MATRIX: AN INCREMENTAL ALGORITHM FOR INFERRING IMPLICATIVE RULES FROM EXAMPLES BASED ON GOOD CLASSIFICATION TESTS

Xenia Naidenova

Abstract: A new incremental algorithm MATRIX is proposed for inferring implicative logical rules from examples. The concept of a good diagnostic test for a given set of positive examples lies in the basis of this algorithm.

Keywords: learning logical rules from examples, machine learning, inductive inference, good diagnostic test

Introduction

Our approach to machine learning problems is based on the concept of a good diagnostic (classification) test. This concept has been advanced firstly in the framework of inferring functional and implicative dependencies from relations (Naidenova and Polegaeva, 1986). But later the fact has been revealed that the task of inferring all good diagnostic tests for a given set of positive and negative examples can be formulated as the search of the best approximation of a given classification on a given set of examples and that it is this task that all well known machine learning problems can be reduced to (Naidenova, 1996).

This paper is organized as follows. The concept of a good diagnostic test is defined and the problem of inferring all good diagnostic tests for a given classification on a given set of examples is formulated. The next section contains the description of a mathematical model underlying algorithms proposed. Then we give a decomposition of learning algorithms into subtasks that allows to costruct incremental procedure for good tests generating. The concepts of an essential value and an essential example are also introduced and an incremental learning algorithm MATRIX is described.

The Concept of a Good Classification Test

A good diagnostic test for a given set of examples is defined as follows. Let R be a table of examples and S be the set of indices of examples belonging to R. Let $R(k)$ and $S(k)$ be the set of examples and the set of indices of examples from a given class k, respectively.

Denote by $FM = R/R(k)$ the examples of the classes different from class k. Let U be the set of attributes and T be the set of attributes values (values, for short) each of which appears at least in one of the examples of R. Let n be the number of examples of R. We denote the domain of values for an attribute Atr by $dom(Atr)$, where $Atr \in U$.

By $s(a)$, $a \in T$, we denote the subset $\{i \in S$: a appears in t_i, $t_i \in R\}$, where $S = \{1, 2, .., n\}$. Following (Cosmadakis, et al., 1986), we call $s(a)$ the interpretation of $a \in T$ in R. It is possible to say that $s(a)$ is the set of indices of all the examples in R which are covered by the value a.

Since for all $a, b \in dom(Atr)$, $a \neq b$ implies that the intersection $s(a) \cap s(b)$ is empty, the interpretation of any attribute in R is a partition of S into a family of mutually disjoint blocks. By $P(Atr)$, we denote the partition of S induced by the values of an attribute Atr. The definition of $s(a)$ can be extended to the definition of $s(t)$ for any collection t of values as follows: for t, $t \subseteq T$, if $t = a_1\, a_2 \ldots a_m$, then $s(t) = s(a_1) \cap s(a_2) \cap \ldots \cap s(a_m)$.

Definition 1. A collection $t \subseteq T$ ($s(t) \neq \emptyset$) of values, is a diagnostic test for the set $R(k)$ of examples if and only if the following condition is satisfied: $t \not\subset t^*$, $\forall\, t^*$, $t^* \in FM$ (the equivalent condition is $s(t) \subseteq S(k)$).

To say that a collection t of values is a diagnostic test for the set $R(k)$ is equivalent to say that it does not cover any example belonging to the classes different from k. At the same time, the condition $s(t) \subseteq S(k)$ implies that the following implicative dependency is true: 'if t, then k.

It is clear that the set of all diagnostic tests for a given set $R(k)$ of examples (call it '$DT(k)$') is the set of all the collections t of values for which the condition $s(t) \subseteq S(k)$ is true. For any pair of diagnostic tests t_i, t_j from $DT(k)$, only one of the following relations is true: $s(t_i) \subseteq s(t_j)$, $s(t_i) \supseteq s(t_j)$, $s(t_i) \approx s(t_j)$, where the last relation means that $s(t_i)$ and $s(t_j)$ are incomparable, i.e. $s(t_i) \not\subset s(t_j)$ and $s(t_j) \not\subset s(t_i)$. This consideration leads to the concept of a good diagnostic test.

Definition 2. A collection $t \subseteq T$ ($s(t) \neq \varnothing$) of values is a good test for the set $R(k)$ of examples if and only if the following condition is satisfied: $s(t) \subseteq S(k)$ and simultaneously the condition $s(t) \subset s(t^*) \subseteq S(k)$ is not satisfied for any t^*, $t^* \subseteq T$, such that $t^* \neq t$.

