An Approach to Incremental Learning Good Classification Tests

[1]Xenia Naidenova, [2]Vladimir Parkhomenko

[1]Military Medical Academy, Saint-Petersburg, Russian Federation

ksennaid@gmail.com

[2]Saint-Petersburg State Polytechnic University, Russian Federation

parhomenko.v@gmail.com

Abstract. An algorithm of incremental mining implicative logical rules is proposed. This algorithm is based on constructing good classification tests. The incremental approach to constructing these rules allows revealing the interdependence between two fundamental components of human thinking: pattern recognition and knowledge acquisition.

Keywords: Incremental learning, Good classification test, Pattern recognition, Machine learning, Human mental operations

1 Introduction

Methods of incremental symbolic machine learning are developing in several directions. The first one is to construct incrementally concept lattice [1-3]. In [3], an incremental algorithm to construct a lattice from a collection of sets is derived, refined, analyzed, and related to a similar previously published algorithm for constructing concept lattices. The second direction is related to incremental mining association rules [4-5]. This direction includes the incremental approach to mining frequent itemsets based on Galois Lattice Theory [6]. Significantly fewer investigations are devoted to incremental mining logical rules. Utgoff proposed three incremental decision tree induction algorithms [7]. Rough set based incremental method is advanced in [8].

This paper is devoted to incremental learning of logical rules in the form of implications based on the concept of good classification test. Incremental learning is considered not only as a model of inductive human reasoning for implicative logical rule generation but also as an essential part of pattern recognition processes.

2 The Concept of Good Classification Test

Let $G = \{1, 2,\ldots, N\}$ be the set of objects' indices (objects, for short) and $M = \{m_1, m_2, \ldots, m_j, \ldots m_m\}$ be the set of attributes' values (values, for short). Each object is described by a set of values from M. The object descriptions are represented by rows of a table R the columns of which are associated with the attributes taking their values in M. Let $R(+)$ and $G(+)$ be the sets of positive object descriptions and the set of indices of these objects, respectively. Then $R(-) = R/R(+)$ and $G(-) = G/G(+)$ are the set of negative object descriptions and the set of indices of these objects, respectively.

The definition of good tests is based on two mapping $2^G \to 2^M$, $2^M \to 2^G$ determined as follows. $A \subseteq G$, $B \subseteq M$. Denote by B_i, $B_i \subseteq M$, $i = 1,\ldots, N$ the description of object with index i. We define the relations $2^S \to 2^T$, $2^T \to 2^S$ as follows: $A' = \mathrm{val}(A) = \{$intersection of all B_i: $B_i \subseteq M$, $i \in G\}$ and $B' = \mathrm{obj}(B) = \{i: i \in G, B \subseteq B_i\}$. These mapping are Galois's correspondences [9]. Of course, we have $\mathrm{obj}(B) = \{$intersection of all $obj(m)$: $obj(m) \subseteq G$, $m \in B\}$. Operations val(A), obj(B) are reasoning operations (derivation operations).

The generalization operations generalization_of(B) = $B'' = \mathrm{val}(\mathrm{obj}(B))$ and generalization_of(A) = $A'' = \mathrm{obj}(\mathrm{val}(A))$ are are actually closure operators [9]. A set A is closed if $A = \mathrm{obj}(\mathrm{val}(A))$. A set B is closed if $B = \mathrm{val}(\mathrm{obj}(B))$.

Notice that these generalization operations are also used in FCA [10], [11]. For $g \in G$ and $m \in M$, $\{g\}'$ is denoted by g' and called **object intent**, and $\{m\}'$ is denoted by m' and called **value extent**.

Definition 1. **A diagnostic (classification) test** for $R(+)$ is a pair (A, B) such that $B \subseteq M$ ($A = \mathrm{obj}(B) \neq \varnothing$), $A \subseteq G(+)$ and $B \not\subset \mathrm{val}(g)$ & $B \neq \mathrm{val}(g)$, $\forall g$, $g \in G(-)$. Equivalently, $\mathrm{obj}(B) \cap G(-) = \varnothing$.

In general case, a set B is not closed for diagnostic test (A, B), consequently, diagnostic test is not obligatory a concept of FCA [12].

To say that a collection B of values is a diagnostic test for the set $R(k)$ is equivalent to say that it does not cover any object description belonging to the classes different from k. At the same time, the condition $\mathrm{obj}(B) \subseteq G(k)$ implies that the following implicative dependency is true: 'if B, then k' and, consequently, a diagnostic test, as a set of values, makes up the left side of an implication.

