Constructing Galois Lattice in Good Classification Tests Mining

Naidenova, X.A.

Military Medical Academy, Saint-Petersburg, Russian Federation

ksennaid@gmail.com

Abstract. A large class of machine learning algorithms based on mining good classification tests is described. The Galois lattice is used for constructing good classification tests. Special rules are determined for constructing Galois lattices over a given context. All the operations of lattice construction take their interpretations in human mental acts.

Keywords. Good classification test, the Galois lattice, machine learning, human mental operations

1 Introduction

This paper provides a framework for solving diverse and very important problems of constructing machine learning algorithms based on the concept of good classification test. Good classification tests (GCTs) are item sets of a special kind. They serve as a basis for mining implicative logical rules from the data sets. The lattice theory is used as a mathematical language for constructing GCTs. The definition of GCTs is based on correspondences of Galois on $S \times T$, where S is a given set of objects and T is a set of attributes' values (items). Any classification test is a dual element of the Galois Lattice generated over a given context (S, T). All the operations of lattice construction take their interpretations in human mental acts.

2 The Rules of the First and Second Kind

In this paper, we focus on conceptual knowledge the main elements of which are objects, properties (attribute values), and classifications (attributes). Taking into account that implications express the links between concepts (object ↔ class, object ↔ property, property ↔ class) we believe classification reasoning to be based on using and searching for only one type of logical dependencies, namely, implicative dependencies. Implicative assertions are considered as logical rules of the first type including the following ones.

Implication: $a, b, c \rightarrow d$. **Interdiction or forbidden rule**: $a, b, c \rightarrow$ *false* (*never*). This rule can be transformed into several implications such as $a, b \rightarrow$ not c; $a, c \rightarrow$

not b; b, $c \rightarrow$ not a. **Compatibility**: a, b, $c \rightarrow VA$, where VA is the frequency of rule's occurrence. The compatibility is equivalent to the collection of implications as follows: a, b $\rightarrow$ c, VA; a, c $\rightarrow$ b, VA; b, $c \rightarrow b$, VA. Generally, the compatibility rule represents a most frequently observed combination of values. The compatibilities can serve as one of the bases of association rules [1], [2]. **Diagnostic rule**: x, $d \rightarrow a$; x, $b \rightarrow$ not a; d, $b \rightarrow$ *false*. For example, d and b can be two values of the same attribute. This rule works when the truth of 'x' has been proven and it is necessary to determine whether 'a' is true or not. If 'x & d' is true, then 'a' is true, but if 'x & b' is true, then 'a' is false. **Rule of alternatives**: a or $b \rightarrow$ *true* (*always*); a, $b \rightarrow$ *false*. This rule is a variant of interdiction.

Rules of the second type or classification reasoning rules are the rules with the help of which rules of the first type are used, updated, and inferred from data (instances). They embrace both inductive and deductive reasoning rules. Deductive steps of reasoning consist of inferring consequences from some observed facts with the use of implications. For this goal, the main forms of deductive reasoning are applied: modus ponens, modus tollens, modus ponendo tollens, and modus tollendo ponens.

Let X be a collection of true values of some attributes (or evidences) observed simultaneously. Let r be an implication, left(r) and right(r) be the left and the right parts of r, respectively **Using implication**: if left(r) $\subseteq X$, then X can be extended by right(r): $X \leftarrow X \cup$ right(r). Using implication is based on modus ponens: if A, then B; A; hence B. **Using interdiction:** let r be an implication $y \rightarrow$ not k. If left(r) $\subseteq X$, then k is the forbidden value for all extensions of X. Using interdiction is based on modus ponendo tollens: either A or B (A, B – alternatives); A; hence not B; and either A or B; B; hence not A. **Using compatibility:** let r = 'a, b, $c \rightarrow k$, VA', where VA is the support of r. If left(r) $\subseteq X$, then k can be used to extend X along with the calculated value VA for this extension. Calculating VA requires a special consideration. Using compatibility is based on modus ponens. **Using diagnostic rules:** let r be a diagnostic rule 'X, $d \rightarrow a$; X, $b \rightarrow$ not a', where 'X' is true, and 'a', 'not a' are some alternatives. Using diagnostic rule is based on modus ponens and modus ponendo tollens. There are several ways for refuting one of the hypotheses: (1) to infer either d or b using existing knowledge (with the use of deductive reasoning rules); (2) inferring (with the use of inductive reasoning rules of the second type) new implications for distinguishing between the hypotheses 'a' and 'not a'; (3) to address an expert. **Using rule of alternatives** is based on modus tollendo ponens: either A or B (A, B – alternatives); not A; hence B; either A or B; not B; hence A.

