Two Families of Classification Algorithms

Pawel Delimata[1], Mikhail Moshkov[2], Andrzej Skowron[3], and Zbigniew Suraj[4,5]

[1] Chair of Computer Science, University of Rzeszów
Rejtana 16A, 35-310 Rzeszów, Poland
pdelimata@wp.pl
[2] Institute of Computer Science, University of Silesia
Będzińska 39, 41-200 Sosnowiec, Poland
moshkov@us.edu.pl
[3] Institute of Mathematics, Warsaw University
Banacha 2, 02-097 Warsaw, Poland
skowron@mimuw.edu.pl
[4] Chair of Computer Science, University of Rzeszów
Rejtana 16A, 35-310 Rzeszów, Poland
zsuraj@univ.rzeszow.pl
[5] Institute of Computer Science, State School of Higher Education in Jarosław
Czarnieckiego 16, 37-500 Jarosław, Poland

Abstract. In the paper, two families of lazy classification algorithms of polynomial time complexity are considered. These algorithms are based on ordinary and inhibitory rules, but the direct generation of rules is not required. Instead of this, the considered algorithms extract efficiently for a new object some information on the set of rules which is next used by a decision-making procedure.

Keywords: rough sets, decision tables, information systems, rules.

1 Introduction

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 divide the decision table T into a number of information systems S_i, $i \in D$, where D is the set of values of the decision attribute in T. For $i \in D$, 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 made for any i, and a decision i with the maximal "degree" is selected.

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 297–304, 2007.

Note that in [8] for classifying new objects it was proposed to use rules defined by conditional attributes in different decision classes.

In this paper, we consider both ordinary and inhibitory rules of the following form:

$$a_1(x) = b_1 \wedge \ldots \wedge a_t(x) = b_t \Rightarrow a_k(x) = b_k,$$

$$a_1(x) = b_1 \wedge \ldots \wedge a_t(x) = b_t \Rightarrow a_k(x) \neq b_k,$$

respectively.

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 in 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 $d(x) = i$ on the right hand side, where d is the decision attribute.

This approach was generalized by A. Wojna [9] 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].

2 Characteristic Tables

2.1 Information Systems

Let $S = (U, A)$ be an *information system*, where $U = \{u_1, \ldots, u_n\}$ is a finite nonempty set of *objects* and $A = \{a_1, \ldots, a_m\}$ 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 $\omega = \{0, 1, 2, \ldots\}$ is the set of nonnegative integers.

We also assume that the information system $S = (U, A)$ is given by a *tabular representation*, i.e., a table with m columns and n rows. Columns of the table are labeled by attributes $a_1, \ldots, a_m$. At the intersection of i-th row and j-th column the value $a_j(u_i)$ is included. For $i = 1, \ldots, n$ we identify any object $u_i \in U$ with the tuple $(a_1(u_i), \ldots, a_m(u_i))$, i.e., the i-th row of the tabular representation of the information system S.

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 $\mathcal{U}(S) \setminus U$. 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 Ordinary Characteristic Tables

Let us consider a rule

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

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 will be called *ordinary* rules. The rule (1) will be called *realizable for an object* $u \in \mathcal{U}(S)$ if $a_{j_1}(u) = b_1, \ldots, a_{j_t}(u) = b_t$. The rule (1) will be called *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) will be called *true for* S if it is true for any object from U. The rule (1) will be called *realizable for* S if it is realizable for at least one object from U. Denote by $Ord(S)$ the set of all ordinary 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 $Ord(S)$ *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 $Ord(S)$ 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

$$P(u_i, v, a_k) = \{a_k(u) : u \in U, a_j(u) = a_j(v) \text{ for any } a_j \in M(u_i, v)\}.$$

Note that $|P(u_i, v, a_k)| \geq 1$.

Proposition 1. *Let* $S = (U, A)$ *be an information system,* $u_i \in U$, $v \in \mathcal{U}(S)$, $a_k \in A$ *and* $a_k(u_i) \neq a_k(v)$. *Then, in* $Ord(S)$ *there exists a rule* (u_i, a_k)-*contradicting* v *if and only if* $|P(u_i, v, a_k)| = 1$.

Proof. Let $|P(u_i, v, a_k)| = 1$ and $P(u_i, v, a_k) = \{b\}$. In this case, the rule

$$\bigwedge_{a_j \in M(u_i, v)} a_j(x) = a_j(v) \Rightarrow a_k(x) = b, \tag{2}$$

belongs to $Ord(S)$, is realizable for u_i, and is not true for v, since $a_k(v) \neq a_k(u_i) = b$. Therefore, (2) is a rule from $Ord(S)$, which (u_i, a_k)-contradicts v.

