Computing Approximately Minimal Diagnostic Tests

Xenia Naidenova, Vladimir Parkhomenko, Alexander Rudenko

***Abstract*—An algorithm based on mining approximate classification tests is considered. A process of analogical reasoning based on mining these classification tests is described. Implementation of the algorithm is developing specially for machine learning problems in medicine, forest and motor industry. Some new future development directions of the system called DEFINE are advanced. Some examples of the system application are given.**

***Keywords*—Analogy, classification (diagnostic) test, approximation, machine learning.**

I. Introduction

THE aim of the paper is to develop a method of inferring approximate diagnostic (classification) tests. Considering the sets of approximately minimal diagnostic tests as a "characteristics portraits" of object classes we have developed a model of reasoning by analogy.

There is an analogy definition in [1]: "Analogy is a sort of similarity". Then G. Polya shows "how generalization, specialization, and analogy are naturally combined in the effort to attain the desired solution", however he notes: "the essential difference between analogy and other kinds of similarity lies, it seems to me, in the intentions of the thinker". A few methods, which use an analogy, are known in the framework of symbolic machine learning. They usually apply J.S.Mill's similarity principle [2]. For example, case-based reasoning (CBR) can be also considered as inferring by analogy. The precedent is defined as a case, that has been already occurred. It may be an example of similar problem solution. CBR-systems supplement machine learning methodology, i.e. CBR-cycle: similar precedent mining, reusing, control and adaptation, saving of new decision [3].

The model developed in the paper is implemented in the system called DEFINE. The results of this system's application for predicting the type of tree species with the use of aerial photographs is described. The prediction of defects in rotating equipment is also considered. Some new future development directions of the DEFINE are advanced.

The rest of the paper is organized as follows. Sec.II is devoted to defining the characteristic properties of approximately minimal diagnostic tests. Sec.III gives the structure of the main test inferring process. Sec.IV is devoted to a set of examples illustrated all the previous considerations. Some future directions and related work are shown in the last Sec.V.

Xenia A. Naidenova obtained Ph.D. in Computer Science from the St.Petersburg Electrotechnical University. Xenia is a senior researcher of the Group of Psycho Diagnostic Systems Automation at the Military Medical Academy, St.Petersburg, Russia. Email: ksennaidd@gmail.com

Vladimir A. Parkhomenko is a software engineer in the St.Petersburg State Polytechnical University, St.Petersburg, Russia. Email: parhomenko.v@gmail.com

Alexander Rudenko is a student in software engineering in the St.Petersburg Electrotechnical University, St.Petersburg, Russia. Email: sanek1_91@mail.ru

II. A model of reasoning by analogy based on the sets of AMDTs

Approximately minimal or quasi minimal diagnostic test (AMDT) distinguishing an example e from all examples of class Q_x is a collection of attributes $\{A_1, A_2, \ldots, A_k\}$ such that e differs from any example of Q_x by value of at least one attribute of this collection. There are a plethora of algorithms for searching for tests, however if a certain algorithm is chosen then it is possible to consider it as a function $\phi(e, Q_x) = A_1, A_2, \ldots, A_k$.

This function possesses the property that for familiar examples it will return the familiar or the same tests. Let T_{ij} be the set of tests such that any example $e \in Q_i$ is different from all examples of Q_j by at least one test of T_{ij} and for every test $t \in T_{ij}$ there is an example e such that it is different from all examples of Q_j only by this test. In other words, T_{ij} is the necessary and sufficient set of tests for distinguishing Q_i and Q_j. The set T_{ij} is also a function $f(Q_i, Q_j)$ determined by a certain test construction algorithm. The set T_{ij} is considered to be stable or changeable insignificantly with respect to different collections of examples from the same class Q_i.