Generating hypothesis or abduction rule. Let r be an implication $y \rightarrow k$. Then the following hypothesis is generated "if k is true, then y may be true". **Using modus tollens**: let r be an implication $y \rightarrow k$. If 'not k' is inferred, then 'not y' is also inferred.

When applied, these rules generate the reasoning, which is not demonstrative. The deductive reasoning rules act by means of extending an incomplete description X of some evidences and disproving impossible extensions. All generated extensions must not contradict with knowledge (the first-type rules) and an observable real situation, where the reasoning takes place. They must be intrinsically consistent (there are no

prohibited pairs of values in such extensions). The inductive reasoning rules deal with known facts and propositions, observations and experimental results to obtain or correct the first-type rules. For this goal, the main inductive cannons stated by a British logician John Stuart Mill [3] are used: the Method of Agreement, Method of Difference, Joint method of Agreement and Difference.

3 The Concept of Good Classification Test

Denote by R a set of objects and by S the set of indices of objects of R. Let $R(+)$ and $S(+)$ be the sets of positive objects and indices of positive objects, respectively. Then $R(-) = R/R(+)$ is the set of negative objects. Denote by T a set of attributes values or items (values, for short) each of which appears in description at least of one of the objects of R.

The definition of good tests is based on correspondences of Galois G on $S \times T$ and two relations $S \to T$, $T \to S$ [4]. Let $s \subseteq S$, $t \subseteq T$. Denote by t_i, $t_i \subseteq T$, $i = 1,\ldots, N$ the description of object with index i. We define the relations $S \to T$, $T \to S$ as follows: $S \to T$: $t = \text{val}(s) = \{\text{intersection of all } t_i: t_i \subseteq T, i \in s\}$ and $T \to S$: $s = \text{obj}(t) = \{i: i \in S, t \subseteq t_i\}$.

Of course, we have $\text{obj}(t) = \{\text{intersection of all } s(A): s(A) \subseteq S, A \in t\}$. Operations $\text{val}(s)$, $\text{obj}(t)$ are reasoning operations related to discovering the general feature of objects the indices of which belong to s and to discovering the indices of all objects possessing the feature t, respectively.

The operation **generalization_of**(t) = $t' = \text{val}(\text{obj}(t))$ gives the maximal general feature for objects the indices of which are in $s(t)$; the operation **generalization_of**(s) = $s' = \text{obj}(\text{val}(s))$ gives the maximal set of objects possessing the feature $t(s)$.

The generalization operations are actually closure operators [4]. A set s is closed if $s = \text{obj}(\text{val}(s))$. A set t is closed if $t = \text{val}(\text{obj}(t))$.

These generalization operations are not artificially constructed operations. One can perform, mentally, a lot of such operations during a short period of time. We give an example of these operations. Suppose that somebody has seen two films (s) with the participation of Gerard Depardieu ($\text{val}(s)$). After that he tries to know all the films with his participation ($\text{obj}(\text{val}(s))$). One can know that Gerard Depardieu acts with Pierre Richard (t) in several films ($\text{obj}(t)$). After that he may discover that these films are the films of the same producer Francis Veber ($\text{val}(\text{obj}(t))$).