Good diagnostic tests possess the greatest generalization power and give a possibility to obtain the smallest number of implicative rules for describing examples of a given class k.

The Characterization of Classification Tests

Any collection of values can be irredundant, redundant or maximally redundant.

Definition 3. A collection t of values is irredundant if for any value $v \in t$ the following condition is satisfied: $s(t) \subset s(t/v)$.

If a collection t of values is a good test for $R(k)$ and, simultaneously, it is an irredundant collection of values, then any proper subset of t is not a test for $R(k)$.

Definition 4. Let $X \to v$ be an implicative dependency which is satisfied in R between a collection $X \subseteq T$ of values and the value v, $v \in T$. Suppose that a collection $t \subseteq T$ of values contains X. Then the collection t is said to be redundant if it contains also the value v.

If t contains the left and the right sides of some implicative dependency $X \to v$, then the following condition is satisfied: $s(t) = s(t/v)$. In other words, a redundant collection t and the collection t/v of values cover the same set of examples. If a good test for $R(k)$ is a redundant collection of values, then some values can be deleted from it and thus obtain an equivalent good test with a smaller number of values.

Definition 5. A collection $t \subseteq T$ of values is maximally redundant if for any implicative dependency $X \to v$ which is satisfied in R the fact that t contains X implies that t also contains v.

If t is a maximally redundant collection of values, then for any value $v \notin t$, $v \in T$ the following condition is satisfied: $s(t) \supset s(t \cup v)$. In other words, a maximally redundant collection t of values covers the number of examples greater than the collection $(t \cup v)$ of values. If a diagnostic test for a given set $R(k)$ of examples is a good one and it is a maximally redundant collection of values, then by adding to it any value not belonging to it we get a collection of values which is not a good test for $R(k)$.

Any example t in R is a maximally redundant collection of values because for any value $v \notin t$, $v \in T$ $s(t \cup v)$ is equal to $\varnothing$.

For example, in Table 1 the collection '*Blond Bleu*' is a good irredundant test for class 1 and simultaneously it is maximally redundant collection of values. The collection '*Blond Embrown*' is a test for class 2 but it is not good test and simultaneously it is maximally redundant collection of values. The collection '*Embrown*' is a good irredundant test for class 2. The collection '*Red*' is a good irredundant test and the collection '*Tall Red Bleu*' is a maximally redundant and good test for class 1. Any example t in R is a maximally redundant collection of values because for any value $v \notin t$, $v \in T$ $s(t \cup v)$ is equal to $\varnothing$.

Table - 1. Example 1 of Data Classification. (This example is adopted from (Ganascia, 1989)).

Index of example	Height	Color of hair	Color of eyes	Class
1	Low	Blond	Bleu	1
2	Low	Brown	Bleu	2
3	Tall	Brown	Embrown	2
4	Tall	Blond	Embrown	2
5	Tall	Brown	Bleu	2
6	Low	Blond	Embrown	2
7	Tall	Red	Bleu	1
8	Tall	Blond	Bleu	1

An Approach for Constructing Good Irredundant Tests

Let R, T, s(t), t $\subseteq$ T be as defined earlier. We give the following propositions the proof of which can be found in (Naidenova, 1999).

Proposition 1.

The intersection of maximally redundant collections of values is a maximally redundant collection.

Proposition 2.

Every collection of values is contained in one and only one maximally redundant collection with the same interpretation.

Proposition 3.

A good maximal redundant test for R(k) either belongs to the set R(k) or it is equal to the intersection of q examples from R(k) for some q, $2 \le q \le nt$, where nt is the number of examples in R(k).

One of the possible ways for searching for good irredundant tests for a given class of examples is the following: first, find all good maximally redundant tests; second, for each good maximally redundant test, find all good irredundant tests contained in it. This is a convenient strategy as each good irredundant test belongs to one and only one good maximally redundant test with the same interpretation.

It should be more convenient in the following considerations to denote the set $R(k)$ as $R(+)$ (the set of positive examples) and the set $R/R(k)$ as $R(-)$ (the set of negative examples). We will also denote the set $S(k)$ as $s(+)$.