Let us assume that there exists a rule (1) from $Ord(S)$ (u_i, a_k)-contradicting v. Since (1) is realizable for u_i and is not true for v, we have $a_{j_1}, \ldots, a_{j_t} \in M(u_i, v)$. Also (1) is true for S. Hence, the rule

$$\bigwedge_{a_j \in M(u_i, v)} a_j(x) = a_j(v) \Rightarrow a_k(x) = b_k$$

is true for S. Therefore, $P(u_i, v, a_k) = \{b_k\}$ and $|P(u_i, v, a_k)| = 1$. $\square$

From Proposition 1 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 $Ord(S)$ (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 *ordinary characteristic table* $O(S, v)$, 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 $Ord(S)$ (u_i, a_k)-contradicting v.

From Proposition 1 it follows 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 ordinary characteristic table $O(S, v)$.

2.3 Inhibitory Characteristic Tables

Let us consider a rule

$$a_{j_1}(x) = b_1 \wedge \ldots \wedge a_{j_t}(x) = b_t \Rightarrow a_k(x) \neq b_k, \tag{3}$$

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 (3) will be called *realizable for an object* $u \in \mathcal{U}(S)$ if $a_{j_1}(u) = b_1, \ldots, a_{j_t}(u) = b_t$. The rule (3) will be called *true for an object* $u \in \mathcal{U}(S)$ if $a_k(u) \neq b_k$ or (3) is not realizable for u. The rule (3) will be called *true for* S if it is true for any object from U. The rule (3) will be called *realizable for* S if it is realizable for at least one object from U. Denote by $Inh(S)$ 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 (3) 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 (3) 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.

Proposition 2. *Let $S = (U, A)$ be an information system, $u_i \in U$, $v \in \mathcal{U}(S)$, $a_k \in A$ and $a_k(u_i) \neq a_k(v)$. Then in $Inh(S)$ there is a rule (u_i, a_k)-contradicting v if and only if $a_k(v) \notin P(u_i, v, a_k)$.*

Proof. Let $a_k(v) \notin P(u_i, v, a_k)$. In this case, the rule

$$\bigwedge_{a_j \in M(u_i, v)} a_j(x) = a_j(v) \Rightarrow a_k(x) \neq a_k(v), \tag{4}$$

belongs to $Inh(S)$, is realizable for u_i, and is not true for v. Therefore, (4) is a rule from $Inh(S)$ (u_i, a_k)-contradicting v.

Let us assume that there exists a rule (3) from $Inh(S)$, (u_i, a_k)-contradicting v. In particular, it means that $a_k(v) = b_k$. Since (3) is realizable for u_i and is not true for v, we have $a_{j_1}, \ldots, a_{j_t} \in M(u_i, v)$. Since (3) is true for S, the rule

$$\bigwedge_{a_j \in M(u_i,v)} a_j(x) = a_j(v) \Rightarrow a_k(x) \neq b_k$$

is true for S. Therefore, $a_k(v) \notin P(u_i, v, a_k)$. □

From Proposition 2 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.

From Proposition 2 it follows 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 three examples of evaluation functions W_1, W_2 and W_3^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, and $L_3^\alpha(Q)$ is defined as the number of columns in Q with at least $\alpha \cdot 100\%$ entries equal to 1. Then

$$W_1(Q) = \frac{L_1(Q)}{mn}, \quad W_2(Q) = \frac{L_2(Q)}{m}, \quad \text{and} \quad W_3^\alpha(Q) = \frac{L_3^\alpha(Q)}{m}.$$

It is clear that $W_2 = W_3^1$. Let $S = (U, A)$ be an information system and $v \in \mathcal{U}(S)$. Note that if $v \in U$ then $W_1(O(S, v)) = W_2(O(S, v)) = W_3^\alpha(O(S, v)) = 1$ and $W_1(I(S, v)) = W_2(I(S, v)) = W_3^\alpha(I(S, v)) = 1$ for any α $(0 < \alpha \leq 1)$.

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 compute a value of decision attribute on v. To this end, we use O-classification algorithms and I-classification algorithms based on the ordinary characteristic table and the inhibitory characteristic table.

Let D be the set of values of decision attribute. For each $i \in D$, 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.

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

I-algorithm. For a given object v and $i \in D$ we construct the inhibitory characteristic table $I(S_i, v)$. Next, for each $i \in D$ we find the value of the evaluation function W for $I(S_i, v)$. For each $i \in D$ 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 $i \in D$ 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: O-algorithm with the evaluation functions W_1, W_2 and W_3^α, and I-algorithm with the evaluation functions W_1, W_2 and W_3^α. To evaluate error rate of an algorithm on a decision table we use either train-and-test method or cross-validation method.

The following decision tables from [6] were used in our experiments: monk1 (6 conditional attributes, 124 objects in training set, 432 objects in testing set), monk2 (6 conditional attributes, 169 objects in training set, 432 objects in testing set), monk3 (6 conditional attributes, 122 objects in training set, 432 objects in testing set), lymphography (18 conditional attributes, 148 objects, 10-fold cross-validation), diabetes (8 conditional attributes, 768 objects, 12-fold cross-validation, attributes are discretized by an algorithm from RSES2 [7]), breast-cancer (9 conditional attributes, 286 objects, 10-fold cross-validation), primary-tumor (17 conditional attributes, 339 objects, 10-fold cross-validation, missing values are filled by an algorithm from RSES2).