Notice that these generalization operations are also used in FCA [5], [6]: a pair C = (s, t), $s \subseteq S$, $t \subseteq T$, is called a concept if $s = \text{obj}(t)$ and simultaneously $t = \text{val}(s)$, i. e., for a concept C = (s, t) both s and t are closed. Usually, the set s is called **the extent** of C (in our notation, it is the set of indices of objects possessing the feature t) and the set t of values is called **the intent** of C.

Let $S(+)$ and $S(-) = S \setminus S(+)$ be the sets of indices of positive and negative objects respectively.

Definition 1. **A classification test** for $R(+)$ is a pair (s, t) such that $t \subseteq T$ ($s = \text{obj}(t) \neq \emptyset$), $s \subseteq S(+)$ & $t \not\subset t'$, $\forall t'$, t' is the description of an object belonging to $R(-)$.

In general case, a set t is not closed for classification test (s, t), i. e., the condition val(obj(t)) = t is not always satisfied; consequently, a classification test is not obligatory a concept of FCA [5].

Definition 2. A classification test (s, t), $t \subseteq T$ (s = obj(t) $\neq \varnothing$) is **good** for $R(+)$ if and only if any extension $s' = s \cup i$, $i \notin s$, $i \in S(+)$ implies that (s', val(s')) is not a test for $R(+)$.

Definition 3. A good classification test (s, t), $t \subseteq T$ (s = obj(t) $\neq \varnothing$) for $R(+)$ is **irredundant** if any narrowing $t' = t \backslash A$, $A \in t$ implies that (obj(t'), t')) is not a test for $R(+)$.

Definition 4. A good classification test for $S(+)$ is **maximally redundant** if any extension of $t' = t \cup A$, $A \notin t$, $A \in T$ implies that (obj($t \cup A$), t') is not a good test for $R(+)$.

It is possible to show that good maximally redundant tests (GMRTs) are closed maximal frequent itemsets and good irredundant tests (GIRTs) are minimal generators [2] of GMRTs.

Generating all types of tests is based on inferring the chains of pairs (s, t) ordered by the inclusion relation. The set of all concepts ordered by the relation $\leq$, where (s, t) $\leq$ (s^*, t^*) is satisfied if and only if $s \subseteq s^*$ and $t \supseteq t^*$, $s \in 2^S$, $t \in 2^t$, is an algebraic lattice with operations $\cap$, $\cup$ [5].

4 Constructing Galois Lattice

Inferring the chains of dual lattice elements ordered by the inclusion relation lies in the foundation of generating all types of classification tests. The following inductive transitions from one element of a chain to its nearest element in the lattice are used: (i) from s_q to s_{q+1}, (ii) from t_q to t_{q+1}, (iii) from s_q to s_{q-1}, and (iv) from t_q to t_{q-1}, where q, q+1, q-1 are the cardinalities of enumerated subsets.

Inductive transitions can be **smooth** or **boundary**. Under smooth transition, extending (narrowing) of collections of values (objects) is going with preserving a given property of them. These properties are, for example, "to be a test for a given class of objects", "to be an irredundant collection of values", "to be a good test for a given class of objects" and some others. A transition is said to be boundary if it changes a given property of collections of values (objects) into the opposite one. For realizing the inductive transitions we use the following rules: **generalization and specification rules, and dual generalization and specification rules**.

The generalization rule is used to get all the collections of objects $s_{q+1} = \{i_1, i_2, \ldots i_q, i_{q+1}\}$ from a collection $s_q = \{i_1, i_2, \ldots i_q\}$ such that (s_q, val(s_q)) and (s_{q+1}, val(s_{q+1})) are tests for a given class of objects. The termination condition for constructing a chain of generalizations is: for all the extension s_{q+1} of s_q, (s_{q+1}, val(s_{q+1})) is not a test for a given class of positive objects. The generalization rule uses, as a leading process, an ascending chain ($s_0 \subseteq \ldots \subseteq s_i \subseteq s_{i+1} \subseteq \ldots \subseteq s_m$) and the operation generalization_of(s) = s' = obj(val(s)) for each obtained collection of objects in case of inferring GMRTs [7].