It is clear that the set of all diagnostic tests for a given set $R(k)$ (call it '$DT(k)$') is the set of all B such that the condition $\mathrm{obj}(B) \subseteq G(k)$ is true. For any pair of diagnostic tests from $DT(k)$ only one of the following relations is true:$\mathrm{obj}(B_i) \subseteq \mathrm{obj}(B_j)$, $\mathrm{obj}(B_i) \supseteq \mathrm{obj}(B_j)$, $\mathrm{obj}(B_i) \approx \mathrm{obj}(B_j)$, where the last relation means that $\mathrm{obj}(B_i)$ and $\mathrm{obj}(B_j)$ are incomparable, i.e. $\mathrm{obj}(B_i) \not\subset \mathrm{obj}(B_j)$ and $\mathrm{obj}(B_j) \not\subset \mathrm{obj}(B_i)$. This consideration leads to the concept of a good diagnostic test: they are maximal elements of partially ordered set $DT(k)$.

Definition 2. A classification test (A, B), $B \subseteq M$ ($A = \mathrm{obj}(B) \neq \varnothing$) is **good** for $R(+)$ if and only if any extension $A^* = A \cup i$, $i \notin A$, $i \in G(+)$ implies that $(A^*, \mathrm{val}(A^*))$ is not a test for $R(+)$.

Definition 3. A good diagnostic test (A, B), $B \subseteq M$ (A = obj(B) $\neq \varnothing$) for $R(+)$ is **irredundant** if any narrowing $B^* = B \backslash m$, $m \in B$ implies that (obj(B^*), B^*)) is not a test for $R(+)$.

Definition 4. A good diagnostic test for $R(+)$ is **maximally redundant** if any extension of $B^* = B \cup m$, $m \notin B$, $m \in M$ implies that (obj(B^*, B^*) is not a good test for $R(+)$.

It is possible to show that good maximally redundant tests (GMRTs) are closed, consequently, they are formal concepts in term of the FCA, but they are not always frequent itemsets [12]. In what follows, we shall consider mining GMRTs.

The first algorithm for inferring all GMRTs for a given class of objects with its theoretical foundation has been proposed in [13] and analyzed in [14]. Then an algorithm ASTRA has been proposed and realized in a program system SIZIF [15]. The algorithms NIAGaRa and DIAGaRa have been described in [16] and [17], respectively. Diagnostic Test Machine (DTM) [18] is a software based on supervised mining good diagnostic tests. The experiments conducted with the publicly available dataset of 8124 mushrooms have showed that the result of the DTM turned out to be better with respect to classification accuracy (97,5%) than the results (95%) informed in [19] for the same set of data.

Any algorithm for mining GMRTs can be used as a part of incremental algorithm solving the same task.

The Decomposition of Good Test Inferring into Subtasks

To transform good classification test inferring into an incremental process, we introduce two kinds of subtasks [15], [16]: for a given set of positive examples: 1) given a set of values $B \subseteq M$, (obj(B), B) is a test, find all $B^* \subset B$ such that (obj(B^*), B^*) is a GMRT; 2) given a non-empty set of values $X \subseteq M$ such that (obj(X), X) is not a test, find all Y, $X \subset Y$, such that (obj(Y), Y) is a GMRT.

The subtask of the first kind. We introduce a concept of projection proj(R)[t] of a given positive object description 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') & ($t' \in R(+)$) & ((obj(z), z) is a test for $R(+)$)}.

If 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 t, then (obj(t), t) is a GMRT.

The subtask of the second kind. We introduce a concept of attributive projection proj(R)[B] of a given set B of values on a given set $R(+)$ of positive examples. The projection proj(R)[B] = {t: ($t \in R(+)$) & (B appears in t)}. Another way to define this projection is: proj(R)[B] = {t_i: $i \in$ (obj(B) $\cap$ $G(+)$)}. If attributive projection is not empty and contains more than one element, then it is a subtask for inferring all GMRTs containing B. If B appears in one and only one object description t, then there is only one GMRT: (obj(t), t).

The following theorem gives the foundation of reducing projections [15], [16].

Theorem 1. Let $m \in M$, (obj(X), X) be a maximally redundant test for a given set R(+) of positive objects and obj(m) $\subseteq$ obj(X). Then m does not belong to any GMRT for R(+) different from (obj(X), X).