Let T_{ij} be the set of tests distinguishing the sets Q_i, Q_j of examples. Let T_{xj} be the set of tests distinguishing the sets Q_x, Q_j, where Q_x, Q_i are taken from the same sampling (class) of examples. We assume that tests of T_{ij} and T_{xj}, completely coincide or at least greatly intersect. Analogical reasoning is defined as follows [4], [5], [6]. Assume that the sets T_{ij} for all training sets Q_i, Q_j of examples, $i, j \in \{1, 2, \ldots, nk\}$, where nk is the number of classes, have been obtained by a certain algorithm. Let Q_x be a subset of examples belonging to one and the same but unknown class $x \in \{1, 2, \ldots, nk\}$. Construct sets of tests, $T_{xj} = f(Q_x, Q_j)$, $j \in \{1, 2, \ldots, nk\}$. If Q_x belongs to class $k \in \{1, 2, \ldots, nk\}$, then, in accordance with our assumption, the set of tests T_{xj} must be more similar to T_{kj}, than to T_{ij} for all $i \neq k$, $i \in \{1, 2, \ldots, nk\}$.

This method can be considered as "inference by analogy" because we use the assumption of analogical properties of tests for similar examples constructed with the use of one and the same functional transformation (algorithm). The main problem of this method is related to the choice of the criterion or the measure of similarity between sets of tests. It is more reliable to use several criteria and to make decision based on the rule of "voting" between these criteria.

Tab.II contains a list of quasi-minimal tests for all pairs Q_i, Q_j, $i, j \in \{1, 2, 3, 4, 5\}$of classes. We have two examples

TABLE I
THE SET OF TRAINING EXAMPLES

No\Attr	1	2	3	4	5	6	7	8	9	Q
1	1	1	1	1	1	1	1	1	1	Q_1
2	1	1	2	2	1	1	2	1	1	Q_1
3	2	1	2	2	1	1	3	2	4	Q_1
4	2	1	2	4	1	1	3	2	4	Q_1
5	1	2	1	1	2	1	1	2	1	Q_2
6	1	2	3	3	2	1	3	1	1	Q_2
7	2	2	1	2	1	2	4	2	2	Q_2
8	2	3	4	4	1	1	3	2	2	Q_3
9	2	1	4	4	1	1	4	2	3	Q_3
10	3	4	5	3	3	3	1	3	4	Q_4
11	4	5	5	5	2	2	4	2	1	Q_4
12	3	5	5	1	2	1	5	2	5	Q_4
13	3	4	3	5	4	4	2	4	1	Q_5
14	4	2	3	3	2	4	2	2	1	Q_5

TABLE II
THE SETS OF TESTS FOR GIVEN CLASSES OF OBJECTS

Pairs of Q	Tests T_{ij}	No of tests
$Q_1 - Q_1$	4,7	1
$Q_1 - Q_2$	2	1
$Q_1 - Q_3$	3, 9	2
$Q_1 - Q_4$	1, 2, 3, 5	4
$Q_1 - Q_5$	1, 2, 3, 4, 5, 6	6
$Q_2 - Q_1$	2	1
$Q_2 - Q_2$	4, 7	2
$Q_2 - Q_3$	2, 3, 4	3
$Q_2 - Q_4$	1, 2, 3	3
$Q_2 - Q_5$	1, 6, 7	3
$Q_3 - Q_1$	3, 9	2
$Q_3 - Q_2$	2, 3, 4	3
$Q_3 - Q_3$	2, 7, 9	3
$Q_3 - Q_4$	1, 2, 3, 4,5, 9	6
$Q_3 - Q_5$	1, 2, 3, 4,5, 6, 7, 9	8
$Q_4 - Q_1$	1, 2, 3, 5	4
$Q_4 - Q_2$	1, 2, 3	3
$Q_4 - Q_3$	1, 2, 3, 4,5, 9	6
$Q_4 - Q_4$	4, 6, 7, 9	4
$Q_4 - Q_5$	6, 7	2
$Q_5 - Q_1$	1, 2, 3, 4, 5, 6	6
$Q_5 - Q_2$	1, 6, 7	3
$Q_5 - Q_3$	1, 2, 3, 4,5, 6, 7, 9	8
$Q_5 - Q_4$	6, 7	2
$Q_5 - Q_5$	8	1

be represented for predicting the class to which they belong to: $Q_x = \{(3,4,3,5,3,4,2,1,1),(4,4,3,3,2,4,2,3,5)\}$. Tab.III contains quasi-minimal tests distinguishing Q_x from the sets Q_1, Q_2, Q_3, Q_4, Q_5.

