UDC 001.891:65.011.56

PROBLEM OF A DISCRETE DATA ARRAY APPROXIMATION BY A SET OF ELEMENTARY GEOMETRIC ALGORITHMS

Povkhan I. F. – Dr. Sc., Assistant Professor, Assistant Professor at the System Software Department, State Higher Education Institution Uzhhorod National University, Uzhhorod, Ukraine.

Mitsa O. V. – Dr. Sc., Assistant Professor, Head of the Information Control Systems and Technologies Department, State Higher Education Institution Uzhhorod National University, Uzhhorod, Ukraine.

Mulesa O. Y. – Dr. Sc., Assistant Professor, Assistant Professor at the Cybernetics and Applied Mathematics, State Higher Education Institution Uzhhorod National University, Uzhhorod, Ukraine.

Melnyk O. O. – PhD, Assistant Professor, Assistant Professor at the System Software Department, State Higher Education Institution Uzhhorod National University, Uzhhorod, Ukraine.

ABSTRACT

Context. In this paper, a problem of a discrete data array approximation by a set of elementary geometric algorithms and a recognition model representation in a form of algorithmic classification tree has been solved. The object of the present study is a concept of a classification tree in a form of an algorithm trees. The subject of this study are the relevant models, methods, algorithms and schemes of different classification tree construction.

Objective. The goal of this work is to create a simple and efficient method and algorithmic scheme of building the tree-like recognition and classification models on the basis of the algorithm trees for training selections of large-volume discrete information characterized by a modular structure of independent recognition algorithms assessed in accordance with the initial training selection data for a wide class of applied tasks.

Method. A scheme of classification tree (algorithm tree) synthesis has been suggested being based on the data array approximation by a set of elementary geometric algorithms that constructs a tree-like structure (the ACT model) for a preset initial training selection of arbitrary size. The latter consists of a set of autonomous classification/recognition algorithms assessed at each step of the ACT construction according to the initial selection. A method of the algorithmic classification tree construction has been developed with the basic idea of step-by-step arbitrary-volume and structure initial selection approximation by a set of elementary geometric classification algorithms. When forming a current algorithm tree vertex, node and generalized attribute, this method provides alignment of the most effective and high-quality elementary classification algorithms from the initial set and complete construction of only those paths in the ACT structure, where the most of classification errors occur. The scheme of synthesizing the resulting classification tree and the ACT model developed allows one to reduce considerably the tree size and complexity. The ACT construction structural complexity is being assessed on the basis of a number of transitions, vertices and tiers of the ACT structure that allows the quality of its further analysis to be increased, the efficient decomposition mechanism to be provided and the ACT structure to be built in conditions of fixed limitation sets. The algorithm tree synthesis method allows one to construct different-type tree-like recognition models with various sets of elementary classifiers at the preset accuracy for a wide class of artificial intelligence theory problems.

Results. The method of discrete training selection approximation by a set of elementary geometric algorithms developed and presented in this work has received program realization and was studied and compared with those of logical tree classification on the basis of elementary attribute selection for solving the real geological data recognition problem.

Conclusions. Both general analysis and experiments carried out in this work confirmed capability of developed mechanism of constructing the algorithm tree structures and demonstrate possibility of its promising use for solving a wide spectrum of applied recognition and classification problems. The outlooks of the further studies and approbations might be related to creating the other-type algorithmic classification tree methods with other initial sets of elementary classifiers, optimizing its program realizations, as well experimental studying this method for a wider circle of applied problems.

KEYWORDS: algorithmic classification tree, image recognition, classification, classification algorithm, branching criterion, geometric algorithm.

ABBREVIATIONS

TS is a training selection;
ST is a test selection;
RS is a recognition system;
IR is an image recognition;
GA is a generalized attribute;
RF is a recognition function;
LCT is a logical classification tree;
ACT is an algorithmic classification tree;
GAT is a generalized attribute tree;
BAS is a branched attribute selection.

