The column labeled by "$I < D$" contains the number of decision tables for which the error rate of I-algorithm is less than the error rate of D-algorithm. The column labeled by "$D = I$" contains the number of decision tables for which the error rate of D-algorithm is equal to the error rate of I-algorithm.

Table 1: Results of experiments.

Eval. func.	$D < I$	$I < D$	$D = I$
W_1	14	14	13
W_2	13	12	16
W_3^α	6	23	12
W_4	13	15	13
W_5^α	13	12	16
$\sum$	59	76	70

In particular (see Table 1), in 59 cases the error rate of D-algorithm is less than the error rate of I-algorithm, in 76 cases the error rate of I-algorithm is less than the error rate of D-algorithm, and in 70 cases D-algorithm and I-algorithm have the same error rate.

In experiments the DMES system [5] was used.

5 Conclusions

In the paper, two families of lazy classification algorithms are considered which are based on the evaluation of the number of types of true and realizable deterministic or inhibitory rules which give us arguments "against" some decisions for new objects. The reported experiments for algorithms based on inhibitory rules are, often, better than the results for algorithms based on deterministic rules. This fact can be considered as an experimental confirmation of theoretical results from [12, 14, 7] which show that the inhibitory rules can sometimes represent more knowledge encoded in information systems than the deterministic rules.

References

[1] D. W. Aha, Ed. (1997). *Lazy Learning.* Kluwer Academic Publishers, Dordrecht.

[2] J. G. Bazan (1998). Discovery of decision rules by matching new objects against data tables. In *Proceedings of the First International Conference on Rough Sets and Current Trends in Computing.* Warsaw, Poland. Lecture Notes in Artificial Intelligence, volume 1424, Springer-Verlag, Heidelberg, pages 521-528.

[3] J. G. Bazan (1998). A comparison of dynamic and non-dynamic rough set methods for extracting laws from decision table. *Rough Sets in Knowledge Discovery.* Edited by L. Polkowski and A. Skowron. Physica-Verlag, Heidelberg, pages 321-365.

[4] J. G. Bazan (1998). Methods of approximate reasoning for synthesis of decision algorithms. Ph.D. Thesis. Warsaw University (in Polish).

[5] *Data Mining Exploration System Homepage* http://www.univ.rzeszow.pl/rspn (Software).

[6] P. Delimata, M. Moshkov, A. Skowron, Z. Suraj (2007). Two families of classification algorithms. In *Proceedings of the International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing.* Lecture Notes in Computer Science, volume 4482, Springer-Verlag, Heidelberg, pages 297-304.

[7] M. Moshkov, A. Skowron, Z. Suraj. On maximal consistent extensions of information systems. In *Proceedings of the Conference Decision Support Systems*, Zakopane, Poland, December 2006 (to appear).

[8] D. J. Newman, S. Hettich, C. L. Blake, C. J. Merz (1998). *UCI Repository of machine learning databases.* University of California, Irvine, Department of Information and Computer Sciences.

[9] Z. Pawlak (1991). *Rough Sets — Theoretical Aspects of Reasoning about Data.* Kluwer Academic Publishers, Dordrecht.

[10] Z. Pawlak, A. Skowron (2007). Rudiments of rough sets. *Information Sciences*, volume 177(1), pages 3-27; Rough sets: Some extensions. *Information Sciences*, volume 177(1), pages 28-40; Rough sets and boolean reasoning. *Information Sciences*, volume 177(1), pages 41-73.

[11] *Rough Set Exploration System Homepage* http://logic.mimuw.edu.pl/∼rses.

[12] A. Skowron, Z. Suraj (1993). Rough sets and concurrency. *Bulletin of the Polish Academy of Sciences*, volume 41(3), pages 237-254.

[13] A. Skowron, Z. Suraj (1995). Discovery of concurrent data models from experimental tables: a rough set approach. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, Montreal, Canada, August, 1995, AAAI Press, Menlo Park CA, pages 288-293.

[14] Z. Suraj (2001). Some remarks on extensions and restrictions of information systems. In *Proceedings of the International Conference on Rough Sets and Current Trends in Computing.* Lecture Notes in Artificial Intelligence, volume 2005, Springer-Verlag, Heidelberg, pages 204-211.

[15] A. Wojna (2005). Analogy-based reasoning in classifier construction (Ph.D. Thesis), *Transactions on Rough Sets IV*, Lecture Notes in Computer Science, volume 3700, Springer-Verlag, Heidelberg, pages 277-374.