Compare the sets of tests for every pair (Q_xQ_j), $j \in \{1,2,3,4,5\}$ with the sets of tests T_{ij} for all training sets Q_i, Q_j, $i,j \in 1,2,\ldots,5$ (Tab.II). One of the possible decision rules says that if Q_x and Q_y are taken from the same class, then the intersection of corresponding sets of tests $\{Q_xQ_j, Q_yQ_j\}$, $j \in \{1,2,3,4,5\}$ must be greater than for

TABLE III
THE SETS OF TESTS FOR EXAMPLES OF UNKNOWN CLASS

Pairs of Q	Tests T_{ij}	No of tests
$Q_x - Q_1$	1, 2, 3, 4, 5, 6	6
$Q_x - Q_2$	1, 2, 3, 6, 7	5
$Q_x - Q_3$	1, 2, 3, 4, 5, 6,7, 8,9	9
$Q_x - Q_4$	6,7	2
$Q_x - Q_5$	8,9	2

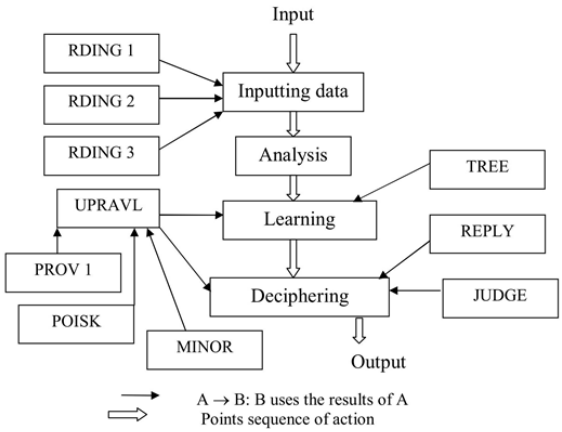

Fig. 1. The Structure of the System DEFINE

Q_xQ_z, for all $z \neq y$. The intersection of two sets of tests is referred to as the subset of coincident tests in these sets. Using this decision rule, we can conclude that Q_x and Q_5 are included in the same class. Let us remark that this example is taken from the real task, and the set Q_x of objects is predicted correctly.

III. DEFINE: THE SYSTEM FOR ANALOGICAL REASONING

The system DEFINE [7], [5], [6] is based on the machine learning method described above. Principal structure of DEFINE is shown in Fig.1.

The pseudocode of the main DEFINE algorithm is depicted in the Fig. 2. Suppose $T = \emptyset$, then an example d_+ or Q_i is taken and QMT is constructed by $POISK()$ to distinguish this example from all the negative examples of Q_j. For each following d_+, it is checked whether at least one of the tests already obtained accepts it.

If there is a test accepting d_+ , then:

1) the pair "index of test, index of d_+" is memorized;
2) the next d_+ is selected from Q_i.

If there is not a test accepting d_+, then $POISK()$ searches for QMT to distinguish it from all the examples of Q_j. In Fig. 3 each $t_{quasi-min}$ is constructed such that it accepts as many positive examples as possible and rejects all d_-. Heuristic algorithm POISK() is used. $kw(d_+, d_-)$ returns the number of attributes distinguishing d_+ from d_- by their values. $mw(attr)$ returns the number of negative examples different from d_+ by value of $attr$.

The advantages of the algorithm POISK() are following: the search for tests is fast; the memory space is small (it is

linear w.r.t the number of d and $attr$). The drawback is the dependence on the ordering of d.

Describe the role of each program of the system. The blocks RDING are intended for inputting initial data and its transformation into attribute-value representation. In particular, if features of objects are continuous, then their discretization is required. In the case of using images of objects, the special complex of programs is used for calculating the values of some characteristics of images, extracting objects, calculating features of objects, and transforming them into attribute-value representation. Program SLOT forms the training and controlling sets of objects based on a given rule or a random choice. The programs UPRAVL, PROV1, POISK and MINOR serve for constructing the sets of tests T_{ij} distinguishing the sets Q_i, Q_j of object examples.