NOMENCLATURE

M^n is a manifold of real vectors of dimensionality n;

n is a total number of the problem attributes (the attribute space dimensionality;

w_i is a discrete object of the initial TS;

u is a value of a class of discrete object w belonging;

m is a total number of training pairs (known classification objects) of the initial TS;

l is a value of a class of discrete object x belonging;

f_R is a recognition function;

DOI 10.15588/1607-3274-2021-3-10

x_S^i is a vector (point) of a space set by the problem condition;

H_i is a set of classes set by the initial TS;

G_i is a set of manifolds of relevant objects w_j;

P_i is a constructed generalized attribute;

A_i, A_j is a pair of arbitrary vectors with M^n;

a_j, b is a pair of arbitrary real numbers;

c_0, c_1 is a center of mass of the classes H_0 and H_1;

n_0, n_1 is a capacities (masses) of the classes H_0 and H_1;

z is an arbitrary point of the classes H_0 and H_1 partition plane;

r is a fixed hemisphere radius in the attribute space of the problem;

S is a vector of the classes H_0 and H_1 partition;

ξ is a vicinity of a fixed point with M^n;

ρ is a constructed RS efficiency;

K_0, U is a manifold of all the class H_0 hemispheres;

K_1, Y is a manifold of all the class H_1 hemispheres;

k_i, l_i is a hemisphere sequence;

$d_{\min}(w_i)$ is a radius of a hemisphere with a center at the point w_i;

$d(w_0, w_1)$ is a distance between the points w_0 and w_1;

d_0 is a minimal distance to a neighboring class;

p_i is a capacity of the relevant GA in a form of a hyperellipse;

E is a GA manifold in a form of elementary hyperellipses;

S_i^j is a GA set in a form of hyperplanes;

P_i^j is a GA set in a form of hyperparallelepipeds;

K_i^j is a GA set in form of hyperspheres;

q is a total number of GA in the algorithm tree.

INTRODUCTION

An important problem that often faces the engineer is the task of automatic building the systems for processing large arrays of information and decision-making systems. Effective solving these problems will allow one to pass to computer a difficult work of designing a complex recognition system and to release the creative engineer's potential to solve other, more important, problems. In addition, automation of algorithmic and software design of specific recognition systems is the key to their high efficiency for each real task, and, therefore, it will ensure the rapid development of various branches of science and technology [1–5].

As of today, more than 3.500 recognition algorithms (based on different approaches and concepts) are known having certain limitations when being used (i.e. accuracy, speed, versatility, reliability etc). In addition, each of the algorithms is limited to a specific application, and this is certainly the weakest point not only of these algorithms, but also of recognition systems based on the relevant concepts. Information technologies based on the mathematical models of image recognition are widely used in socio-economic and environmental information processing systems. This is due to the fact that this approach allows one to eliminate the shortcomings of classical methods and achieve fundamentally new results, rationally using the capacity of computer systems [3].

The most of available methods of training selection processing in the recognition function construction do not allow one to achieve the required level of the recognition system accuracy and adjust their complexity in the process of designing these systems. The methods of constructing recognition systems based on the classification tree methods [2] are free of such shortcoming. The peculiarity of the logical tree method is the possibility of complex use of many known recognition algorithms (methods) to solve each specific problem of constructing a recognition scheme. It is based on a single methodology – the optimal approximation of the training selection by a set of generalized attributes (autonomous algorithms) included in some scheme (operator), built in the course of training process.

The **object of study** is a general concept of the decision tree, namely, the algorithmic classification tree constructions built on the basis of a scheme of approximating the training selection array by a set of elementary classification algorithms.

The decision tree concept [6–8] and its relevant library realizations (LightGBM, XGBoost), though being close in idea (the logic tree scheme), do not allow realization of the concept of algorithmic classification tree consisting of a set of vertices – the different-type autonomous classification algorithms. One should take into account that the generalized attribute set tree is, in fact, the algorithmic tree reflection.

The **subject of studies** includes the methods, algorithms and schemes of constructing the classification algorithmic trees in the image recognition tasks.

Presented study allows overcoming these system limitations of constructing the recognition systems by synthesizing the algorithmic trees of classification. Their main specific feature is the modular tree structure and the possibility of applying an arbitrary recognition algorithm or method in the process of synthesizing the classification scheme [9–11].

