System *ADReD* for Discovering Rules based on Hyperplanes

Zbigniew W. Raś[1,2], Agnieszka Dardzińska[3], and Xingzhen Liu[1]

[1] UNC-Charlotte, Department of Computer Science, Charlotte, N.C. 28223, USA
[2] Polish Academy of Sciences, Institute of Computer Science, Ordona 21, Warsaw, Poland
[3] Bialystok Technical Univ., Dept. of Mathematics, ul. Wiejska 45A, 15-351, Bialystok, Poland

Abstract. Decision table describing n objects in terms of k classification attributes and one decision attribute can be seen as a collection of n points in k-dimensional space. Each point is classified either as positive or negative. The goal of this paper is to present an efficient strategy for constructing possibly the smallest number of hyperplanes so each area surrounded by them contains a group of points, mostly of the same type (either positive or negative). A threshold value given by user, uniquely defines what we mean by *mostly*. The strategy presented in [3] shows how to construct a possibly smallest number of pairs of hyperplanes, surrounding any dense cluster of objects, which intersection is a line orthogonal to and intersecting with one of the axes. In this paper that constraint is softened and these hyperplanes are built more independently. The main procedure starts with partitioning all negative objects into dense clusters. The same step is repeated for all positive objects also dividing them into dense clusters. To learn a negative rule, we take all objects in one of this negative clusters jointly with all positive objects. The algorithm, presented in this paper, constructs a minimal number of hyperplanes needed to build classification part of a rule describing this negative cluster. The same procedure is repeated for all the remaining negative clusters. Rules describing positive clusters are constructed the same way. Taking the Wisconsin Breast Cancer Database with 699 instances, as an example, we show that the overall support and confidence of rules, extracted from that database, using our strategy is much higher than the confidence and support of rules obtained using methods based on hyperplanes parallel to axes (See5, Rosetta).

1 Introduction

Some, more sophisticated, data representations may lead us to much better results than the results we get by taking representations seen normally as standard. Clearly, such data representations are often not easy to find. Without no doubts this fact is true in knowledge discovery area where from a sample decision table we need to learn the decision attribute in terms of the remaining attributes, called classification attributes. The problem is, what attributes should be used for data representation and next for knowledge discovery to get rules of possibly highest confidence and highest support.

In this paper, we assume that the decision attribute is binary and objects represented as points in k-dimensional space (k is the number of classification attributes) belong to either positive or negative class. The outcome of learning process depends on the distribution of positive and negative objects in the example space. When we look for an optimal partition of objects into positive and negative regions, most of the learning strategies allow us to build hyperplanes which are only parallel to axes. Clearly, by taking hyper-planes not only parallel to axis we are increasing our chance to construct rules (both certain and threshold-based) of possibly much higher support. In [3], [2] it was shown how to construct new attributes for each object in the decision table. For each object, these attributes are angles between axes and lines crossing that object and each of the axes. Extracted rules based on these new attributes have usually higher confidence and support than rules built from original attributes.

In this paper, we use a similar method to represent points in the example space. However, we do not build a new table representation for the objects stored in the decision table. Initially, a number of hyperplanes is constructed for each pair of axes, each dividing the objects space into two parts. The first part contains only negative objects and their number should be as large as possible. The expression describing the other part is treated as one of the atomic relations for our rules discovery algorithm **ADReD** which has some similarity with LERS (see [4]). **ADReD** requires from the user to provide a threshold value for a minimum acceptable confidence of rules.

We start with some basic definitions needed in this paper.

2 Decision Systems

This section starts with the definition of an information system and a decision system. Next, the notion of a rule, its support and confidence is recalled.

Definition 1:
By an *information system* (see [6]) we mean a triple $S = (X, A, V)$ where:

- X is a nonempty, finite set of objects,
- A is a nonempty, finite set of attributes,
- $V = \bigcup\{V_a : a \in A\}$ is a set of values of attributes from A.
- $a : X \longrightarrow V_a$ is a function for every $a \in A$.