Algorithm UPRAVL
Input: D, Q
Output: T
1. i=0; T=$\emptyset$;
2. **for** ***each*** d_+ *of* Q_i **do**
3. $mt =$ POISK(d_+, Q_j) ;
4. **if** $mt \notin T$ **then**
5. $T = T \cup mt$;
6. discriminatory index(mt)=1;
7. **else** discriminatory index$(mt) + +$;
8. **return**;

Fig. 2. Algorithm UPRAVL of DEFINE system

Algorithm POISK
Input: d_+, Q_j
Output: $t_{quasi-min} \in T$
1. t=$\emptyset$;
2. **for** ***each*** d_- *of* Q_j **do**
3. $kw(d_+, d_-)$;
4. **for** ***each*** $attr \in U$ **do**
5. $mw(attr)$;
6. **while** $Q_j \neq \emptyset$ **do**
7. choose d_- with $min(kw)$;
8. choose $attr$ with $max(mw)$;
9. t=$t \cup attr$;
10. **for** ***each*** $attr$ *distinguishing* d_- *from* d_+ **do**
11. $mw(A) = mw(A) - -$;
12. delete d_- from Q_j;
13. **return**;

Fig. 3. Algorithm POISK of UPRAVL

Program TREE serves for compact representation of the set T_{ij} in the form of special "vector-tree" structure. This structure allows quickly checking whether a test t is contained in T_{ij}. Tests in the form of trees require less volume of memory space. Block "Learning" constructs the set of tests $T = \{T_{ji}\}$ and transforms them into the form of trees. Block "Deciphering" operates the process of constructing tests $T = \{T_{xi}\}$ for unknown or control collection of examples. Program REPLY realizes several decision rules for estimating the degree of similarity between the sets Txi, T_{ji}, $j \in \{1, 2, \ldots, nk\}$, $i = 1, 2, \ldots, nk$. Program JUDGE performs the final decision. Block ""Analysis" investigates initial data and gives the information about the degree of similarity and distinction between given classes of objects and some others informative characteristics. Algorithm TREE serves for transforming the test matrix into the structure of vector-tree or Decision Tree Matrix. Tests are lexicographically ordered and they are represented as the branches of an ordered decision tree, an example of which is given in Tab.IV.

TABLE IV
DECISION TREE MATRIX

Decisions	Ordered decisions	Decision tree
2, 8	1, 3, 2	1 — 3 — 2
1, 3, 2	2, 8	2 — 8 —
2, 8, 15	2, 8, 15	— — 15
7, 5	3, 4, 5	3 — 4 — 5
7, 2, 1	7, 2, 1	7 — 2 — 1
3, 4, 5	7, 5	5

TABLE V
THE VECTOR-TREE REPRESENTATION FOR THE TREE OF TAB.IV

No	1	2	3	4	5	6	7	8	9	10	11
Node	1	8	3	0	2	0	0	2	16	8	0
No	12	13	14	15	16	17	18	19	20	21	22
Node	-1	15	0	0	3	23	4	0	5	0	0
No	23	24	25	26	27	28	29	30	31	32	
Node	7	0	2	30	1	0	0	5	0	0	

The structure of tree is memorized in the form of vector, the example of which, for the tree of Tab.IV, is given in Tab.V.

Generally, the structure of tree is determined as follows: If i-th component of vector-tree contains the value of a node, then $(i + 1)$-th component:

1) contains the index of component containing the value of next node of the same level of tree if such a node exists;
2) is equal to 0 if such a node is absent;

(i + 2)-th component of vector-tree contains:

1) the value of next node of the same branch if such an element exists;
2) 0, if the next node of the same branch is absent and there is not an offshoot of the considered node;
3) -1 if the next node of the same branch is absent but the offshoots of considered node are present.

If j-th component of vector-tree is equal -1, then the value of the next node of offshoot is contained in $(j+1)$-th component of vector-tree. The first element of vector-tree contains the value of the first node of decision tree at the first level.

The first version of DEFINE has been implemented in FORTRAN for running on EC. Its second version of DEFINE has been realized in Turbo C 2.0 DOS 3.0 on computers PC AT/XT and compatible ones with Video adapter CGA or emulating regime CGA. The module Define.exe has been

52 kb executable module. The current DEFINE version is developing for the prediction of defects in rotating equipment. Let us briefly discuss the problem.

A. Software for predicting defects in rotating equipment

We use the spectral information from a vibration detector to predict the defects of rotating equipment. The number of spectral parameters is determined based on the frequency domain and step by frequency.