The Duality of Good Diagnostic Tests

In the definition 2, we used correspondences of Galois G on $S \times T$ and two relations $S \to T$, $T \to S$ (Ore, 1944), (Riguet, 1948). Let $s \subseteq S$, $t \subseteq T$. We define the relations as follows:

$$S \to T: t(s) = \{\text{intersection of all } t_i: t_i \subseteq T, i \in s\} \text{ and } T \to S: s(t) = \{i: i \in S, t \subseteq t_i\}.$$

Extending s by an index j^* of some new example leads to receiving a more general feature of examples:

$$(s \cup j^*) \supseteq s \text{ implies } t(s \cup j^*) \subseteq t(s).$$

Extending t by a new value A leads to decreasing the number of examples possessing the general feature 'tA' in comparison with the number of examples possessing the general feature 't':

$$(t \cup A) \supseteq t \text{ implies } s(t \cup A) \subseteq s(t).$$

We introduce the following generalization operations (functions): generalization_of(t) = t' = $t(s(t))$; generalization_of(s) = s' = $s(t(s))$.

As a result of the generalization of s, the sequence of operations $s \rightarrow t(s) \rightarrow s(t(s))$ gives that $s(t(s)) \supseteq s$. This generalization operation gives all the examples possessing the feature $t(s)$. As a result of the generalization of t, the sequence of operations $t \rightarrow s(t) \rightarrow t(s(t))$ gives that $t(s(t)) \supseteq t$. This generalization operation gives the maximal general feature for examples the indices of which are in $s(t)$.

The Definition of Good Diagnostic Tests as dual objects

We implicitly used two generalization operations in all the considerations of diagnostic tests. Now we define a diagnostic test as a dual object, i.e. as a pair (SL, TA), $SL \subseteq S$, $TA \subseteq T$, $SL = s(TA)$ and $TA = t(SL)$.

Definition 6. Let $PM = \{s_1, s_2, \ldots, s_m\}$ be a family of subsets of some set M. Then PM is a Sperner system (Sperner, 1928) if the following condition is satisfied: $s_i \not\subset s_j$ and $s_j \not\subset s_i$, $\forall(i,j)$, $i \neq j$, $i, j = 1, \ldots, m$.

Definition 7. To find all *Good Maximally Redundant Tests* (GMRTs) for a given class $R(k)$ of examples means to construct a family PS of subsets $s_1, s_2, \ldots, s_{np}$ of the set S such that:

1) $s_j \subseteq S(k)$, $\forall j = 1,\ldots, np$;

2) PS is a Sperner system;

3) each s_j is a maximal set in the sense that adding to it the index i of the example t_i such that $i \notin s_j$, $i \in S$ implies $s(t(s_j \cup i)) \not\subset S(k)$. Putting it in another way, $t(s_j \cup i)$ is not a test for the class k, so there exists such example t^*, $t^* \in R(-)$ that $t(s_j \cup i) \subseteq t^*$.

The set of all GMRTs is determined as follows: $\{t: t(s_j), s_j \in PS, \forall j, j = 1,\ldots, np\}$.

Let R be a table of examples and S, T are defined as before. Let MUT be the set of all dual objects, that is, the set of all pairs (s, t), $s \subseteq S$, $t \subseteq T$, $s = s(t)$ and $t = t(s)$. This set is partially ordered by the relation '$\leq$', where $(s, t) \leq (s^*, t^*)$ is satisfied if and only if $s \subseteq s^*$ and $t \supseteq t^*$.

The set Ψ = (MUT, $\cup$, $\cap$) is an algebraic lattice, where operations $\cup$, $\cap$ are defined for all pairs (s^*, t^*), $(s, t) \in$ MUT in the following way (Wille, 1992):

$$(s^*, t^*) \cup (s, t) = ((s^* \cup s), (t^* \cap t)),$$

$$(s^*, t^*) \cap (s, t) = ((s^* \cap s), (t^* \cup t)).$$

The unit element and the zero element are $(S, \varnothing)$ and $(\varnothing, T)$, respectively.

Inferring good tests is reduced to inferring for any element $(s^*, t^*) \in$ MUT all the elements nearest to it in the lattice with respect to the ordering $\leq$, that is, inferring all (s, t), that $(s^*, t^*) \leq (s, t)$ and there does not exist any (s^{**}, t^{**}) such that $(s^*, t^*) \leq (s^{**}, t^{**}) \leq (s, t)$, or inferring all (s, t), that $(s^*, t^*) \geq (s, t)$ and there does not exist any (s^{**}, t^{**}) such that $(s^*, t^*) \geq (s^{**}, t^{**}) \geq (s, t)$. Inferring the chains of lattice elements ordered by the inclusion relation lies in the foundation of generating all types of diagnostic tests:

(1) $s_0 \subseteq \ldots \subseteq s_i \subseteq s_{i+1} \subseteq \ldots \subseteq s_m$ $(t(s_0) \supseteq t(s_1) \supseteq \ldots \supseteq t(s_i) \supseteq t(s_{i+1}) \supseteq \ldots \supseteq t(s_m))$,

(2) $t_0 \subseteq \ldots \subseteq t_i \subseteq t_{i+1} \subseteq \ldots \subseteq t_m$ $(s(t_0) \supseteq s(t_1) \supseteq \ldots \supseteq s(t_i) \supseteq s(t_{i+1}) \supseteq \ldots \supseteq s(t_m))$.