The specification rule is used to get all the collections of values $t_{q+1} = \{A_1, A_2, \ldots, A_{q+1}\}$ from a collection $t_q = \{A_1, A_2, \ldots, A_q\}$ such that t_q and t_{q+1} are irredundant collections of values and (obj(t_q), t_q) and (obj(t_{q+1}), t_{q+1}) are not tests for a given class of objects. The termination condition for constructing a chain of specifications is: for all the extensions t_{q+1} of t_q, t_{q+1} is either a redundant collection of values or a test for a given class of objects. This rule has been used for inferring GIRTs [8]. The specification rule uses, as a leading process, a descending chain ($t_0 \subseteq \ldots \subseteq t_i \subseteq t_{i+1} \subseteq \ldots \subseteq t_m$). Inferring GIRTs does not require the operation generalization_of(t) = t' = val(obj(t)) for each obtained collection of values.

Both generalization and specification rules realize the Joint Method of Agreement and Difference [3].

The dual generalization (specification) rules relate to narrowing collections of values (objects).

All inductive transitions take their interpretations in human mental acts. The extending of a set of objects with checking the satisfaction of a given assertion is a typical method of inductive reasoning. For example, Claude-Gaspar Bashet de Méziriak, a French mathematician (1581 – 1638) has discovered (without proving it) that apparently every positive number can be expressed as a sum of at most four squares; for example, 5 = 22 + 12, 6 = 22 + 12+ 12, 7 = 22 + 12+ 12+12, 8 = 22 + 22, 9 = 32. Bashet has checked this rule for more than 300 numbers. In pattern recognition, the process of inferring hypotheses about the unknown values of some attributes is reduced to the maximal expansion of a collection of known values of some others attributes in such a way that none of the forbidden pairs of values would belong to this expansion. The contraction of a collection of values is used, for instance, in order to delete redundant (non-informative) values from it. The contraction of a collection of objects is used, for instance, to isolate a certain cluster in a class of objects. Thus, we distinguish lemons in the citrus fruits.

The boundary inductive transitions are used to get:

(1) all the collections t_q from a collection t_{q-1} such that (obj(t_{q-1}), t_{q-1}) is not a test but (obj(t_q), t_q) is a test, for a given set of objects;

(2) all the collections t_{q-1} from a collection t_q such that (obj(t_q), t_q) is a test, but (obj(t_{q-1}), t_{q-1}) is not a test for a given set of objects;

(3) all the collections s_{q-1} from a collection s_q such that (s_q, val(s_q)) is not a test, but (s_{q-1}, val(s_{q-1})) is a test for a given set of objects;

(4) all the collections of s_q from a collection s_{q-1} such that (s_{q-1}, val(s_{q-1})) is a test, but (s_q,val(s_q)) is not a test for a given set of objects.

All the boundary transitions are interpreted as human reasoning operations. Transition (1) is used for distinguishing two diseases with similar symptoms. Transition (2) can be interpreted as including a certain class of objects into a more general one: squares can be named parallelograms, all whose sides are equal. In some intellectual psychological tests, a task is given to remove the “superfluous” (inappropriate) object from a certain group of objects (rose, butterfly, phlox, and dahlia) (transition (3)). Transition (4) can be interpreted as the search for a refuting example. The boundary inductive transitions realize the Methods of Difference and Concomitant Changes [3].

Note that reasoning begins with using a mechanism for restricting the space of searching for tests: (i) for each collection of values (objects), to avoid constructing all its subsets and (ii) to restrict the space of searching only to the subspaces deliberately containing the desired GMRTs or GIRTs. For this goal, admissible and essential values (objects) are used.