Information systems can be seen as generalizations of decision systems. In any decision system together with the set of attributes a partition of that set into conditions and decisions is given. For simplicity reason, we consider decision systems with only one decision. Therefore, the definition of a decision system is formed as follows.

Definition 2:
By a *decision system* we mean any information system $S = (X, A \cup \{d\}, V)$,

where $d \notin A$ is a distinguished attribute called the *decision*. Attributes in A are called *classification* attributes.

Definition 3:
By a set of *positive terms* for S we mean a least set T such that:

- $\mathbf{0}, \mathbf{1} \in T$,
- $w \in T$ for any $w \in V$,
- if $t_1, t_2 \in T$ then $(t_1 + t_2), (t_1 * t_2) \in T$.

Definition 4:
Term $t = t_1 * t_2 * ... * t_n$ is called simple if t is positive and $(\forall j \in \{1, 2, ..., n\})[t_j \in V]$.

Definition 5:
Standard interpretation M of terms in S is defined as follows:

- $M(\mathbf{0}) = \emptyset$, $M(\mathbf{1}) = X$,
- $M(w) = \{x \in X : w \in a(x)\}$ for any $w \in V_a$,
- if t_1, t_2 are terms, then:
  $M(t_1 + t_2) = M(t_1) \cup M(t_2)$,
  $M(t_1 * t_2) = M(t_1) \cap M(t_2)$.

X	a	b	d
x_1	60	60	$-$
x_2	80	60	$+$
x_3	100	60	$+$
x_4	130	60	$-$
x_5	60	50	$+$
x_6	130	50	$+$
x_7	80	40	$-$
x_8	100	40	$+$
x_9	130	40	$-$
x_{10}	100	30	$-$

Table 1. Decision System S

Definition 6:
By a rule in a decision system S we mean any expression of the form $t \longrightarrow d_1$, where t is a simple term for S and $d_1 \in Dom(d)$.

Definition 7:
Support of a rule $t \longrightarrow d_1$ (denoted as $sup(t \longrightarrow d_1)$) in S is defined as $sup(t * d_1)$ which means the number of objects in S having property $t * d_1$.

Definition 8:
Confidence of a rule $t \longrightarrow d_1$ (denoted as $conf(t \longrightarrow d_1)$) in S is defined as $sup(t * d_1)/sup(t)$.

In this paper we assume that all classification attributes in S are numerical. For simplicity reason, only two attributes a and b are taken into consideration in the example below.

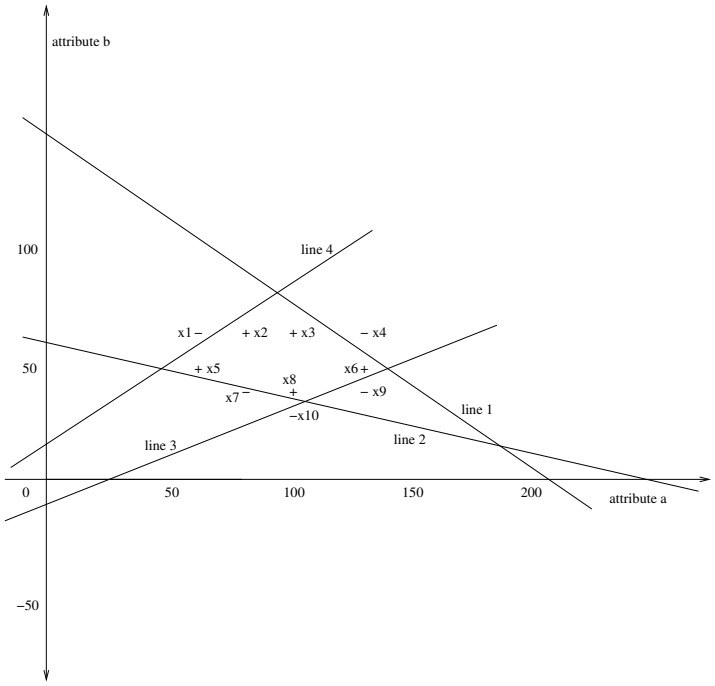

Fig. 1. Hyperplanes surrounding positive points

Example 1:
Assume that $S = (X, A, V)$ is an information system (see Table 1), where: $X = \{x_1, x_2, ..., x_{10}\}$, $A = \{a, b\} \cup \{d\}$, and domains of attributes are defined as: *dom(a)=[0,200], dom(b)=[0,200], dom(d)=+,-.*