We will use only the chain (1) for inferring good diagnostic tests.

Decomposition of Good Classification Tests Inferring into Subtasks

Now we consider some decompositions of the problem that provide the possibility to restrict the domain of searching, to predict, in some degree, the number of tests, and to choose tests with the use of essential values and/or examples. We consider three kinds of subtasks: for a given set of positive examples

1) given a positive example t, find all GMRTs contained in t;

2) given a non-empty collection of values X (maybe only one value A) such that it is not a test, find all GMRTs containing X;

3) given a non-empty collection of values X (maybe only one value A) such that it is not a test and a positive example t, $X \subseteq t$, find all GMRTs containing X and simultaneously contained in t.

Forming the Subtasks

The subtask of the first kind. We introduce the concept of an example's projection proj(R)[t] of a given positive example t on a given set R(+) of positive examples. The proj(R)[t] is the set Z = {z: (z is non empty intersection of t and t') & ($t' \in R$(+)) & (z is a test for R(+))}.

If the proj(R)[t] is not empty and contains more than one element, then it is a subtask for inferring all GMRTs that are in t. If the projection contains one and only one element equal to t, then t is a GMRT.

The subtask of the second kind. We introduce the concept of an attributive projection proj(R)[A] of a given value A on a given set R(+) of positive examples.

The projection proj(R)[A] = {t: ($t \in R$(+)) & (A appears in t)}. Another way to define this projection is: proj(R)[A] = {t_i: $i \in$ ($s(A) \cap s(+)$)}. If the attributive projection is not empty and contains more than one element, then it is a subtask of inferring all GMRTs containing a given value A. If A appears in one and only one example, then A does not belong to any GMRT different from this example.

Forming the projection of A makes sense if A is not a test and the intersection of all positive examples in which A appears is not a test too, i.e. $s(A) \not\subset s(+)$ and $t' = t(s(A) \cap s(+))$ is not a test for a given set of positive examples.

Denote the set {$s(A) \cap s(+)$} *by* splus(A). Generally, we can consider the projection proj(R)[X], $X \subseteq T$.

The subtask of the third kind. Now we introduce proj(R)[t x A] = proj(R)[A x t], where $A \subseteq t$. In order to construct this projection the folowing steps are implemented: 1) to select all examples $t_i \in R$(+) containing A; 2) In each selected example, it is necessary to take only the values which appear in t. The result is the following:

Z = {z: $z = t \cap t_i, \neq \varnothing$, $t_i \in R(+)$, $A \subseteq t_i$, $A \subseteq t$, (z is a test for R(+))}.Generally, we can consider the projection proj(R)[t x X] = proj(R)[X x t], where $X \subseteq t$.

To make the operation of forming a projection perfectly clear we construct the projection proj(R)[t_2 x *Brown*] on the examples of the second class where t_2 = '*Low Brown Bleu*' (Table 1). This projection includes t_2 and the intersections of t_2 with the other positive examples of Class 2, containing the value '*Brown*', i.e. with the examples t_3, t_5 (Table 3).

In order to check whether an element of the projection is a test or not we use the function to_be_test(t) in the following form: to_be_test(t) = if $s(t) \subseteq s(+)$ then *true* else *false*, where s(+) is the set of indices of positive examples, $s(t)$ is the set of indices of all positive and negative examples containing t. If s(-) is the set of indices of negative examples, then $S = s(+) \cup s(-)$ and $s(t) = \{i: t \subseteq t_i, i \in S\}$.

The subtask turns out to be very simple because the intersection of all the rows of the projection is a test for the second class: t({2,3,5}) = '*Brown*', s(*Brown*) = {2,3,5} and {2,3,5} $\subseteq s(+)$.