The **objective of the work** is to elaborate the method of the RS classification model construction on the basis of the scheme of training selection array approximation by a set of elementary classification algorithms. Note that the classification system schemes obtained are characterized by a tree-like structure and the presence of autonomous classification algorithms as their own structural elements.

DOI 10.15588/1607-3274-2021-3-10

1 PROBLEM STATEMENT

In this paper, we shall consider algorithmic implementations of a generalized attribute (algorithm) formation in the algorithmic tree method, which is used in the finished software system and is based on the geometric separation of images using geometric objects. It should be noted that for an arbitrary geometric algorithm or recognition method the main role is played by the distance between the objects in the space for which the recognition problem is solved.

Let M^n be a manifold n of dimensional real vectors (i.e. we deal with the n-dimensional space). Under images we mean the system of subsets (classes) $H_1,...,H_k$ in the manifold M^n. The training selection, in turn, is set by a sequence:

$$(w_1, f_R(w_1),...,w_1, f_R(w_m)). \quad (1)$$

Here $w_i \in M^n, f_R(w_j) \in \{1,2,...,k\}(i=1,...,m)$, and if $f_R(w_i) = u, u \in \{1,...,k\}$, then the object w_i belongs to the image H_u. The objects w_i are, in fact, the vectors (points) $x_1^i, x_2^i,...,x_S^i$ set in this space.

To simplify explanation of the principal idea, let us assume that $k \in \{0,1\}$, i.e. the objects may belong to the two classes H_0 or H_1 (the binary classification case) only. Furthermore, we shall assume that a certain metrics is set at the space M^n, i.e. this space is metrical. We shall denote some distance between the vectors A_i and A_j in this space as $\|A_i - A_j\|$. It is evident that the way of choosing the distance between them has several solutions, for instance:

$$\begin{aligned} &\text{a)} \|A_i - A_j\| = \sqrt{\sum_{m=1}^{S}(x_i^m - x_j^m)^2}; \\ &\text{b)} \|A_i - A_j\| = \sum_{m=1}^{S}\left|x_i^m - x_j^m\right|; \\ &\text{c)} \|A_i - A_j\| = \max_m\left|x_i^m - x_j^m\right|. \end{aligned} \quad (2)$$

Note that here $A_i = x_1^i,...,x_S^i$, $A_j = x_1^j,...,x_S^j$. For example, the hyperplane algorithm operation results in one or more generalized attributes P_j, which, in fact, are the parameters of some hyperplanes that allow all the classes H_i of the manifold M^n to be partitioned. A quite large number of such partitioning (hyeperplane constructing) ways are available here.

Thus, this work faces a problem of constructing the classification tree model with parameters p and structure L that should be optimal $F(L(p, w_i), f_R(w_i)) \to opt)$ with respect to the initial TS data.

2 REVIEW OF THE LITERATURE

This study continues a cycle of works devoted to the problems of discrete object recognition (classification) tree-like schemes [9–15]. They raise issues of constructing, using and optimizing logical trees. Thus, it is known from [16] that the resulting classification rule (scheme) constructed by an arbitrary method or algorithm of branched attribute selection has a tree-like logical structure. The logical tree consists of vertices (attributes) grouped by tiers and obtained at a certain step (stage) of the recognition tree construction [17]. An important problem that arises from [10] is that of synthesizing recognition trees, which will be actually represented by the algorithm tree (graph).

In contrast to existing methods, the main feature of tree recognition systems is that the importance of individual attributes (groups of attributes or algorithms) is determined with respect to the function that defines the objects partition into classes [18, 19]. Please keep in mind that the numerical value of the specified importance characterizes the error of the objects partition into classes. Suppose that at the first step of recognition tree constructing, an arbitrary recognition algorithm is used, and, as a result, we obtain some formula (generalized attribute). This formula implements a certain level of recognition. The function takes several values depending on the attribute values. These values characterize the paths (classes), and there are ways in which this formula "works well", there are those in which it "works bad", and there is no improvement in the level of recognition. It is clear that for these attribute values (paths) one has to take other algorithm that will create other formula (generalized attribute), etc. Thus, in recognition methods based on (algorithmic) trees, it is necessary to repeat such algorithm selection until we obtain the required level of recognition quality [11].