The above decision system (also used, as an example, in paper [3]) can be represented as a collection of 10 points in 2-dimensional space. These points are partitioned into two classes (positive and negative) representing 2 values of the attribute d. Their graphical representation is given in Fig. 1. Also, you

can notice that four lines are sufficient to surround all positive objects in S keeping all negative objects outside.

Applying, for instance, system LERS to S (see [4]), the following certain rules are extracted from S:
$[(b,50) \rightarrow (d,+)]$, $[(a,100) * (b,60) \rightarrow (d,+)]$, $[(a,100) * (b,40) \rightarrow (d,+)]$, $[(a,100) * (b,60) \rightarrow (d,+)]$
and
$[(b,30) \rightarrow (d,-)]$, $[(a,60) * (b,60) \rightarrow (d,-)]$, $[(a,80) * (b,40) \rightarrow (d,-)]$, $[(a,130) * (b,40) \rightarrow (d,-)]$, $[(a,130) * (b,60) \rightarrow (d,-)]$.

These rules have very small support which is due both to the distribution of 10 points in S and $LERS$ quantization strategy which does not support construction of rectangles not parallel to axes a and b. Slezak and Wroblewski proposed to construct rules using new attributes which are linear combinations of existing ones (see [8]). This approach was continued by Bazan in [1]. Said and Dardzinska in [7] proposed a strategy for classification and automatic identification of Arabic characters. Their strategy is based on the construction of new attributes with values being angles between lines connecting some distinguished points in Arabic characters. Similar idea was used by Dardzinska in [2] and Dardzinska and Ras in [3] to find a new representation for objects in a decision system. The strategy presented in this paper has some similarity with the method proposed in [3] as far as the representation of data but they differ entirely in their approaches to rules extraction from S.

3 Construction of Hyperplanes

Before we present the steps of our new hyperplane-based rule discovery strategy, let us go back to Figure 1 and the same to the 2-dimensional representation of objects in S. Our goal is to surround its all positive objects by minimal number of lines each originating from one of the two axes (intersecting with it) corresponding to attributes a and b. For instance, lines (*line1*, *line2*, *line3* and, *line4*) originate from axis corresponding to attribute a. It can be easily checked that minimum four lines are needed in our example to separate all positive points from the points marked negative in S. In practice, some threshold value which gives the maximal percentage of negative points allowed in a positive cluster is given. Clearly the same threshold value is usually used for the maximal percentage of positive points allowed in a negative cluster. Also, in this paper both thresholds values will be the same. For instance, if the threshold value is set up as 1/5, then one of the lines *line1*, *line2* or *line4* is not needed to describe the cluster.

What do we mean by an optimal *line1* intersecting with axis corresponding to attribute a and how such a line can be constructed? Our goal is to maximize the number of negative points in the subspace described by $line1 > 0$

while keeping all positive points in the subspace described by *line1* ≤ 0. If a successful candidate for *line1* is found, it is called optimal. We denote its intersection point with the axis a by $a(n_1)$ and the corresponding angle by $\omega(b, a(n_1))$ (see Figure 2). Similarly, in search for *line2* intersecting with axis corresponding to attribute a, our goal is to maximize the number of negative points in the subspace described by *line2* < 0 while keeping all positive points in the subspace *line2* ≥ 0. When a successful candidate for *line2* is found it is called again optimal. We denote its intersection point with the axis a by $a(m_1)$ and the corresponding angle by $\omega(b, a(m_1))$. Clearly, we may have a number of optimal candidates constructed for *line1* and *line2*. Some of them can be semantically equivalent (see the definition below).