1 An Approach to Incremental Inferring GMRTs

Incremental supervised learning is necessary when a new portion of observations becomes available over time. Suppose that each new object comes with the indication of its class membership. The following actions are necessary with arrival of a new object: 1) checking whether it is possible to perform generalization of some existing rules (tests) for the class to which a new object belongs (a class of positive objects, for certainty); 2) inferring all GMRTs induced by the new object description; 3) checking the validity of rules (tests) for negative objects, and, if it is necessary, modifying the tests that are invalid (test for negative objects is invalid if its intent is included in a new (positive) object description). Thus the following mental acts are performed:

- Pattern recognition and generalization of knowledge (increasing the power of already existing inductive knowledge);
- Increasing knowledge (inferring new knowledge);
- Correcting knowledge (diagnostic reasoning).

The first act modifies already existing tests (rules). The second act is reduced to subtask of the first kind. The third act can be implemented by the following ways. In the first way, we delete invalid tests (rules) and, by the use of subtask of the first kind, we must find new GMRTs generated by negative objects's descriptions that have been covered by invalid tests. In the second way, this act can be reduced to subtasks of the second kind. Then we obtain diagnostic logical assertions in the form: X, $d \rightarrow$ negative class of objects; X, $b \rightarrow$ positive class of objects; d, $b \rightarrow$ false, where X, d, b $\subset$ M, and X is object intent of invalid test .

Algorithm DIAGaRa is used for solving both kinds of subtasks. Currently, we realize the first way with deleting invalid tests.

2 DIAGaRa: an Algorithm for Inferring GMRTs

The decomposition of inferring GMRTs into subtasks of first and second kinds gives the possibility to construct incremental algorithms. The simplest way to do it consists of the following steps: choose object description (value), form subtask, solve subtask, delete object description (value) after the subtask is over, and check the condition of ending the main task. In this process, already obtained tests are used for pruning the search space.

DIAGaRa is based on using a basic recursive procedure for solving subtask of the first kind The initial information for finding all the GMRTs contained in a positive example (object) description is the projection of this example on the current set R(+). It is essential that the projection is simply a subset of examples (object descriptions) defined on a certain restricted subset B^* of values. Let A^* be the subset of indices of objects from $R(+)$ which have produced the projection.

Generally, it is useful to introduce the weight W(B) of any set B of values in the projection: $W(B) = \|splus(B)\| = \|obj(B) \cap A^*\|$ is the number of positive object descriptions of the projection containing B. Let WMIN be the minimal permissible value of weight. Currently, we assume that WMIN = 1.

Let STGOOD be the partially ordered set of elements $A \subseteq A^*$ satisfying the condition that (A, val(A)) is a good test for $R(+)$. The basic recursive procedure consists of applying the sequence of the following steps:

Step 1. Check whether the intersection of all the elements of projection corresponds to a test and if so, then A^* is stored in STGOOD if (A^*, val(A^*)) is currently a good test; in this case, the subtask is over. Otherwise the next step is performed.

Step 2. The generalization operation is performed as follows: $B' = \text{val}(splus(m))$, $m \in B^*$; if B' is object intent of a test, then m is deleted from the projection and *splus*(m) is stored in *STGOOD* if *splus*(m) is currently value extent of a good test.

Step 3. The value m is deleted from the projection if $splus(m) \subseteq s$ for some $s \in$ *STGOOD*.

Step 4. If at least one value has been deleted from the projection, then the reduction of the projection is necessary. The reduction consists in deleting the elements of projection that do not correspond to tests (as a result of previous eliminating values). If, under reduction, at least one element has been deleted from the projection, then Step 2, Step 3, and Step 4 are repeated.

Step 5. Check whether the subtask is over or not. The subtask is over when either the projection is empty or the intersection of all elements of the projection corresponds to a test (see, please, Step 1). If the subtask is not over, then an element of this projection is selected, new subtask is formed, and the basic algorithm runs recursively.

An Approach for Forming the Set *STGOOD*. The important part of the basic algorithm is how to form the set *STGOOD*. Let $L(S)$ be the set of all subsets of the set S. $L(S)$ is the set lattice. The ordering determined in the set lattice coincides with the set-theoretical inclusion. It will be said that subset s_1 is absorbed by subset s_2, that is $s_1 \leq s_2$, if and only if the inclusion relation is hold between them, that is $s_1 \subseteq s_2$. Under formation of *STGOOD*, a set s is stored in *STGOOD* if and only if it is not absorbed by any element of this set. It is necessary also to delete from *STGOOD* all the elements in it that are absorbed by s. Thus, when the algorithm is over, *STGOOD* contains all the collections of objects that correspond to GMRTs and only such collections. Essentially the process of forming *STGOOD* is an incremental procedure of finding all maximal elements of a partially ordered set.