Table - 3. The Intersections of Example t_2 with the Examples of Class 2. (t_2 x *Brown*)

Index of example	Height	Color of hair	Color of eyes	test?
2	Low	Brown	Bleu	Yes
3		Brown		Yes
5		Brown	Bleu	Yes

Reducing the Subtasks

The following theorem gives the foundation for reducing projections of any kind. The proof of this theorem can be found in (Naidenova et al., 1995).

THEOREM 1.

Let *A* be a value from *T*, *X* be a maximally redundant test for a given set *R*(+) of positive examples and $s(A) \subseteq s(X)$. Then *A* does not belong to any maximally redundant good test for *R*(+) different from *X*.

Deleting values from a projection can imply deleting rows stopping to be tests. Deleting rows from a projection can imply deleting values satisfying the condition of the Theorem 1. To illustrate the way of reducing projections, we consider another partition of the rows of Table 1 into the sets of positive and negative examples as shown in Table 4.

Table - 4. The Example 2 of a Data Classification.

Index of example	Height	Color of hair	Color of eyes	Class
1	Low	Blond	Bleu	1
2	Low	Brown	Bleu	1
3	Tall	Brown	Embrown	1
4	Tall	Blond	Embrown	2
5	Tall	Brown	Bleu	2
6	Low	Blond	Embrown	2
7	Tall	Red	Bleu	2
8	Tall	Blond	Bleu	2

Let s(+) be equal to {4,5,6,7,8}. The value 'Red' is a test for positive examples because s(Red) = splus(Red) = {7}. Delete 'Red' from the projection. The value 'Bleu' is not a test because s(Bleu) = {1,2,5,7,8}. But splus(Bleu) = {5,7,8} and t(splus(Bleu)) = 'Tall Bleu' is a test for Class 2. Delete 'Bleu' from examples of Class 2 as shown in Table 5. Now the rows t_5 and t_7 are not tests for Class 2 and they can be deleted.

Table - 5. The Example of a projection reduced.

Index of example	Height	Color of hair	Color of eyes	Class	Test?
1	Low	Blond	Bleu	1	Yes
2	Low	Brown	Bleu	1	Yes
3	Tall	Brown	Embrown	1	Yes
4	Tall	Blond	Embrown	2	Yes
5	Tall	Brown		2	No
6	Low	Blond	Embrown	2	Yes

7	Tall		2	No
8	Tall	Blond	2	Yes

Choosing Values and Examples for the Formation of Subtasks

Definition 8. Let *t* be a collection of values that is a test for a given set of positive examples. We say that the value *A* in *t* is essential if (t/A) is not a test for a given set of positive examples.

Generally, we are interested in finding the maximal subset $sbmax(t) \subset t$ such that *t* is a test but $sbmax(t)$ is not a test for a given set of positive examples. Then $sbmin(t) = t/\ sbmax(t)$ is the minimal set of essential values in *t*.

Definition 9. Let *s* be a subset of indices of positive examples; assume also that $t(s)$ is not a test. The example t_j, $j \in s$ is to be said an essential one if $t(s/j)$ proves to be a test for a given set of positive examples.

Generally, we are interested in finding the maximal subset $sbmax(s) \subset s$ such that $t(s)$ is not a test but $t' = t(sbmax(s))$ is a test for a given set of positive examples. Then $sbmin(s) = s/sbmax(s)$ is the minimal set of indices of essential examples in *s*.

In order to construct the projection proj(R)[t_i x A] = proj(R)[A x t_i] for a subtask of the third kind it is very convenient to take t_i and A such that $i \in sbmin(splus(A))$ and $A \in sbmin(t_i)$.

An Approach for Searching for Essential Values

Let t be a test for positive examples. Construct the set of intersections $\{t \cap t': t' \in R(-)\}$. It is clear that these intersections are not tests for positive examples. Take one of the intersections with the maximal number of values in it. The values complementing the maximal intersection in t is one of the minimal sets of essential values in t.

Return to Table 6. Exclude the value 'Red' (we know that 'Red' is a test for Class 2) and find the minimal subsets of essential values for t_4, t_5, t_6, t_7, and t_8. The result is in Table 6.