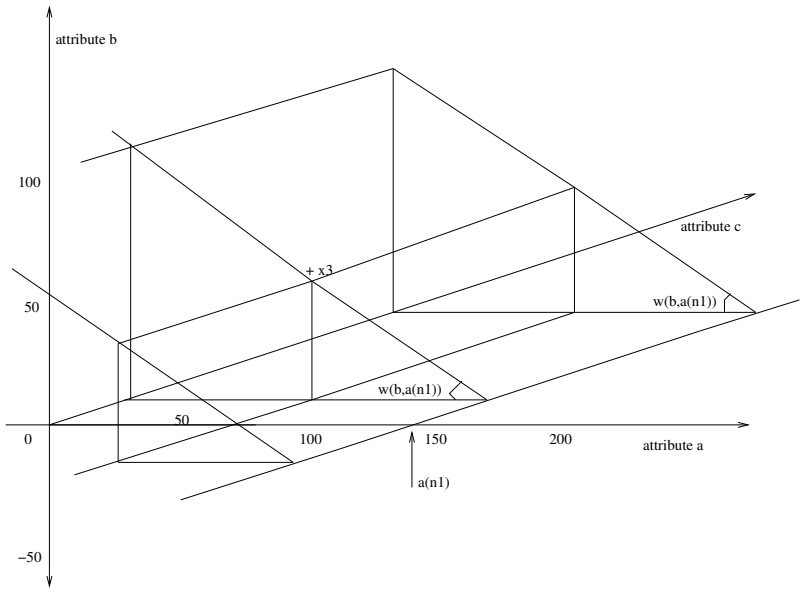

Fig. 2. Construction of a hyperplane

Definition 9:
Let $S = (X, A \cup \{d\}, V)$ be a decision system where $card(A) = k$. Assuming that $h_1 = 0$, $h_2 = 0$ are equations representing two hyperplanes in k-dimensional space, we say that these hyperplanes are semantically equivalent in S if the set of objects in S satisfying the relation $h_1 \geq 0$ is equal to the set of objects satisfying the relation $h_2 \geq 0$. We also say that a set of hyperplanes is orthogonal in S, if it does not contain two hyperplanes which are semantically equivalent in S.

It can be easily checked that the set $\{line1, line2, line3, line4\}$ is orthogonal in S . There are many semantically equivalent maximal orthogonal sets of hyperplanes in S. Clearly, finding one of them is sufficient for our purpose.

Let us use variable z_1 to denote axis representing attribute a and variable z_2 to denote axis representing attribute b. Assuming that the angle between *line1* and axis z_1 is $\omega(z_2, z_1(n_1))$ where $z_1(n_1)$ is the intersection point between these two lines, *line1* has the equation: $[z_1 - z_1(n_1)] \cdot \sin[\omega(z_2, z_1(n_1))] + z_2 \cdot \cos[\omega(z_2, z_1(n_1))] = 0$. Clearly, if we properly change the position of the point $z_1(n_1)$ on axis z_1, we get equations of the remaining 3 lines (*line2*, *line3*, and *line4*).

We are ready to present the strategy of constructing hyperplanes in k-dimensional space. Assume that $S = (X, A \cup \{d\}, V)$ is a decision system, where $A = \{a_1, a_2, ..., a_k\}$ and $V_d = \{+, -\}$. We use variable z_i to denote axis representing attribute a_i, for any $i \in \{1, 2, ..., k\}$. Now, for any pair z_i, z_j of axes, an orthogonal set $H(i, j)$ of hyperplanes is constructed. The corresponding strategy is outlined below.