The set *TGOOD* of all the GMRTs is obtained as follows: $TGOOD = \{(s, \text{val}(s)), (\forall s)\ (s \in STGOOD)\}$.

3 INGOT: An Incremental Algorithm for Inferring All GMRTs

The first act is performed by the procedure GENERALIZATION (STGOOD, j^*).

```
The procedure GENERALIZATION (STGOOD(+), j*).
Input: j* is the index of new example (object), the set
STGOOD(+) of GMRTs for the class of positive examples,
the set R(-) of negative examples.
Output: STGOOD(+)modified by the generalization.
Begin
(∀s) (s ∈ STGOOD(+))
  if to_be_test({s ∪ j*}, val({s ∪ j*})) = true then
  s ← generalization (s ∪ j*);
end
```

The second act is reduced to the subtask of the first kind. The procedure FORMSUBTASK(*j*) aims at preparing initial data for inferring all the GMRTs contained in description *t* of object with index *j*:

```
The procedure FORMSUBTASK(j, R(class(j)), G(class(j)),
STGOOD(class(j)).
Input: j, R((class (j)), R(−) = R/R(class(j)),
G(class(j)), STGOOD(class(j)). Output:
proj(R(class(j))[j];
              Begin
proj(R(class(j))[j]   ← {{j}}; nts   ← G(class(j);
(∀i) i ∈ nts, i ≠ j
   if to_be_test(({j, i}, val({j, i})) = true then do
     Begin
     insert i into proj(R(class(j))[j];
     end
end
```

Four possible situations can take place when a new object comes to the learning system:

- The knowledge base is empty;
- The knowledge base contains only objects of the positive class to which a new object belongs;
- The knowledge base contains only objects of the negative class;
- The knowledge base contains objects of both the positive and the negative classes.

The second situation 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 can be found in [20].

Let CONCEPTGENERALIZATION [j*](*G*(+), *STGOOD*(+)) be the procedure of generalization of positive examples in the absence of negative examples. Next, the procedure INGOT is presented.

```
The procedure INGOT(j*).
Input: j*, class(j*), t* - description of j*-object, R,
G, STGOOD = STGOOD(+) ∪ STGOOD(−). Output: STGOOD.
begin
k ← class(j*); G(+)   ←  G(k); R(+)   ← R(k); R(-)   ←
R/R(+), G(-)   ← G/G(+);
N   ← N + 1;  j*   ← N, where N is the number of ob-
jects;
G(+)   ← j* ∪ G(+);   R(+)   ← t* ∪ R(+);
STGOOD(+)   ←  STGOOD(k);
STGOOD(-)   ← ∪ STGOOD/STGOOD(+);
if  N  = 1 then  STGOOD(+) ← {j*} ∪ STGOOD(+);  else
if  N  ≠ 1 and ‖G(+)‖ = 1 then
begin
STGOOD(+)   ←  {j*} ∪ STGOOD(+);
(∀s), s ∈ STGOOD(-), val(s) ⊆ t*
   (∀j), j ∈ G(−), s ⊆ val(j)
    FORMSUBTASK (j, R(−), G(−), STGOOD(−);
    DIAGaRa(proj(R(−)[j], STGOOD(−));
    end
end
else if  N ≠ 1 and G(-) = ∅ then
CONCEPTGENERALIZATION [j*](G(+), STGOOD(+));
else   /* N  ≠ 1 and ‖G(+)‖ ≠ 1 and G(-) ≠ ∅  */
begin
if STGOOD(+) ≠ ∅ then
GENERALIZATION(STGOOD(+), j*); end
FORMSUBTASK (j*, R(+), G(+), STGOOD(+));
DIAGaRa(proj(R(+)[j*], STGOOD(+));
   (∀s), s ∈ STGOOD(-), val(s) ⊆ t*
       (∀j), j ∈ G(−), s ⊆ val(j)
        FORMSUBTASK (j, R(−), G(−), STGOOD(−);
        DIAGaRa(proj(R(−))[j], STGOOD(−));
        end
    end
end
```

The data in Table 1 is for processing by algorithm INGOT (Example 1) for each object description step by step.