First, consider the boundary transition (1): getting all the collections t_q from a collection t_{q-1} such that (obj(t_{q-1}), t_{q-1}) is not a test but (obj(t_q), t_q) is a test for a given set of objects. For this transition, we use **the inductive diagnostic rule** and a method for choosing values to extend t_{q-1}. We extend t_{q-1} by choosing values that appear simultaneously with it in the objects of $R(+)$ and do not appear in any object of $R(-)$. These values are to be said essential ones.

Consider the boundary inductive transition (3): getting all the collections s_{q-1} from a collection s_q such that (s_q, val(s_q)) is not a test, but (s_{q-1}, val(s_{q-1})) is a test for a given set of objects. For this transition, we use **the dual inductive diagnostic rule** and a method for choosing objects to delete them from s_q. By analogy with an essential value, we define an essential object (index of essential object).

Let s be a subset of objects belonging to a given positive class of objects; assume also that (s, val(s)) is not a test. The object t_j, $j \in s$ is to be said an essential in s if ($s\backslash j$, val($s\backslash j$)) is a test for a given set of positive objects. Generally, we are interested in finding the maximal subset sbmax(s) $\subset s$ such that (s, val(s)) is not a test but (sbmax(s), val(sbmax(s)) is a test for a given set of positive objects.

Table 1. Deductive Rules of the First Type Obtained with the Use of Inductive Reasoning Rules

Reasoning rules	Inferred rules
Generalization rule	Implications
Specification rule	Implications
Inductive diagnostic rule	Diagnostic rules
Dual inductive diagnostic rule	Compatibility rules

The dual inductive diagnostic rule can be used for inferring compatibility rules of the first type. The number of objects in sbmax(s) can be understood as a measure of "carrying-out" for an acquired rule related to sbmax(s), namely, val(sbmax(s)) $\rightarrow$ $k(R(+))$ frequently, where $k(R(+))$ is the name of the set $R(+)$.

The inductive rules generate logical rules of the first type (see, please Table 1).

During the lattice construction, the deductive rules of the first type (implications, interdictions, rules of compatibility (approximate implications), and diagnostic rules) are generated and used immediately for pruning the search space.

5 Reducing Inductive Transition to the Second Type Rules

We give some examples of realizing the generalization rule for inferring all GMRTs. Any realization of this rule must allow, for each element s, the following actions: a) to avoid constructing the set of all its subsets, b) to avoid the repetitive generation of it.

Let S(test) be the partially ordered set of elements $s = \{i_1, i_2, \ldots i_q\}$, $q = 1, 2, \ldots, nt - 1$ obtained as a result of generalizations and satisfying the following condition: (s, val(s)) is a test for a given class $R(+)$ of objects. Here nt denotes the number of positive objects. Let $STGOOD$ be the partially ordered set of elements s satisfying the following condition: (s, val(s)) is a GMRT for $R(+)$. Consider some methods for choosing objects admissible for extending s [7].

Method 1. Suppose that S(test) and $STGOOD$ are not empty and $s \in S$(test). Construct the set V: $V = \{\cup s', s \subseteq s', s' \in \{S(\text{test}) \cup STGOOD\}\}$.

If we want an extension of s not to be included in any element of $\{S(\text{test}) \cup STGOOD\}$, we must use, for extending s, the objects not appearing simultaneously with s in the set V. The set of objects, candidates for extending s, is equal to: CAND(s) = $nts \backslash V$, where $nts = \{\cup s, s \in S(\text{test})\}$.

An object $j^* \in$ CAND(s) is not admissible for extending s if at least for one object $i \in s$ the pair $\{i, j^*\}$ either does not correspond to a test or it corresponds to a good test (it belongs to $STGOOD$).

Let Q be the set of forbidden pairs of objects for extending s: $Q = \{\{i, j\} \subseteq S(+)$: ($\{i, j\}$, val($\{i, j\}$) is not a test for $R(+)\}$. Then the set of admissible objects is $select$(s) = $\{i, i \in$ CAND(s): $(\forall j)$ ($j \in s$), $\{i, j\} \notin \{STGOOD$ or $Q\}\}$.