Let the resulting spectral information be an object. Then the type of defect is referred to a class. The set of objects without defects can also be regarded as a class. The number of objects of each class in the training sample is determined by the required precision of the control deciphering. The number of objects increases until the necessary quality of deciphering is achieved.

The main part of program is implemented in the language PHP. This script language uses HTML, JavaScript and AJAX to process the input data. Inside the program, the data is stored in MySQL database. To make a desktop application, all scripts are packed in the shell written in Delphi.

We plan to make a web-oriented application to get a feedback from the potential users. The program will be independent from the operation system platform, i.e. the internet and web-browser are required. There is another advantage of PHP implementation. It is associated with the high speed of development. The program consists of four main components called Preferences, Data, Analysis and Learning.

Preferences component helps to set the frequency domain and step by frequency. The minimal number of learning steps is calculated here.

In this part, the frequency range and the step by frequency are selected. It is necessary in order to determine the number of parameters which will be used for predicting the defects. Here the prediction accuracy is calculated on the basis of which the minimum number of learning stages is determined.

Data component has a tool for editing the data of defects, i.e. one can add and delete the items from the list of defects. There is also a tool for editing the learning sets within the base of spectral signals. The component is closely related with a MySQL database.

Learning component takes the information from the data base, mines the logical rules (AMDTs) to predict the classes of objects.

Analysis component realizes the process of machine learning. It uses the results from the previous components and has the following functions:

- logical rules (AMDTs) mining between objects of unknown class and objects of learning classes;
- the rules from the previous step are compared with the rules from Learning component;
- predicting the class of unknown objects.

IV. The Results of DEFINEs Application

The system DEFINE has been used for deciphering the predominant species of trees based on aero photographic images with the scale 1 : 3000 [7]. The forest parts have been picked out in the Chagodotchenskij forestry of Vologda region. The following types of trees have been chosen: pine-tree, aspen, birch, and fir-tree. The class of pine-trees has been partitioned into two subclasses: pine-tree 1 (the trees of 70 years old), and pine-tree 2 (the trees of 115 years old).

For training and controlling sets of samples, the trees that are well predicted through stereoscope have been picked out with space distribution approximately equal to 15 -20 trees par 4-5 hectares.

Images of trees have been analyzed by using the stereoscope. An operator has estimated visually the following set of photometric and texture properties of trees: color of illuminated part of crown (1), color of shaded part of crown (2), form of projection of crown (3), form of the edge of projection (4), form of illuminated part of crown (5), form of shaded part of crown (6), structure of crown (7), texture of crown (8), passage from the illuminated to the shaded part crown (9), density of crown (10), closeness of crown (11), form of branches (12), size of branches (13), form of apex (14), and convexity of crown.

For evaluating the color, the scale of color standards has been used. For coding the other properties, the semantic scales have been developed. The number of gradations of properties on the semantic scales was within the limits from 3 to 12. 500 images of trees (100 trees for each species) have been analyzed independently by two operators. The training set of samples has contained 60 descriptions for each species of trees, the set of control samples has contained 40 descriptions for each species of trees.

Two methods have been used for deciphering. The first method (Method 1) deals with predicting the species to which belongs a subset of control trees taken from an unknown class. The decision rule is based on predicting the number of completely coincident tests in the conformable matrixes of tests T_{xi}, T_{ji}, $j, i, x \in \{$birch, pine-tree 1, pine-tree 2, aspen, fir-tree$\}$. The totalities of control examples are considered belonging to the species of trees for which the sum of agreements is the greatest, i.e. the result is i-th species for which $\sum ||T_{xi} \cap T_{ji}||$, $j \in \{$birch, pine-tree 1, pine-tree 2, aspen, fir-tree$\}$ is maximal among all $i \in \{$birch, pine-tree 1, pine-tree 2, aspen, fir-tree$\}$. Deciphering the species of trees by Method 1 has given 100% true reply for each species.

The second method (Method 2) has been implemented for predicting the species to which belongs a single sample of tree not belonging to training sets of trees. This methods is detailed in [7]. In Tab.VI the percentage of correct answers obtained with the use of Method 2 is given. In this case, the part of 22% of the pine trees of 70 years old has been predicted not correctly.