Table 1. The Data for Generating GMRTs (Example 1)

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 2a. The Records of Step-by-Step Results of the Procedure INGOT.

*j**	Class(*j**)	*STGOOD*(1), *STGOOD*(2)
{1};	1	*STGOOD*(1): {{1}};
{2}	1	*STGOOD*(1): {{1,2}};
{3}	2	*STGOOD*(1): {{1,2}}; *STGOOD*(2): {{3}};
{4}	2	*STGOOD*(1): {{1,2}}; *STGOOD*(2): {{3}, {4}};
{5},	2	*STGOOD*(1): {{1,2}}; *STGOOD*(2): {{3}, {4,5}};
{6}	1	*STGOOD*(1): {{1,2}, {2,6}}; *STGOOD*(2): {{3}, {4,5}};
{7}	2	*STGOOD*(1): {{1,2}, {6}}; *STGOOD*(2): {{3,7}, {4,5}};
{8}	1	*STGOOD*(1): {{1,2,8}, {6}}; *STGOOD*(2): {{3,7}, {4,5}};
{9}	2	*STGOOD*(1): {{1,2,8}, {6}}; *STGOOD*(2): {{3,7}, {4,5}, {5,9}}.

Table 2b. The Records of Step-by-Step Results of the Procedure INGOT (continuation).

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

		STGOOD(2): {3,7,12,13},{4,5,10},{5,9,10,13},{10,11},{9,11},{11,12}
{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}.

In Tables 2a and 2b, the sets STGOOD(1) and STGOOD(2) accumulate the sets of objects corresponding to the GMRTs for Class 1 and Class 2, respectively, at each step of the algorithm. Table 3 contains the results of the procedure INGOT.

Table 3. The Sets TGOOD(1) and TGOOD(2) Produced by the Procedure INGOT

TGOOD(1)	*TGOOD*(2)
({1,2,8}, Sunny High)	({4,5,10}, Rain No)
({6,14), Rain Yes)	({5,9,10,13}, Normal No)
-	({10,11}, Mild Normal)
-	({9,11}, Sunny Normal)
-	({3,7,12,13}, Overcast)

The traning set of next example is in Table 4. It contains the description of 25 students (persons) characterized by positive (Class 1) and negative (Class 2) dynamics of intellectual development during a given period of time. The persons are described by factors of the MMPI method modified in Russia by L. Sobchik [21].

Table 4. The Training Set of Data for Example 2

	L	F	K	Hy	Pd	Mf	Pa	Pt	Ma	Class
1	4	3	5	3	3	4	3	4	4	2
2	4	4	5	3	3	3	2	4	4	2
3	4	3	4	3	3	3	3	3	4	2
4	3	3	4	3	3	3	3	3	4	2
5	4	3	4	3	3	4	3	3	3	2
6	5	4	5	3	4	2	4	3	3	2
7	5	3	5	4	4	3	4	4	4	2
8	4	3	4	3	3	4	3	3	3	2
1	3	3	5	4	4	4	3	4	4	1
2	2	3	4	3	3	3	3	3	3	1
3	3	3	5	3	3	2	4	4	3	1
4	3	3	4	3	4	4	2	3	5	1
5	3	3	5	4	4	4	3	4	3	1
6	4	2	4	4	4	4	2	3	3	1
7	3	3	3	2	3	4	3	2	5	1
8	3	3	4	3	4	4	3	3	3	1
9	2	4	5	4	3	4	4	4	4	1
10	3	3	5	3	3	2	4	3	4	1
11	3	4	4	3	3	4	2	3	4	1
12	3	3	4	4	2	4	3	3	4	1

13	5	3	5	4	4	4	4	4	4	1
14	3	3	4	3	4	4	2	4	4	1
15	3	3	4	3	3	2	2	3	4	1
16	5	3	4	2	3	3	4	3	3	1
17	3	3	5	4	3	5	4	4	3	1

Incremental learning is partitioned into several stages (Table 5). Stage 1: training set contains 6 first persons of Class 1 and 6 first persons of Class 2. The result of Stage 1 is in Tables 6.