Assume first that $z_j(n_1)$ is a point on axis z_j. For any object $u_m = (u_{m1}, u_{m2}, ..., u_{mi}, ..., u_{mj}, ..., u_{mk}) \in X$, two points

$p_1(u_m) = (u_{m1}, u_{m2}, ..., 0, ...u_{mj}, ..., u_{mk})$ and
$p_2(u_m, n_1) = (u_{m1}, u_{m2}, ..., 0, ..., z_j(n_1), ..., u_{mk})$

are constructed (see Figure 2). Let $L(u_m, p_2(u_m, n_1))$ be the line connecting object u_m with point $p_2(u_m, n_1)$ and $L(p_1(u_m), p_2(u_m, n_1))$ be the line connecting point $p_1(u_m)$ with the point $p_2(u_m, n_1)$. By $\omega(u_m, z_i, z_j(n_1))$ we denote the angle between these two lines. This angle is computed for all objects $u_m \in X$.

Now, we calculate two threshold values:

- $max(n_1) = max\{\omega(u_m, z_i, z_j(n_1)) : u_m \in X \wedge d(u_m) = +\}$
- $min(n_1) = min\{\omega(u_m, z_i, z_j(n_1)) : u_m \in X \wedge d(u_m) = +\}$.

These thresholds are used to compute two sets:

- $U(max(n_1)) = \{u_m \in X : \omega(u_m, z_i, z_j(n_1)) > max(n_1)\}$
- $U(min(n_1)) = \{u_m \in X : \omega(u_m, z_i, z_j(n_1)) < min(n_1)\}$.

So, $U(max(n_1))$ is the set of all negative objects in X which satisfy the relation $[z_j - z_j(n_1)] \cdot \sin[max(n_1)] + z_i \cdot \cos[max(n_1)] > 0$.

Similarly, $U(min(n_1))$ is the set of all negative objects in X which satisfy the relation $[z_j - z_j(n_1)] \cdot \sin[min(n_1)] + z_i \cdot \cos[min(n_1)] < 0$.

The set $H(i, j)$ is a maximal orthogonal subset of the set

$\{[z_j - z_j(n_1)] \cdot \sin[max(n_1)] + z_i \cdot \cos[max(n_1)] = 0 : n_1 \in N\} \cup$
$\{[z_j - z_j(n_1)] \cdot \sin[min(n_1)] + z_i \cdot \cos[min(n_1)] = 0 : n_1 \in N\}$,

where N are integers. Assuming that the testing points $z_j(n_1)$ for the axis z_j are given, there is an efficient strategy for computing the set $H(i, j)$.

Now, we take the maximal orthogonal subset H of the set $\bigcup\{H(i, j) : 1 \leq i \leq k \wedge 1 \leq j \leq k\}$. In the next step of our procedure, each hyperplane

$[z_j - z_j(n_1)] \cdot \sin[max(n_1)] + z_i \cdot \cos[max(n_1)] = 0$ in H is replaced by the relation $[z_j - z_j(n_1)] \cdot \sin[max(n_1)] + z_i \cdot \cos[max(n_1)] \leq 0$
and each hyperplane

$[z_j - z_j(n_1)] \cdot \sin[min(n_1)] + z_i \cdot \cos[min(n_1)] = 0$ in H is replaced by the relation $[z_j - z_j(n_1)] \cdot \sin[min(n_1)] + z_i \cdot \cos[min(n_1)] \geq 0$.

The resulting set of relations is denoted by H_R and it is treated as the set of atomic expressions for the LERS-type procedure generating rules describing values of attribute d in terms of attributes from A in the decision system S.

Now, assuming that
$h = [[z_j - z_j(n_1)] \cdot \sin[max(n_1)] + z_i \cdot \cos[max(n_1)] \leq 0]$, its supporting set $Sup(h)$ is equal to $X - U(max(n_1))$.

Assuming that
$h = [[z_j - z_j(n_1)] \cdot \sin[min(n_1)] + z_i \cdot \cos[min(n_1)] \geq 0]$, its supporting set $Sup(h)$ is equal to $X - U(min(n_1))$.

Finally, by $\lambda(Sup(h))$ we mean $[card(\{x \in Sup(h) : d(x) = -\}]/[card(Sup(h))]$.