A basic issue of selecting branching criterion in the classification tree scheme structure, whose selection may be affected by the problem specificity, was raised in [20]. The issues of generation and interpretation of classification rules in the LCT structures are raised in [14]. The problem of assessing the attribute informativity [15] in constructing the classification tree vertices remains topical and requires further studying the direction of preprocessing and analyzing the initial data structures [21]. Moreover, the presence of limitations in terms of generating the LCT structure is the disadvantage of the functionalities of assessing the attribute quality in the above works. Thus, the work [8] opens fundamental questions regarding decision tree generation for the case of uninformative attributes. A possible way to improve this work might be the use of attribute combinations and sets to generate informative vertices of the LCT structures. The ability of the LCT/ACT (i.e. logical and algorithmic classification trees) structures to perform one-dimensional branching and analyze individual variable impact, importance and quality allows one to work with different-type variables in the form of predicates.

DOI 10.15588/1607-3274-2021-3-10

For the case of the ACT models, the question of assessing the quality of the corresponding structure branches, i.e. of the autonomous classification algorithms, remains relevant [10]. Here the search for effective criteria for branching tree structures is the way of improving the ACT structure methods. This concept of logical trees is actively used in the intellectual data analysis, where the ultimate goal is to synthesize a model that predicts the value of the target variable based on a set of initial data at the system input. In [22], an important issue of analysis of the classification quality of the decision tree sets was raised. A possible way to improve the overall quality of classification is the use of decision tree ensembles, bugging and boosting mechanisms [23–25]. Note that these schemes will provide the classification model with necessary accuracy only if there is an effective branching criterion [24]. In this case, as a test of the constructed LCT models, it is appropriate to use the cross-validation scheme taken from [26].

The disadvantage of this approach is the final complexity of the classification models and the need in the procedure of final model cutting-off. In regards to application, there are a large number of methods and algorithms that implement the decision tree concept, but the most widely used and widespread are their two representatives (i.e. the C4.5/C5.0 Ross Quinlan scheme and the CART (Classification and Regression Tree) scheme). The C4.5/C5.0 scheme uses the so-called theoretical-information criterion as a criterion for selecting a node, while the CART algorithm is based on calculating the Gini index (a statistical indicator of the attribute difference), which takes into account the relative distances between class distributions within metrics. The main examples, parameters and mechanisms of this classification tree scheme can be developed from open resources [27–31]. The shortcoming here is the relatively weak efficiency in terms of vertex selection in comparison with other modern methods and schemes of the LCT structures. The main idea of methods and algorithms of branched selection of ACT algorithm attributes and vertices, in contrary to the neural network approach [32–34], can be defined as the optimal approximation of some initial TS by a set of ranked classification algorithms. Then the central question from [11] comes to the fore, i.e. a problem of choosing an effective branching criterion, selecting vertices, attributes and features of discrete objects for the LCT schemes and choosing algorithms for ACT.

3 MATERIALS AND METHODS

At the next stage of the study, for the case of binary classification, we shall construct such a generalized attribute (a particular hyperplane) that would separate these classes in the most efficient way. One of the easiest ways of such construction would be:

$$P(x_1^i, x_2^i, ..., x_s^i) = \begin{cases} 1, \text{if} \quad \sum\limits_{j=1}^{S} a_j x_j^i + b \geq 0; \\ 0, \text{if} \quad \sum\limits_{j=1}^{S} a_j x_j^i + b < 0. \end{cases}$$

$$P(x_1^i, x_2^i, ..., x_s^i) = \begin{cases} 1, if \quad \sum\limits_{j=1}^{S} (x_j^i - a_j)^2 - b^2 \leq 0; \\ 0, \text{if} \quad \sum\limits_{j=1}^{S} (x_j^i - a_j)^2 - b^2 > 0. \end{cases}$$

Note that here a_j, b are the arbitrary real numbers.