Stage 2 is a pattern recognition stage; the control set contains persons 7 and 8 of Class 2 and persons 7 – 17 of Class 1. All persons of Class 2 and 5 persons (8, 9, 13, 14, 17) of Class 1 have been recognized correctly. Persons 10, 11, 15 of Class 1 have been recognized as persons of Class 2, and persons 7, 12, 16 of Class 1 have been assigned to neither of these classes. Results of Stages 3-7 are given in Tables 7-11. Each table contains only new rules generated in corresponding stage.

Table 5. Stages of Incremental Learning

Stage	Traning sets		Searching rules for		Rules are in Table
	Class 1	Class 2	Class 1	Class 2	
1	Persons 1-6	Persons 1-6	Yes	Yes	6
2	Pattern recognition				
3	Persons 1-6	Persons 1- 8	No	Yes	7
4	Persons 1-6 and 8, 9, 13, 14, and 17	Persons 1-8	Yes	No	8
5	Persons 1-6, and 8-11, 13-15, and 17	Persons 1-8	Yes	No	9
6	Person 1-17	Person 1-8	Yes	No	10
7	Persons 1-17	Persons 1-8	No	Yes	11

Table 6. The Result of Stage 1

№ of rule	L	F	K	Hy	Pd	Mf	Pa	Pt	Ma	Class	Persons
1					4	4				1	{1,4,5,6}
2	3	3	5							1	{1,3,5}
3	2	3	4	3	3	3	3	3	3	1	{2}
1	4			3	3					2	{1,2,3,5}
3				3	3				4	2	{1,2,3,6}
5		4	5	3						2	{2,4}

Table 7. The result of Stage 3

№ of rule	L	F	K	Hy	Pd	Mf	Pa	Pt	Ma	Class	Persons
1	4			3	3					2	{1,2,3,5,8}
2						3			4	2	{2,3,6,7}
3				3	3				4	2	{1,2,3,6}

4	5		5		4		4			2	{4,7}
5		4	5	3						2	{2,4}

During Stage 4, Rule 4 for Class 2 (see, please, Table 7) is deleted (Rule 4 $\subset$ val(13) for person 13 of Class 1).

During Stage 6, Rule 3 for Class 2 (see, please, Table 7) is deleted (Rule 3 $\subset$ val(11) for person 11 of Class 1).

Table 8. The result of Stage 4.

№ of rule	L	F	K	Hy	Pd	Mf	Pa	Pt	Ma	Class	Persons
1					4	4				1	{1,4,5,6,8,13,14}
2	2				3					1	{2,9}
3			5		3		4	4		1	{3,9,17}
4				4					3	1	{5,6,17}
5				4		4				1	{1,5,6,9,13}
6	3	3							3	1	{3,5,8,17}
7	3	3						4		1	{1,3,5,14,17}

Table 9. The result of Stage 5.

№ of rule	L	F	K	Hy	Pd	Mf	Pa	Pt	Ma	Class	Persons
8			4				2			1	{4,6,11}
9	3	3		3	3	2				1	{3,10,15}
10		4			3	4			4	1	{9,11}
11	3					4				1	{1,4,5,8,11,14}
12			5		3		4			1	{3,9,10,17}
13	3	3	5							1	{1,3,5,10,17}

Table 10. The result of Stage 6

№ of rule	L	F	K	Hy	Pd	Mf	Pa	Pt	Ma	Class	Persons
14		3		2	3					1	{7,16}
15		3	4		3	3		3	3	1	{2,16}
16	3	3		4						1	{1,5,12,17}

Stage 7: correcting the rules for Class 2. The result is in Table 10.

Table 11. Final Rules for Class 2 (Stage 7)

№ of rule	L	F	K	Hy	Pd	Mf	Pa	Pt	Ma	Class	Persons
1	4			3	3					2	{1,2,3,5,8}
2						3			4	2	{2,3,6,7}
5		4	5	3						2	{2,4}
6		3		3	3		3		4	2	{1,3,6}

4 The Integrative Inductive-Deductive Model of Reasoning

We considered only supervised learning, but integrative inductive-deductive reasoning includes unsupervised learning too. This mode of learning is involved in reasoning when a new portion of objects (examples) becomes available over time but without indication of their class membership. In this case, a teacher is absent. Only knowledge is available. A new object description can currently be complete or incomplete, i.e. some attribute values can be unknown or not observable. If we deal with completely described object, then the following results of reasoning are possible: 1) class of new object is determined; 2) there are several classes to which new object can belong to (a situation of uncertainty); 3) object is unknown.