Algorithm ADReD$(S, H_R(S), \lambda)$,
Input
Decision system $S = (X, A \cup \{d\}, V)$, where $V_d = \{+, -\}$,
Set of relations $H_R(S)$ for S constructed above
Minimal confidence threshold value λ.
Output
Set of Rules RL
begin
$RL := \emptyset$; $k := card(A)$; $H(1) := H_R(S)$;
for all $h \in H_R(S)$ **do** $mark(h) := F$;
for all $j \leq k$ **do** $H(j) = \emptyset$;
begin
for all $h \in H_R(S)$ **do**
  if $\lambda(Sup(h)) \leq \lambda$ **then**
    begin
    $mark(h) := T$;
    $H(1) := H(1) - \{h\}$;
    $RL := RL \cup \{h \longrightarrow +\}$
    end
$j := 1$;
while $j \leq k - 1$ **do**
  for all pairs $(h_1, h_2) \in (H(j), H(1))$ **do**
    if $\lambda[Sup(h_1) \cap Sup(h_2)] \leq \lambda$ **then** $RL := RL \cup \{[h_1 \star h_2] \longrightarrow +\}$
      else if $Sup(h_1) \neq Sup(h_1 \star h_2)$
        then $H(j+1) := H(j+1) \cup \{h_1 \star h_2\}$
  $j := j + 1$
end

There are two basic options for constructing $H_R(S)$. This orthogonal set of hyperplanes can be defined either as the collection of all sets in $\{U(max(n_1)) : n_1 \in N\} \cup \{U(min(n_1) : n_1 \in N\}$ or as the collection of only maximal sets in $\{U(max(n_1)) : n_1 \in N\} \cup \{U(min(n_1) : n_1 \in N\}$ under set-theoretical inclusion. In the second case, the time complexity of the algorithm *ADReD* may significantly decrease but the generated rules no longer have to be optimal.

4 Test Results

Algorithm *ADReD* was tested on the Wisconsin Breast Cancer Decision Table available at [*ftp://ftp.ics.uci.edu/pub/machine-learning-databases/breast-cancer-wisconsin/*] which contains 699 objects described by 10 classification attributes and one decision attribute of two values: Benign (458 objects), Malignant (241 objects). We tested *ADReD* taking initially the collection of all sets in $\{U(max(n_1)) : n_1 \in N\} \cup \{U(min(n_1) : n_1 \in N\}$ as $H_R(S)$ denoted in this section by $All - H_R(S)$ and next the collection of only maximal sets in $\{U(max(n_1)) : n_1 \in N\} \cup \{U(min(n_1) : n_1 \in N\}$ as $H_R(S)$ denoted in this section by $Max - H_R(S)$.

The list of all 11 attributes and their domains, used in the Wisconsin Breast Cancer Decision Table, is given below:

$Attr\_Number$	$Attr\_Name$	$Domain$
1	$Sample\_Code\_Number$	Id_Number
2	$Clump\_Thickness$	$1-10$
3	$Uniformity\_of\_Cell\_Size$	$1-10$
4	$Uniformity\_of\_Cell\_Shape$	$1-10$
5	$Marginal\_Adhesion$	$1-10$
6	$Single\_Epithelial\_Cell\_Size$	$1-10$
7	$Bare\_Nuclei$	$1-10$
8	$Bland\_Chromatin$	$1-10$
9	$Normal\_Nucleoli$	$1-10$
10	$Mitoses$	$1-10$
11	$Class$	$\{benign, malignant\}$

Table 2. List of Attributes

We used SAS Enterprise Miner to partition the data into two clusters given below:

Benign cluster (485 objects): (450 benign and 35 malignant).
Malignant cluster (214 objects): (206 malignant and 8 benign).

The malignant cluster only has 8 benign objects in it, so there is not much room for improvement. Hence, we have chosen the benign cluster as the test cluster. The current confidence is 450/485 = 0.9125. By applying *ADReD* program, we can exclude more malignant samples from benign cluster. The next two tables show test results obtained from *ADReD*.