Table - 6. The Essential Values for the Examples *t_4, t_5, t_6, t_7, and t_8.*

Index of example	Height	Color of hair	Color of eyes	Subsets of essential values	Class
1	Low	Blond	Bleu		1
2	Low	Brown	Bleu		1
3	Tall	Brown	Embrown		1
4	Tall	Blond	Embrown	{Blond}	2
5	Tall	Brown	Bleu	{Bleu}, {Tall}	2
6	Low	Blond	Embrown	{Embrown}	2
7	Tall		Bleu	{Tall}, {Bleu}	2
8	Tall	Blond	Bleu	{Tall}	2

An Approach for Searching for Essential Examples

Let *STGOOD* be the partially ordered set of elements *s* satisfying the condition that $t(s)$ is a good maximally redundant test (GMRT) for $R(+)$. We can use the set *STGOOD* to find indices of essential examples in some subset s^* of indices for which $t(s^*)$ is not a test. Let $s^* = \{i_1, i_2, \ldots, i_q\}$. Construct the set of intersections $\{s^* \cap s': s' \in STGOOD\}$. Any obtained intersection $s^* \cap s'$ corresponds to a test for positive examples. Take one of the intersections with the maximal number of indices. The subset of s^* complementing in s^* the maximal intersection is one of the minimal sets of indices of essential examples in s^*. For instance, $s^* = \{2,3,4,7,8\}$, $s' = \{2,3,4,7\}$, $s' \in STGOOD$, hence 8 is the index of essential example t_8 in s^*.

In the beginning of inferring GMRTs, the set *STGOOD* is empty. The procedure with the use of which a quasi-maximal subset of s^* that corresponds to a test is obtained has been described in (Naidenova, 2005).

MATRIX – an Algorithm for Incremental Inferring Good Maximally Redundant Diagnostic Tests

Incremental learning is necessary when a new portion of training examples becomes available over time. Suppose that each new example comes with the indication of its class membership. The following actions are necessary with the arrival of a new example:

- Check whether it is possible to perform generalization of some existing GMRTs for the class to which the new example belongs (class of positive examples), i.e., whether it is possible to extend the set of examples covered by some existing GMRTs or not.

- Infer all the GMRTs contained in the new example.

- Check the validity of the existing GMRTs for negative examples, and if it necessary:

Modify tests that are not valid (test for negative examples is not valid if it is included in a positive example, i.e., in other words, it accepts an example of positive class).

Thus the process of inferring all the GMRTs is divided into the subtasks that conform to three acts of reasoning:

- Pattern recognition or using already known rules (tests) for determining the class membership of a new positive example and generalization of these rules (deductive reasoning and increasing the inductive base of already existing knowledge). This act is performed by the procedure GENERALIZATION (STGOOD, j*) (Figure 1).

- Inferring new rules (tests) that are included in a new positive example. This act can be reduced to the subtask of the first kind or to the subtask(s) of the third kind.

The procedure
GENERALIZATION ($STGOOD(+)$, j^*).

Input: j^*, the set $STGOOD(+)$ of known GMRTs for the class of positive examples, the set $R(-)$ of negative examples.
Output: $STGOOD(+)$ modified by the generalization.

Begin
$(\forall s)$ $(s \in STGOOD(+))$
  if to_be_test($t(s \cup j^*)$) = true **then**
  $s \leftarrow$ generalization $(s \cup j^*)$;
end

Figure 1. The Procedure for Generalizing the Existing GMRTs.

The procedure MATRIX.

Input : j^*, R, S, $STGOOD$. **Output:** R, S, $STGOOD$.
begin
$k \leftarrow$ class(j^*); $S(+) \leftarrow S(k)$; $R(+) \leftarrow R(k)$; $R(-) \leftarrow R/R(+)$;
$N \leftarrow N + 1$; $j^* \leftarrow N$, where N is the number of examples;
$S(+) \leftarrow j^* \cup S(+)$; $R(+) \leftarrow t_{j^*} \cup R(+)$;
$STGOOD(+) \leftarrow STGOOD(k)$;
$STGOOD(-) \leftarrow \cup\, STGOOD(kl), \forall kl, kl \neq k$;
if $N = 1$ **then** $STGOOD(+) \leftarrow \{j^*\} \cup STGOOD(+)$; **else**
if $N \neq 1$ and $\|S(+)\| = 1$ **then**

begin
$STGOOD(+) \leftarrow \{j^*\} \cup STGOOD(+)$;
if $(\exists s), s \in STGOOD(-), t(s) \subseteq t_{j^*}$
then CORRECT($t(s)$); **end**
else
if $N \neq 1$ and $S(-) = \varnothing$ **then**

CONCEPTGENERALIZATION [*j]**($S(+)$, $STGOOD(+)$);
else /* $N \neq 1$ and $\|S(+)\| \neq 1$ and $S(-) \neq \varnothing$ */
begin
if $STGOOD(+) \neq \varnothing$ **then**
GENERALIZATION($STGOOD(+)$, j^*); **end**
FORMSUBTASK (j^*);
DIAGaRa [*j]** (S(test)(+), R, S, $STGOOD$ (+));
if $(\exists\, s), s \in STGOOD(-), t(s) \subseteq t_{j^*}$
then CORRECT ($t(s)$);
end

Figure - 2. The Incremental Procedure MATRIX

- Correcting rules (tests) of alternative (negative) classes that accept a new positive example (deductive and inductive diagnostic reasoning to modify knowledge). This act can be reduced to the subtask of the second kind or the subtask(s) of the third kind.