The set Q can be generated before searching for all GMRTs for $R(+)$.

Method 2. In this method, the set CAND(s) is determined as follows. Let $s^* = \{s \cup j\}$ be an extension of s, where $j \notin s$. Then val(s^*) $\subseteq$ val(s). Hence the intersection of val(s) and val(j) must be not empty. The set CAND(s) = $\{j: j \in nts \backslash s$, val($j$) $\cap$ val(s) $\neq \emptyset\}$.

Table 2. The use of reasoning rules of the second type

Process	Rule of the second type
Forming Q	Generating forbidden Rules
Forming CAND(s)	Joint method of Agreement and Difference
Forming select(s)	Using forbidden rules
Forming ext(s)	Method of Agreement
Function_to_be test(t)	Using implication
Generalization_of($snew$)	Closing operation

The set ext(s) contains all the possible extensions of s in the form $snew = (s \cup j)$, $j \in select(s)$ and $snew$ corresponds to a test for $R(+)$. This procedure of forming $ext(s)$ executes the function generalization_of($snew$) for each element $snew \in ext(s)$.

The generalization rule is a complex process in which both deductive and inductive reasoning rules of the second type are used (please, see Table 2). The knowledge

acquired via a generalization process (the sets Q, L, CAND(s), S(test), *STGOOD*) is used for pruning the search in the domain space.

Searching for only admissible variants of generalization is not an artificially constructed operation. A lot of examples of using this rule in human thinking can be given. For example, if your child were allergic to oranges, then you would not buy these fruits but also orange juice and products containing orange extracts. A good gardener knows the plants that cannot be adjacent in a garden. The problems related to placing individuals, appointing somebody to the post, finding lodging for somebody deal with partitioning a set of objects or persons into groups by taking into account forbidden pairs of them.

6 The Decomposition of Good Test Inferring into Subtasks

To transform good classification tests inferring into an incremental process, we introduce two kinds of subtasks [7], [9]: for a given set of positive examples: 1) Given a positive example t, find all GMRTs contained in t, more exactly, all $t' \subset t$, (obj(t'), t') is a GMRT; 2) Given a non-empty collection of values X such that it is not a test, find all GMRTs containing X, more exactly, all Y, $X \subset Y$, (obj(Y), Y) is a GMRT.

Each example contains only some subset of values from T; hence each subtask of the first kind is simpler than the initial one. Each subset X of T appears only in a part of all examples; hence each subtask of the second kind is simpler than the initial one.

There are the analogies of these subtasks in natural human reasoning. Describing a situation, one can conclude from different subsets of the features associated with this situation. Usually, if one tells a story from his life, then somebody else recalls a similar story possessing several equivalent features. We give, as an example, a fragment of the reasoning of Dersu Usala, the trapper, the hero of the famous book of Arseniev, V. K. [10]. He divided the situation into the fragments in accordance with separate observed facts and then he concluded from each observation independently.

On the shore, there was the trace of bonfire. First of all, Dersu noted that the fire ignited at one and the same place many times. He concluded that here was a constant ford across the river. Then he said that three days ago a man passed the night near the bonfire. It was an old man, the Chinese, a trapper. He did not sleep during entire night, and, in the morning, he did not cross the river and he left. Dersu deduced that only one person was here from the only one track on the sand. He deduced that the person was a trapper on the basis of a wooden rod used for making traps for small animals. That this was the Chinese, Dersu learned from the manner to arrange bivouac. That this was an old man, Dersu deduced after inspecting the deserted footwear: young person first tramples nose edge of foot-wear, but old man tramples heel.

The subtask of the first kind. We introduce the concept of an object's (example's) projection proj(R)[t] of a given positive object 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 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 t, then (obj(t), 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] = $\{t_i: i \in (obj(A) \cap s(+))\}$. 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 object X, then A does not belong to any GMRT different from X.