$Threshold\,\lambda$	0.04	0.03	0.02	0.01
$Number\_of\_rules\_generated$	1175	319	126	1
$Best\_rule\_confidence$	0.9825	0.9890	0.9912	0.9912
$Searching\_time\_in\_seconds$	9.5	8	6	5

Table 3. *ADReD* based on $All - H_R(S)$

$Threshold\,\lambda$	0.04	0.03	0.02	0.01
$Number\_of\_rules\_generated$	808	279	116	1
$Best\_rule\_confidence$	0.9825	0.9890	0.9912	0.9912
$Searching\_time\_in\_seconds$	6.6	6.5	5	4.7

Table 4. *ADReD* based on $Max - H_R(S)$

Clearly the strategy based on $Max - H_R(S)$ is faster, although it cannot generate as many rules as the strategy based on $All - H_R(S)$. For this dataset both strategies can find the best rule which is able to exclude 31 out of 35 malignant objects in the benign cluster.

Now, we show a sample rule generated by *ADReD*. Coordinates (attributes) are denoted here by $x0, x1, x2, x3, x4, x5, x6, x7, x8, x9$:

Rule No. 1
Confidence = 0.9911894273127754,
Classification Error Rate = 0.00881057268722467,
Number of negative objects excluded = 31,
Excluded negative objects index: [200, 168, 30, 214, 40, 72, 31, 16, 35, 213, 49, 59, 296, 73, 169, 70, 29, 157, 42, 100, 178, 11, 18, 145, 39, 44, 165, 38, 121, 37, 320]

$[(x5+14)*\sin(48)+x0*\cos(48) \leq 0] \wedge [(x7+20)*\sin(122)+x6*\cos(122) \geq 0] \wedge [(x5+20)*\sin(38)+x7*\cos(38) \leq 0] \wedge [(x4+20)*\sin(177)+x5*\cos(177) \leq$

$0] \wedge [(x3+12)*\sin(59)+x2*\cos(59) \leq 0] \wedge [(x8+2)*\sin(42)+x0*\cos(42) \leq 0] \wedge [(x6+14)*\sin(63)+x0*\cos(63) \leq 0] \rightarrow [Class = 2]$ *(benign)*

Finally, we have compared our system *ADReD* with *See*5. On the tested database the classification error rate for *ADReD* is 0.88%, whereas the error rate for *See*5 is 2.1%.

References

1. Bazan, J., Szczuka, M., Wroblewski, J. (2002) A new version of rough set exploration system, in **Proceedings of RSCTC-02**, in Malvern, PA, LNAI 2475, Spinger-Verlag, 397-404
2. Dardzinska, A. (2003) Rule discovery based on new attributes construction, in **Proceedings of RSKD Workshop**, April 11-12, Warsaw, Poland, 76-84
3. Dardzinska, A, Ras, Z.W. (2003) How to increase support of rules to be discovered from DB, in **Proceedings of the Symposium on Methods of AI** (AI-METH2003), Silesian University of Technology, Gliwice, Poland, 123-128
4. Grzymala-Busse, J. (1997) A new version of the rule induction system LERS, in **Fundamenta Informaticae**, Vol. 31, No. 1, 27-39
5. Manber, U. (1989) Introduction to algorithms, a creative approch, Addison Wesley
6. Pawlak, Z. (1991) Rough sets-theoretical aspects of reasoning about data, Kluwer, Dordrecht
7. Said, K., Dardzinska, A. (2000) Cursive letters language processing: Muqla model and Toeplitz matrices approach, in **Proceedings of the 4-th International Conference** (FQAS-00), Warsaw, Poland, October 25-28, Physica-Verlag, 326-333
8. Slezak, D., Wroblewski, J. (1999) Classification algorithms based on linear combinations of features, in **Proceedings of PKDD-99**, LNAI 1704, Springer-Verlag, 548-553