Table 1 contains results of experiments (error rates) for O-algorithm and I-algorithm with the evaluation functions W_1 and W_2, and for each of the

Table 1. Results of experiments with evaluation functions W_1 and W_2

Decision table	O-alg., W_1	O-alg., W_2	I-alg., W_1	I-alg., W_2	err. rates [3]
monk1	0.292	0.443	0.114	0.496	0.000–0.240
monk2	0.260	0.311	0.255	0.341	0.000–0.430
monk3	0.267	0.325	0.119	0.322	0.000–0.160
lymphography	0.272	0.922	0.215	0.922	0.157–0.380
diabetes	0.348	0.421	0.320	0.455	0.224–0.335
breast-cancer	0.240	0.261	0.233	0.268	0.220–0.490
primary-tumor	0.634	0.840	0.634	0.846	0.550–0.790
average err. rate	0.330	0.503	0.270	0.521	0.164–0.404

considered tables. The last row contains average error rates. The last column contains some known results – the best and the worst error rates for algorithms compared in the survey [3].

The obtained results show that the evaluation function W_1 is noticeably better than the evaluation function W_2, and I-algorithm with the evaluation function W_1 is better than O-algorithm with the evaluation function W_1. The last result follows from the fact that the inhibitory rules have much higher chance to have larger support in the decision tables than the ordinary rules.

The outputs returned by I-algorithm with the evaluation function W_1 for each of decision tables are comparable with the results reported in [3], but are worse than the best results mentioned in [3].

Table 2 contains results of experiments (error rates) for two types of algorithms: O-algorithm with the evaluation function W_3^α, and I-algorithm with the evaluation function W_3^α, where $\alpha \in \{0.50, 0.55, \ldots, 0.95, 1.00\}$. For each decision table and for algorithms of each type the best result (with the minimal error rate) and the corresponding α to this result are presented in the table. The last row contains average error rates. The last column contains some known results – the best and the worst error rates for algorithms discussed in [3].

The obtained results show that the use of the parameterized evaluation functions W_3^α, where $\alpha \in \{0.50, 0.55, \ldots, 0.95, 1.00\}$, makes it possible to improve

Table 2. Results of experiments with evaluation functions W_3^α

Decision table	O-alg., W_3^α	α	I-alg., W_3^α	α	err. rates [3]
monk1	0.172	0.95	0.195	0.85	0.000–0.240
monk2	0.301	0.95	0.283	0.95	0.000–0.430
monk3	0.325	1.00	0.044	0.65	0.000–0.160
lymphography	0.293	0.55	0.272	0.65	0.157–0.380
diabetes	0.421	1.00	0.351	0.95	0.224–0.335
breast-cancer	0.229	0.80	0.225	0.70	0.220–0.490
primary-tumor	0.658	0.75	0.655	0.70	0.550–0.790
average err. rate	0.343		0.289		0.164–0.404

the performance of I-algorithm with the evaluation function W_1 for tables monk3 and breast-cancer.

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 rules which give us "negative" information about new objects. In the further investigations we are planning to consider also the number of types of true rules which give us "positive" information about new objects. Also we are planning to consider more wide parametric families of evaluation functions which will allow to learn classification algorithms.

References

1. Aha, D.W. (Ed.): Lazy Learning. Kluwer Academic Publishers, Dordrecht, Boston, London, 1997
2. Bazan, J.G.: Discovery of decision rules by matching new objects against data tables. Proceedings of the First International Conference on Rough Sets and Current Trends in Computing. Warsaw, Poland. Lecture Notes in Artificial Intelligence **1424**, Springer-Verlag, Heidelberg (1998) 521-528
3. Bazan, J.G.: 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 (1998) 321–365
4. Bazan, J.G.: Methods of approximate reasoning for synthesis of decision algorithms. Ph.D. Thesis. Warsaw University (1998) (in Polish)
5. Data Mining Exploration System Homepage http://www.univ.rzeszow.pl/rspn (Software)
6. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases http://www.ics.uci.edu/∼mlearn/MLRepository.html. University of California, Irvine, Department of Information and Computer Sciences (1998)
7. Rough Set Exploration System Homepage http://logic.mimuw.edu.pl/∼rses
8. Skowron, A., Suraj, Z.: Discovery of concurrent data models from experimental tables: a rough set approach. Proceedings of the First International Conference on Knowledge Discovery and Data Mining, Montreal, Canada, August, 1995, AAAI Press, Menlo Park CA (1995) 288–293
9. Wojna, A.: Analogy-based reasoning in classifier construction (Ph.D. Thesis), Transactions on Rough Sets IV, Lecture Notes in Computer Science **3700**, Springer-Verlag, Heidelberg (2005) 277–374