Forming the projection of A makes sense if A is not a test and the intersection of all positive objects in which A appears is not a test too, i.e., $obj(A) \not\subset s(+)$ and $t' = t(obj(A) \cap s(+))$ does not correspond to a test for $R(+)$. The procedures using these subtasks for inferring GMRTs can be found in [7], [9].

Restricting the search for tests to a sub-context of given context favors completely separating tests [11], i.e., increases the possibility to find values each of which belongs only to one GMRT in this sub-context. Choosing subcontexts can be controlled by a domain ontology.

We introduce the following operations: choosing object (value) for subtasks, forming and reducing subtasks. The choice of values (objects) for forming subtasks requires a special consideration. It is convenient using essential values in an object and essential objects in a projection for the decomposition of inferring good tests into subtasks of the first or second kind. The following theorem gives the foundation for reducing projections [9].

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

Solving subtasks of the first kind initializes deleting object descriptions (item sets), deleting item sets from projection may be followed by deleting values (items) satisfying Theorem 1 or becoming less frequently. Deleting values (items) from item sets may result in deleting item sets not containing any tests for a given class of objects.

7 An Approach to Incremental Inferring Good Tests

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 the 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), that is, whether it is possible to extend the set of objects covered by some existing rules or not; 2) inferring all good classification tests contained in the new object description; 3) checking the validity of rules (tests) for negative objects, and, if it is necessary, modifying the tests that are not valid (test for negative objects is not valid if it is included in a new (positive) object description). The second act can be

reduced to the subtask of the first kind. The third act can be reduced either to the inductive diagnostic rule followed by the subtasks of the first kind or only to the subtask of the second kind. These acts have been implemented in an incremental algorithm INGOMAR for inferring GMRTs [7].

8 Conclusion

The methodology presented in this paper provides a framework for solving diverse and very important problems of constructing machine learning algorithms based on a unified logical model in which it is possible to interpret any elementary step of logical inferring as a human mental operation. This methodology deals with object classifications and their approximations by the use of classification tests constructed in a given features space. This fact allows managing the procedures of discovering knowledge from data by the aid of domain ontology.

References

1. Jipp, J., Gatzer, U., Nakhaeizadeh, G..: Algorithms for Association Rule Mining – a General Survey and Comparison. ACM SIGKDD Explorations 2(1), 58-64 (2000).
2. Zaki, M.J.: Mining Non-Redundant Association Rules. Data Mining and Knowledge Discovery 9, 223-248 (2004).
3. Mill, J. S.: The System of Logic, Ratiocinative and Inductive, Being a Connected View of the Principles of Evidence, and the Methods of Scientific Investigation, Vol.1. John W. Parker, West Strand, London (1872).
4. Ore, O.: Galois Connexions. Transactions of American Mathematical Society 55(1), 493-513 (1944).
5. Wille, R.: Concept Lattices and Conceptual Knowledge System. Computers & Mathematics with. Applications (Oxford, England) 23(6-9), 493-515 (1992).
6. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin/Heidelberg (1999).
7. Naidenova, X .A.: Machine Learning Methods for Commonsense Reasoning Processes. Interactive Models. Inference Science Reference, Hershey, New York (2009).
8. Megretskaya, I. A.: Construction of Natural Classification Tests for Knowledge Base Generation. In: Pecherskij, Y. (ed), The Problem of Expert System Application in National Economy: Reports of the Republican Workshop, pp. 89-93. Mathematical Institute with Computer Centre of Moldova Academy of Sciences, Kishinev, Moldava (1988) (in Russian).
9. 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).
10. Arseniev, V. K.: Dersu, the Trapper: Exploring, Trapping, Hunting in Ussuria. (1st ed.). E. P. Dulton, N.Y. (1941).
11. Dickson, T. J.: On a Problem Concerning Separating Systems of a Finite Set. J. of Comb. Theory 7, 191-196 (1969).