In situation with incomplete object description, we can try to infer hypotheses about unknown values of attributes (it is reasoning based on "past experience"); if an object is unknown, we can try to select a set of training examples that are similar to this object in most degree and to infer new rules for describing this set of examples.

Consider some instances of pattern recognition reasoning by using the rules obtained by the procedure INGOT (Table 3).

Example 1. New weather descriptions are complete, for example, <Overcast, Cool, High, No>; <Sunny, Mild, Normal, No>; <Sunny, Mild, High, Yes>. In all these cases, we find the rules, which allow us to recognize the weather class.

Example 2. If weather descriptions are incomplete, then it is possible that neither of the rules is applicable. But we can use the training set of examples to infer possible variants of weather class. Assume that the weather description is: <Rain, Mild, High>. We construct the decision tree as follows: Rain: Class 2 (Observations 4, 5, 10), Class 1 (Observations 6, 14); Mild: Class 2 (Observation 4, 10), Class 1 (Observation 14); High: Class 2 (Observation 4), Class 1 (Observation 14). It is a situation of uncertainty. Consequently, a step of conditional or diagnostic reasoning is needed. We can consider hypothetically some possible values of attribute Windy; then we comclude that "**if** Windy = No, **then** Class 2"; "**if** Windy = Yes, **then** Class 1". Really, we have obtained the following diagnostic rule: "**If** we observe that (Outlook = Rain) & (Temperature = Mild) & (Humidity = High), then (**if** Windy = No, **then** Class 2; **else** Class 1). Note that, the process of pattern recognition includes some inductive step of reasoning.

Example 3. The weather description is: <Hot, Yes>. The reasoning tree is: Hot: Class 1 (Observations 1, 2), Class 2 (Observations 3, 13); Yes: Class 1 (Observations 2), Class 2 (Observations -). Now we can formulate hypothetically a new forbidden rule: "Hot, Yes → Class 2, **false**" or, in another form, "If we observe that (Temperature = Hot) & (Windy = Yes), then it is never observed Class 2".

Example 4. The weather description is: <Sunny, Mild, Low, No>. Here we meet a new value of Humidity – "Low". Assume that the sets of values of Humidity and Temperature are ordered and Low < Normal < High and Mild < Cool < Cold. Assume that the functions of distance on the attribute domains are also defined. Then in the pattern recognition process, it is possible to infer that <Sunny, Mild, Low, No> is nearer to the example of Class 2 <Sunny, Cool, Normal, No> than to the example of

Class 1 <Sunny, Mild, High, No>. A new feature for Class 2 can be formed, namely, <Sunny, Low >.

One of the possible models of deductive plausible human reasoning based on implicative logical rules can be found in [22].

One of the important problems of integrating deductive and inductive reasoning is connected with creating some on-line interactive method for modifying context of reasoning. Failures in reasoning or appearance of new data can require to add new attributes to the context. The task of incremental generating a logical context for email messages classification is considered in [23]. This article presents a method for incremetal constructing a logical context by the assignment of new attributes to object descriptions. The existing context plays the role of a dynamic schema to help users to keep consistency in their object descriptions.

5 Conclusion

In this paper, the decomposition of inferring good classification tests into subtasks of the first and second kinds is presented. This decomposition allows, in principle, to transform the process of inferring good tests into a “step by step” reasoning process.

We have described some inductive algorithm INGOT for inferring good maximally redundant classification tests. We did not focus on the efficiency of this algorithm; we intend to give more attention to the complexity problems in future contributions.

The development of full on-line integrated deductive and inductive reasoning is of great interest. The main problem in this direction is the development of an on-line interactive model to support users in constructing and modifying the context of deductive-inductive reasoning.

References

1. Godin, R., Missaoui, R., Alaoui H.: Incremental Concept Formation Algorithm Based on Galois (Concept) Lattices. Computational Intelligence, 11(2), 246-267 (1995).
2. Dean van der Merwel, F., Obiedkov, S., & Kouriel, D.: AddIntent: A New Incremental Algorithm for Constructing Concept Lattices. LNCS, vol. 2961 (pp. 372-385) (2004).
3. Kourie, D.G, Obiedkov, S., Watsona B.W., & Dean van der Merwe, F.: An incremental algorithm to construct a lattice of set intersections. Science of Computer Programming 74, 128-142 (2009).
4. Ravindra Patel, K. Swami & R. Pardasani: Lattice Based Algorithm for Incremental Mining of Association Rules. International Journal of Theoretical and Applied Computer Sciences. 1(1), 119-128 (2006).
5. Aaron Ceglar and John F. Roddick: Incremental Association Mining using a Closed-Set Lattice. Journal of Research and Practice in Information Technology, 39(1), 35-45 (2007).
6. Valtchev, P., Missaoui, R., Godin, R., Meridji, M.: Generating Frequent Itemsets Incrementally: Two Novel Approaches Based on Galois Lattice Theory. J. Expt. Theor. Artif. Intell. 14, 115–142 (2002).