All these subtasks can be solved by DIAGaRa, the basic procedure for inferring GMRTs (Naidenova, 2005).

We must consider four possible situations that can take place when a new example comes to the learning system: 1) the data base is empty; 2) the data base contains only examples of the class to which a new example belongs; 3) The data base contains only examples of the negative class with respect to a new example; 4) the data base contains examples both of the positive and the negative classes. Case 2 conforms to the generalization process taking into account only the similarity relation between examples of the same class. This problem is known in the literature as inductive inference of generalization hypotheses or unsupervised generalization. An algorithm for solving this problem in the framework of a mathematical model based on Galois's connections can be found in (Kuznetzov, 1993). Let CONCEPTGENERALIZATION [j*](S(+), STGOOD(+)) be the procedure of generalization of positive examples in the absence of negative examples.

The algorithm MATRIX for inferring GMRTs is presented in Figure 2. CORRECT (*t*) is the procedure of modifing test *t*, FORMSUBTASK(*j*) is the procedure of forming subtasks.

Appendix: An Example of Using the Algorithm MATRIX

The data in Table 7 are intended for processing by the incremental learning procedure MATRIX. This table is adopted from (Quinlan and Rivest, 1989). The sets STGOOD(1) and STGOOD(2) in Tables 8 accumulate the collections of indices that correspond to the GMRTs for the examples of Class 1 and Class 2, respectively, at each step of the algorithm. Only one new example is added at each step of the procedure.

Table - 7. The Data for Processing by the Incremental Procedure MATRIX

Index of example	Outlook	Temperature	Humidity	WindY	Class
1	Sunny	Hot	High	No	1
2	Sunny	Hot	High	Yes	1
3	Overcast	Hot	High	No	2
4	Rain	Mild	High	No	2
5	Rain	Cool	Normal	No	2
6	Rain	Cool	Normal	Yes	1
7	Overcast	Cool	Normal	Yes	2
8	Sunny	Mild	High	No	1
9	Sunny	Cool	Normal	No	2
10	Rain	Mild	Normal	No	2
11	Sunny	Mild	Normal	Yes	2
12	Overcast	Mild	High	Yes	2
13	Overcast	Hot	Normal	No	2
14	Rain	Mild	High	Yes	1

Table – 8. The Records of Step-by-Step Results of the Incremental Procedure MATRIX.

j*	class(J*)	STGOOD(1), STGOOD(2)	j*	class(J*)	STGOOD(1), STGOOD(2)
{1}	1	STGOOD(1): {1}	{8}	1	STGOOD(1):{1,2,8}, {6}
{2}	1	STGOOD(1):{1,2}	{9}	2	STGOOD(2):{3,7},{4,5},{5,9}
{3}	2	STGOOD(2):{3}	{10}	2	STGOOD(2):{3,7},{4,5,10},{5,9,10}
{4}	2	STGOOD(2):{3}, {4}	{11}	2	STGOOD(2):{3,7},{4,5,10}, {5,9,10},{10,11},{9,11}
{5},	2	STGOOD(2):{3}, {4,5}	{12}	2	STGOOD(2): {3,7,12},{4,5,10}, {5,9,10},{10,11},{9,11},{11,12}
{6}	1	STGOOD(1):{1,2}, {2, 6}	{13}	2	STGOOD(2): {3,7,12,13},{4,5,10}, {5,9,10,13},{10,11},{9,11},{11,12}
{7}	2	STGOOD(2):{3,7},{4,5} STGOOD(1):{1,2}, { 6}	{14}	1	STGOOD(1): {1,2,8}, {6,14} STGOOD(2): {3,7,12,13},{4,5,10}, {5,9,10,13},{10,11},{9,11}

Table 9 contains all the GMRTs obtained for the examples of Class 1 and Class 2: TGOOD(*j*) = {*t*(*s*): $s \in$ STGOOD(*j*)}.