Also, it is clear that a simple and effective division of the classes H_0, H_1 is possible only if the hypothesis of compactness in the space of these classes is valig. The latter means that all points of each of these classes are at a relatively small distance from each other, i.e. there are such points (centers of mass of these classes), around which they are grouped. Obviously, in this case it is appropriate to choose the arithmetic mean of the submabifolds H_0, H_1 as the above centers. Thus, when constructing the GA, it is sometimes important to take into account the capacity of each of the submanifolds. We shall mean by the capacity (mass) of a manifold the number of pairs of objects from the training selection of type (1), for which relation $f_R(w_i) = j$ holds true:

$$c_0 = \frac{\sum\limits_{w_i \in H_0} w_i}{n_0}, c_1 = \frac{\sum\limits_{w_i \in H_1} w_i}{n_1}. \quad (3)$$

Just the points $c_0 = (c_1^0, ..., c_S^0)$ and $c_1 = (c_1^1, ..., c_S^1)$ shall be the centers of mass of the classes H_0 and H_1, while the numbers n_0 and n_1 will be their capacities (masses). Since n_0, n_1 have to influence directly P (i.e. the hyperplane location in space), it seems reasonable to assume as follows:

$$\frac{s - c_0}{c_1 - s} = \frac{n_0}{n_1}. \quad (4)$$

It follows directly from (4) that:

$$s = \frac{n_1 c_0 + n_0 c_1}{n_0 + n_1}. \quad (5)$$

In accordance with the training selection data, the centers of mass of the classes H_0, H_1 and their capacities

DOI 10.15588/1607-3274-2021-3-10

n_0, n_1, as well as the point S, are calculated. This point S will be a result of selection of the most successful division of the classes H_0, H_1. Such selection will be the better, the wider are the relevant classes separated from each other in space. After constructing the attribute P, all the points w not exceeding s belong to the class H_0, whereas all the remaining points belong to the class H_1.

In view of all mentioned above, we will step-by-step describe the scheme of division of classes by means of a hyperplane. Note only that in our case the training process is carried out at once for all objects of the initial selection, but this assumption is not a limitation.

At the initial stage, based on the training selection data, calculation of the centers of mass and capacity of the classes H_0, H_1 is carried out. Note that these two operations can be done with a single processing of the training selection. The next step is to find the point of division between the centers of mass of the classes, through which the desired plane will pass separating them. Therefore, first the distance between the centers of mass of the classes is calculated depending on the type of the distance determination in space, which was fixed at the beginning of the algorithm. At the second stage, depending on the capacity of the classes H_0, H_1, location of the point S that lies on the line connecting them is to be found. We draw a plane through a certain point S normally to the line connecting the centers of mass of two opposite classes. The equation of this plane will look like:

$$(z-s)\cdot(c_1-c_0)=0. \qquad (6)$$

Note that z denotes here an arbitrary point of the plane (6), and we deal with a scalar product of the vectors $x \cdot y$. Then the following attribute could be taken as the generalized attribute P:

$$P(z)=\begin{cases}1, if\,(z-s)\cdot(c_1-c_0)\geq 0;\\ 0, if\,(z-s)\cdot(c_1-c_0)<0.\end{cases} \qquad (7)$$

If $P(z)=1$, then z should be related to the class H_1, while if $P(z)=0$, z should be related to the class H_0. Thus, in this case the classes H_0 and H_1 will be separated by a plane (6) that crosses a straight line, which connects the centers of mass of the above classes at the point S and, in addition, is orthogonal to this line.

After the generalized attribute P is built, it should be checked. Such check of the generalized attribute P shall include repeated checking all the objects of the training selection for their recognition correctness. One essential peculiarity should be noted here, namely, the errors of two categories are possible in testing the attribute P during the recognition tree method operation, when the attribute P will be only one of many constructed generalized attributes. The first category of errors in not critical, when the dividing plane due to not fully correct construction will fail to cover a part of objects belonging to one of classes (it is not critical, because 'not covered' part of objects will be passed for processing to another recognition algorithm). The second error category, a critical one, will lead to impossibility to build the dividing plane and occurs in case the above plane covers objects of other class.