The analysis of the stability of tests has been also carried out. Tab.VII contains the results of experiments according to the data of one of the operators. We observe the disappearance of some tests with decreasing the volume of training set of examples. The number of unique tests proves to be not great. The results demonstrate the possibility to decrease the volume of training set in subsequent experiments. The frequency of occurring attributes in tests shows the usefulness or their informative power. Attributes 10 and 15 did not enter any

TABLE VI
DECIPHERING THE SPECIES OF TREES (METHOD 2)

Tree type	Birch	Pine-tree1	Pine-tree2	Aspen	Fir-tree
Birch	100%				
Pine-tree1		100%			
Pine-tree2		22%	78%		
Aspen				100%	
Fir-tree					100%

TABLE VII
ESTIMATION OF TEST STABILITY (THE SCALE 1 : 3000, OPERATOR 1)

The volume of training set	100%	60%		40%	
Repeated\Unique	R	R	U	R	U
Tree pairs to be deciphered					
Birch-Pine-tree1	4	3	-	2	-
Birch-Pine-tree2	3	3	-	1	-
Birch-aspen	12	10	-	10	1
Birch-fir-tree	2	1	-	2	-
Pine-tree1-Pine-tree2	11	8	1	8	-
Pine-tree1-aspen	3	3	-	2	-
Pine-tree1-fir-tree	5	5	-	3	-
Pine-tree2-aspen	7	6	-	4	-
Pine-tree2-fir-tree	4	4	2	2	-
Aspen-fir-tree	2	2	-	2	-

TABLE VIII
RESULTS OF PREDICTING THE DEFECTS IN ROTATING EQUIPMENT

Spectral signals	Q_1	Q_2	Q_3
Defects in rolling bearings	83%	89%	81%

test, so they are the least informative. Attribute 2 possesses the greatest informative power. Attributes 1, 3, 4, 5, 6, 7, 12, and 14 are informative.

The system DEFINE has been applied very successfully for processing spectral information [8]. We have made also some experiments for predicting the defects in rotating equipment. There were 100 spectral signals both for each class of object defect and for the normal object working without defects. The program has been tuned as follows: the frequency domain was from 0 to 100 Hz and the step by frequency was 1 Hz. The prediction accuracy was 85%. Three different spectral signals of defects have been given for predicting their classes. The results of the prediction see, please, in Tab.VIII.

V. RELATED WORK AND SOME FUTURE DIRECTIONS FOR THE DEFINE APPLICATION

Analogy is one the basic human commonsense reasoning operation. Generalization of a set of familiar objects allows solving many predicting and identifying problems. The system DEFINE is useful for solving such problems in which objects of alternative classes are not statistically distinguished by values of their observed features, but have, nevertheless, a stable relationship between their elements, for example, a set of informative spectral channels in spectral images of the Earth's land, minerals, etc. The problem solved by the system DEFINE can be the following ones:

- Restoration and correction of images by analogy;
- The prediction of missing data, for example, missing values of some characteristics during psychological testing of respondents;
- Identifying changes in the images of the same objects or persons;
- Clustering of text documents.

One of the most important problems of image processing is the problem of restoring or correcting corrupted images In many applications, it is of crucial importance. Examples are many medical images obtained from various sources like ultra-sound or x-rays or images from cameras on remote locations. It is often impossible to correct images distorted by damaged optics or undesirable movement, or by spots because we have not any precise information about damage nature and parameters.

In paper [9], the authors propose a method to restore images based on image analogies. Having one "good" image, before distortions and the same image with distortions, they develop a filter using neural networks that transforms the distorted image to the original without distortions. A single filter reverses all the distortions without separating them or even trying to determine them. Such a filter is later used to process all images from the same source.

Horng and Chen [10] proposed a real-time driver fatigue detection system based on eye tracking and dynamic template matching. The system consists of several parts: face detection, eye detection, eye tracking and fatigue detection. To remedy the low accuracy of the previous variant of the system in matching and inefficiency in search, the authors of [10] proposed two new matching functions, the edge map overlapping (EMO) and the edge pixel count (EPC) to enhance matching accuracy. The algorithms of the system DEFINE can be used directly for face, eye, and fatigue detection.