This application of the algorithm MATRIX did not require any call for the procedure DIAGARA. Only one address was necessary to the procedure CONCEPTGENERALIZATION[*j]* in the beginnig of inferring GMRTs. Only two addresses were necessary to the procedure CORRECT(*t*(*s*)) after the arrival of Example 7 and Example 14. All the subtasks were very simple and allowed reading the tests directly without calling for the procedure DIAGaRa.

Table - 9. The Sets TGOOD(1) and TGOOD(2) Produced by the Procedure MATRIX

TGOOD(1)	TGOOD(2)
Sunny High	Rain No
Rain Yes	Normal No
-	Mild Normal
-	Sunny Normal
-	Overcast

Bibliography

[Cosmadakis et al., 1986] S. Cosmadakis, P. C. Kanellakis, N. Spyratos, "Partition Semantics for Relations", *Journal of Computer and System Sciences*, Vol. 33, No. 2, pp.203-233, 1986. [Demetrovics and Vu, 1993] J. Demetrovics and D. T. Vu, "Generating Armstrong Relation Schemes and Inferring Functional Dependencies from Relations", *International Journal on Information Theory & Applications*, Vol. 1, No. 4, pp.3-12, 1993.

[Ganascia, 1989] J.- Gabriel. Ganascia, "EKAW - 89 Tutorial Notes: Machine Learning", *Third European Workshop on Knowledge Acquisition for Knowledge-Based Systems*, Paris, France, pp. 287-296, 1989.

[Kuznetsov, 1993] S. O. Kuznetsov, "Fast Algorithm of Constructing All the Intersections of Finite Semi-Lattice Objects", *Proceedings of VINITI*, Series 2, No. 1, pp. 17-20, 1993.

[Naidenova, 1996] X. A. Naidenova, "Reducing Machine Learning Tasks to the Approximation of a Given Classification on a Given Set of Examples", *Proceedings of the 5-th National Conference at Artificial Intelligence*, Kazan, Tatarstan, Vol. 1, pp. 275-279, 1996.

[Naidenova, 1999] X. A. Naidenova, "The Data-Knowledge Transformation", in: "*Text Procesing and Cognitive Technologies", Paper Collection*, editor Solovyev, V. D., - Pushchino, Russia, Vol. 3, pp. 130-151, 1999.

[Naidenova, 2005] X. A. Naidenova, "DIAGARA : An Incremental Algorithm for Inferring Implicative Rules from Examples", International Journal 'Information Theories & Applications', Vol. 12, pp. 171-186.

[Naidenova et al., 1995] X. A. Naidenova, M. V. Plaksin, V. L. Shagalov, "Inductive Inferring All Good Classification Tests", *Proceedings of International Conference "Knowledge-Dialog-Solution*", Jalta, Ukraine, Vol. 1, pp.79-84, 1995b.

[Naidenova and Polegaeva, 1986] X. A. Naidenova, J. G. Polegaeva, "An Algorithm of Finding the Best Diagnostic Tests", *The 4-th All Union Conference "Application of Mathematical Logic Methods"*, Theses of Papers, Mintz, G; E, Lorents, P. P. (Eds), Institute of Cybernetics, National Acad. of Sciences of Estonia, Tallinn, Estonia, pp. 63-67, 1986.

[Ore, 1944] O. Ore, "*Galois Connexions*", Trans. Amer. Math. Society, Vol. 55, No. 1, pp. 493-513, 1944.

[Quinlan and Rivest,1989] J. R. Quinlan, and R. L. Rivest, "Inferring Decision Trees Using the Minimum Description Length Principle", Information and Computation, Vol. 80, No. 3, pp. 227-248, 1989.

[Riguet, 1948] J. Riguet, "Relations Binaires, Fermetures, Correspondences de Galois", *Bull. Soc. Math*., France, Vol. 76., No 3, pp.114-155, 1948.

[Sperner, 1928] E. Sperner, "Eine satz uber Untermengen einer Endlichen Menge". *Mat. Z.*, Vol. 27, No. 11, pp. 544-548, 1928.

[Wille, 1992] R. Wille, "Concept Lattices and Conceptual Knowledge System", *Computer Math. Appl.*, Vol. 23, No. 6-9, pp. 493-515, 1992.

Author's Information

Naidenova Xenia Alexandrovna - Military medical academy, Saint-Petersburg, Stoikosty street, 26-1-248, e-mail: naidenovaxen@gmail.com