The algorithmic implementation of both training and recognition suggested above can be applied to various class regions, the full separation of which is achieved especially for a set of objects of class selection from non-intersecting space regions. Such sets of objects (points) can be completely separated from those of points from any other class, if their regions in space do not intersect. Note that the main disadvantage of this implementation is that, in general, separation of the class regions is, as a rule, not optimal and depends directly on the efficiency of the choice of the attributes. The training selection informativity and adequacy are also important [9, 15].

Consider the local algorithm of generating attributes in a form of hyperspheres. The most efficient and universal geometric algorithm is the hypersphere one. The scheme of this algorithm, which will be described below, in order to form generalized attributes (the GA sets) , builds at each step a hypersphere of the following form: $\sqrt{\sum_{i=1}^{n}(x_i-a_i)^2}\leq r$. The algorithm of training selection approximation by hyperspheres is very similar, in general, to the hyperellipse algorithm (and is its further conceptual development), where the attribute is described by inequality $\sum_{i=1}^{n}c_i(x-x_i)^2\leq r$. Its high versatility, in contrary to other algorithms, is based on the fact that even if the compactness hypothesis is not fulfilled, it is possible to construct a sequence of attributes that would separate objects of one class from those of the other one. Note that the only condition for this is prohibition of overlapping objects of different classes. The principle of operation of this algorithm is to approximate the region of the corresponding class by a set of hyperspheres that would cover all its objects (points). Moreover, the main attention should be paid not so much to the description of the class region by hyperspheres, as to the efficiency and cost-effectiveness of this description. That is, the process of approximation of an arbitrary class should take place with the construction of a minimum number of hyperspheres (generalized attributes).

Note that in order to store in the computer memory some hyperspheres that approximate a certain region, it is enough for each of them to memorize the coordinates of the hypersphere center and radius. If we give a certain parameter (radius), each point in the space (training selection element) will become a hypersphere. Let us describe a simplified scheme of the hypersphere algorithm and consider the following possible assumptions:

DOI 10.15588/1607-3274-2021-3-10

a) there are no object with M^n in the ξ-vicinity of each point w_i of the initial selection;

b) there are objects with M^n in the ξ-vicinity of point w_i, but all of them belong to the same class together with point w_i;

c) there are objects with M^n in the ξ-vicinity of point w_i, but they may belong to different classes.

Assumption (a) is not entirely preferable, as it is unlikely that some approximation by a certain sequence of geometric figures will be better than the training selection itself. Assumption (b) will be the closest to reality, however, in this case of the ξ-vicinity of the point w_i, we know information about the point w_i itself only. From the previous statements we can draw the following conclusion: each point (object) w_i belongs to a certain class H_j together with the ξ-vicinity.

The initial stage of the hypersphere algorithm deals with processing the data of the training selection in order to calculate the center of mass of the class H_0. When processing the data of the training selection of the form (1), for each point (vector) w_i, for which the relation $f_R(w_i) = 0$ is valid (it is assumed that the object belongs to the neighboring class H_0), the distance to the neighboring class H_1 is calculated in parallel, being determined by the ratio:

$$d_{\min}(w_i) = \min_{w_j \in H_1} (w_i - w_j). \quad (8)$$

The quantity $d_{\min}(w_i)$ is used to determine the radius of a hyper sphere centered at the point w_i. The radius $r(w_i)$ will be equal to $d_{\min}(w_i)/2$ and results from the following considerations: dividing the distance to the neighboring class in half is necessary to ensure that the hyperspheres with centers at the points $w_i (w_i \in H_0)$ and $w_j (w_j \in H_1)$ do not intersect. This, in turn, may lead to uncertainty in the classification process.

After actual determining the radii of the hyperspheres describing the class H_0, one may construct a certain manifold K_0 and write to it all the hyperspheres of the class H_0. Note that their number will be equal to that of objects in the training selection for which equality $f_R(w_i) = 0, (i = 1,...,m)$ will hold true. Note that we have fulfilled the first task (i.e. description of the class H_0 region with the help of hyperspheres), although it is clear that this approximation will not be optimal.