Document clustering aims to discover natural grouping among documents in such a way that documents with in a cluster are similar to one another and are dissimilar to documents in other clusters [11]. The system DEFINE can be successfully used for clustering of text documents and their topic extracting [12]. For this goal, it is advisable to use as document features (attributes) not only keywords but also typical word associations (patterns) and phrases reflecting topic of texts, professional terminology, style and manner of text representation.

VI. CONCLUSION

The method of classification reasoning 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 a mode of analogical commonsense reasoning is used [13]. The peculiarities of this model include the fast algorithms for constructing approximately minimal diagnostic tests and the use of memorized training examples of objects to identify new objects and predict their class membership.

REFERENCES

[1] G. Polya, "Induction and analogy in mathematics," in *Mathematics and plausible reasoning*, vol. 1. Princeton: Princeton University Press, 1954.

[2] J. S. Mill, *The System of Logic Ratiocinative and Inductive Being a Connected View of the Principles of Evidence, and the Methods of Scientific Investigation*. Longmans, 1872.

[3] A. Aamodt and E. Plaza, "Case-based reasoning; foundational issues, methodological variations, and system approaches," *AI Communications*, vol. 7, no. 1, pp. 39–59, 1994.

[4] K. Najdenova, "A relational model of the analysis of experimental data," *Engineering Cybernetics*, vol. 20, no. 4, pp. 99–115, 1982.

[5] X. A. Naidenova and J. G. Polegaeva, "DEFINE - the system for generating hypotheses and reasoning by analogy on the basis of inferring functional dependencies," in *The Problem of Expert System Creation. Preprint of the Leningrad Institute for Informatics and Automation of the USSR Academy of Sciences (LIIAN)*, V. Ponomarev, Ed. Leningrad, USSR: LIIAN, 1989, vol. 111, pp. 20–21, (in Russian).

[6] X. Naidenova and V. Parkhomenko, "A fast heuristics for inferring approximately minimal diagnostic tests," in *Recent advances in mathematical methods in applied sciences: Proc. of the 2014 Int. Conf. on Mathematical Models and Methods in Applied Sciences (MMAS '14), Proc. of the 2014 Int. Conf. on Economics and Applied Statistics (EAS '14)*, Y. Senichenkov and et al., Eds. WSEAS, 2014, pp. 330–334.

[7] X. Naidenova, J. Krilova, and I. Gnedash, "Deciphering objects based on relations of distinction and identity of objects descriptiosn in multivalued feature spaces," in *"Applying distant data and computers for investigating the natural resources of the Earth". Preprint*, V. Ponomarev, Ed. LSICC (Leningrad Scientific Investigative Computer Centre) of cademy of Sciences of USSR, Leningrad, 1983, vol. 111, pp. 29–46, (in Russian).

[8] X. A. Naidenova and J. G. Polegaeva, "Application of similarity-distinction relations for processing multi-spectral information," in *Theses of papers of All Union Conference "Image processing and remote investigations"*, V. P. Pyatkin, Ed., vol. 3, Novosibirsk, USSR, 1983, pp. 67–68, (in Russian).

[9] M. Tuba and J. Tasic, "Image analogies based filters for composite distortions," *International Journal of Mathematics and computers in simulation*, vol. 7, pp. 532–540, 2013.

[10] W.-B. Horng and et. al, "Improvements of driver fatigue detection system based on eye tracking and dynamic template matching," *WSEAS Transactions on Information Science and Applications*, vol. 9, pp. 14–23, 2012.

[11] P. Perumal and R. Nedunchezhian, "A hybrid machine learning based k-means clustering algorithms for document clustering," *WSEAS Transactions on Information Science and Applications*, vol. 9, pp. 282–293, 2012.

[12] X. Naidenova. (2006) The role of machine learning in processing natural language processing. (in Russian). [Online]. Available: http://www.dialog-21.ru/digests/dialog2006/materials/pdf/Naidenova.pdf

[13] X. A. Naidenova and J. G. Polegaeva, "Model of human reasoning for deciphering forest's images and its implementation on computer," in *Theses of papers and reports of school-seminar "Semiotic aspects of the intellectual activity formalization"*, Kutaisy, Georgia Soviet Socialist Republic, 1985, (in Russian).