7. Utgoff P. E.: An Improved Algorithm for Incremental Induction of Decision Trees. Proceedings of the Eleventh International Conference of Machine Learning (pp. 318-325) (1994).
8. Z. Zheng, G. Wang, Y. Wu: A Rough Set and Rule Tree Based Incremental Knowledge Acquisition Algorithm. LNAI, 2639 (pp. 122-129).Springer-Verlag (2003).
9. Ore, O.: Galois Connexions. Transactions of American Mathematical Society 55(1), 493-513 (1944).
10. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin/Heidelberg (1999).
11. Kuznetsov, S.O.: Machine Learning on the Basis of Formal Concept Analysis. Automation and Remote Control 62(10), 1543-1564 (2001).
12. Naidenova, X.A.: Good Classification Tests as Formal Concepts. F. Domenach, D.I. Ignatov, and J. Poelmans (Eds): ICFCA 2012, LNAI 7278, pp. 211-226 (2012).
13. Naidenova, X.A., Polegaeva, J.G.: SISIF – the System of knowledge acquisition from experimental facts. Alty, J.L., Mikulich, L.I. (Eds.): Industrial Applications of Artificial Intelligence, (pp. 87–92). Elsevier Science Publishers B.V., Amsterdam (1991).
14. Naidenova, X.A.: Data-Knowledge Transformation. V. Solovyev (Ed.): Text Processing and Cognitive Technologies, Issue 3, (pp. 130-151). Pushchino (1999).
15. Naidenova, X. A., Plaksin, M. V., Shagalov, V. L.: Inductive Inferring All Good Classification Tests. In: Valkman, J. (Ed.): Knowledge-Dialog-Solution, Proceedings of International Conference in Two Volumes, vol. 1 (pp. 79-84). Kiev Institute of Applied Informatics, Jalta, Ukraine (1995).
16. Naidenova, X.A.: An Incremental Learning Algorithm for Inferring Logical Rules from Exampled in the Framework of the Common Reasoning Process. E. Triantaphyllou and G. Felici (Eds.): Data Mining and Knowledge Discovery Approaches Based on Rule Induction Techniques, (pp. 89-147). Springer, Heidelberg, Germany (2006).
17. Naidenova, X.A.: DIAGARA: An Incremental Algorithm for Inferring Implicative Rules from Examples. Intern. Journal "Information Theories & Application" 12(2), 171-196 (2005).
18. Naidenova, X.A., Shagalov, V.L.: Diagnostic Test Machine. Auer, M. (Ed.): Proceedings of the ICL 2009 – Interactive Computer Aided Learning Conference, Austria, CD (pp. 505–507). Kassel University Press (2009).
19. Schlimmer, J.S.: Concept Acquisition through Representational Adjustment. Technical Report 87-19. Department of Information and Computer Science, University of California, Irvine (1987).
20. Kuznetsov, S.O.: A Fast Algorithm for Computing All Intersections of Objects in a Finite Semi-lattice. Automatic Documentation and Mathematical Linguistics, 27(5), 11–21 (1993).
21. Sobchik, L.N: Standardized Multi-Factorial Method of Personality Investigation (SMIL (MMPI modified)). Practical Guide. "Rech", Moscow, Russian Federation (2007).
22. Naidenova, X.A.: Constructing Galois Lattices as a Commonsense Reasoning Process. X. Naidenova and D. Ignatov (Eds.): Diagnostic Test Approaches to Machine Learning and Commonsense Reasoning Systems, (pp. 34-70). IGI Global (2012).
23. Ferré, S., & Ridoux, O.: The use of Associative Concepts in the Incremental Building of a Logical Context. Conceptual Structures: Integration and Interfaces, Proceedings of the10th International Conference on Conceptual Structures (ICCS'02), LNCS 2393 (pp. 299-313). Berlin/Heidelberg, Spinger (2002).