This stage of the algorithm operation relates to finding the most efficient and cost-effective description of the hyperspheres of the class H_0 region. To do this, for each of the elements of the manifold K_0 (hyperspheres w_i with radius $d_{\min}(w_i)/2$) we calculate the quantity $m(w_i)$ equal to the number of all w_j for which $f_R(w_i) = 0, (i = 1,...,m)$ and the following relation holds true:

$$r(w_i) < d(w_i, w_j). \quad (9)$$

Here $d(w_i, w_j)$ determines the distance in space between the objects w_i and w_j. Therefore, it is clear that the quantity $m(w_i)$ characterizes the number of points in the space of the class H_0, which will be described by a hypersphere centered at w_i. Let's call this quantity the hypersphere capacity. After calculating the hypersphere capacities from the manifold K_0, one has to select (and remove) that with the largest capacity, i.e. k_1, and place it into the manifold U, which will contain the hyperspheres of the optimal H_0 class approximation. The manifold K_0 is also subject to additional processing, i.e. removing from it all the elements (hyperspheres), the centers of which fall under the hypersphere k_1 region, i.e. meet the condition $r(k_1) < d(w_i, w_j)$.

After constructing the first element of the sequence of approximating hyperspheres of the manifold U, this construction scheme must be repeated, returning to the second stage. The process will continue until the desired sequence of hyperspheres is constructed that fully describes the class H_0. Note that even in the worst case of the location of the classes H_0, H_1 in the space the generalized attribute, nevertheless, will be constructed, and only the number of hyperspheres will be equal to the number of objects w_i from the training selection that belong to H_0. After describing the region of the class H_0 (hyperspheres belong to the manifold $U(k_1, k_2, ...)$), a sequence of approximating hyperspheres for the next class H_1 is constructed according to the similar scheme (result is $Y(l_1, l_2, ...)$).

The process of recognizing new objects w_i that enter the input of the classification system will proceed via the following scheme: first the hyperspheres of the manifold U will be checked, i.e., if $|k_j - w_i| \le r(k_j)$, the object w_i belongs to the class H_0. Then the hyperspheres of the manifold Y will be checked, and if $|l_j - w_i| \le r(l_j)$, then the object w_i belongs to the class H_1. However, there is a third option, when for a certain w_i none of the previous conditions will be met, then this object will be denied classification. The reason is not getting into any of the

DOI 10.15588/1607-3274-2021-3-10

class H_0, H_1 regions. The restriction we imposed at the beginning of the description of the hypersphere algorithm scheme (i.e. that the training selection specifies partitioning into two classes H_0, H_1) is not mandatory. An arbitrary selection of the objects w_i that define partitioning into an arbitrary number of classes H_i can be reduced to a training selection of two classes, for example, partitioning into H_0 and into all other classes, i.e. $H_1, H_2, ..., H_i$. After finding the approximating sequence for H_0, the training selection is again partitioned into H_1 and $H_2, H_3, ..., H_i$, and so on.

The main disadvantage of the above scheme of approximation by the sampling hyperspheres is the significant amount of calculations that appear with a dramatic increase of the training selection volume. First of all, this is due again to the constant cyclic processing of the selection to build each subsequent hypersphere, that is, if the desired sequence of hyperspheres consists of ten elements, the selection will be processed the same number of times. There is only one way to overcome this problem, i.e. to find a sequence of hyperspheres that approximates the working class with a less number of elements. This can be achieved after making structural changes to the scheme of the hypersphere algorithm.

To do this, at the very beginning of the algorithm operation the center of mass of the class c_0 according to scheme (3) and the value $d_{\min}(c_0)$ of the distance to the neighboring class (8) are calculated for the class H_0. After performing the above operations, we obtain the first desired hypersphere with a center at the point c_0 of the radius $r(c_0) = d_{\min}(c_0)/2$. Of course, having calculated its capacity and corrected the training selection, one can see that the selection size is significantly reduced, i.e. most of working class elements will be excluded from it. Next, the scheme of operation of the algorithm for constructing a sequence of hyperspheres will coincide with that previously described, but the number of steps will be significantly reduced due to the first, successfully constructed, element of the sequence sought.

Consider the local algorithm for generating attributes in the form of hyperellipses. In the previous local algorithm, some hypersphere was chosen at each step to form attributes, but it is clear that other spatial geometric objects, such as hyperellipses or hyperparallelepipeds, can also be used as attributes (see discussion below). Let us describe the essence of the algorithmic implementation of training selection approximation by hyperellipse, which is a further development of the concept of space approximation by hyperspheres and in some cases allows a simpler and more cost-saving description of the class region to be achieved.

Let us assume again that a known classification training selection with the objects w_i of type (1) is set in the n-dimensional space. To simplify the algorithm step explanation, we assume that $k = \{0,1\}$, i.e. the objects may belong to the two classes, i.e. H_0 or H_1 only. In addition, we shall assume that a certain metrics (2) is set in this space.

The initial stage will be the primary processing of the training sample and the construction of two manifolds G_0 and G_1, which will contain the objects of selection of the corresponding classes H_0, H_1, i.e. $w_j \in G_i$, if $w_j \in H_i$.

Such distribution of the training selection objects is necessary for the further construction of hyperellipses and calculation of the minimum distance to the next class. Having fixed, for example, a manifold H_0, we shall calculate for each object w_i the following number $d_{\min}$ that belongs to it:

$$d_{\min}(w_i) = \min_{\substack{w_0 \in H_0 \\ w_1 \in H_1}} d(w_0, w_1). \qquad (10)$$

Here $d(w_0, w_1)$ is a distance between the points w_0 and w_1. In fact, the quantity $d_{\min}(w_i)$ shall fix for each object w_i with H_0 a minimal distance to the neighboring class H_1 (i.e. to the nearest object of this class).

Recall that, by definition, an ellipse is the geometric location of points, the sum of the distances from which to two fixed points (foci) is a constant value. It is clear that it is natural to take pairs of objects (vectors, points) from the set H_0 as the above foci.

After calculating the measure of distance between each of the objects of the class H_0 and the neighboring class H_1, one has to sort and select all the pairs of points (w_i, w_j) from this class, for which the following inequality will be valid:

$$d(w_i, a) + d(w_j, a) < d_0(w_i, w_j). \qquad (11)$$

Here $(a \in H_1)$, and $d_0(w_i, w_j) = \min_{a \in H_1}((w_i, w_j), a)$. In fact, this inequality expresses the condition of the ellipse. The quantity d_0 characterizes the minimum distance to the next class. All the pairs of points (w_i, w_j) found will be selected in a special manifold E, having additionally introduced the quantity $r(w_i, w_j)$, i.e. the radius of the hyperellipse, which will be equal to $d_0(w_i, w_j)/2$. Its division in half is carried out for the reasons similar to the reasoning of the division of the hypersphere radius in the previous algorithm. Note only that all pairs of points (w_i, w_j) found must satisfy the condition of the hyperellipse (11), in the right part of which should be the value of its radius.

After performing the second stage, we obtain a manifold E that will contain all possible hyperellipses

DOI 10.15588/1607-3274-2021-3-10

actually describing the class H_0. An important point here is that none of them will contain points (objects) of another class H_1, but this approximation will not be complete. That is, in contrast to the hypersphere algorithm, in our case there may be points (objects) that will not lie in the region of any of the hyperellipses of the manifold E. It is clear that to a large extent this will be the result of the complex arrangement of classes (images) in space.

After the previous actions, we obtain an approximation by the sequence of hyperellipses of the manifold E rom a given training selection. After that, one has only to select the most effective description of the class, i.e. to find among the elements of the manifold E those that would provide the most optimal description of the class H_0. To ensure this condition fulfillment, it is necessary to calculate for each hyperellipse its capacity p_i. This value for each hyperellipse will show the number of objects with H_0, the distance to which is less than or equal to its radius.

It is clear that, knowing the capacities of each of the hyperellipse of the manifold E, we can already select that one for which this value will be the largest, and therefore, carry out a partial approximation. Note only that, in the terminology of the logical tree methods, hyperellipse will already be some generalized attribute, but covering (unambiguously classifying) a certain part of the class H_0 only. That is, the process of construction (selection) of further hyperellipses must be continued. To do this, one has to remove from the class H_0 those objects w_i that fall within the classification region of the first generalized attribute found (it is clear that their number is equal to the hyperellipse capacity). After that, one may proceed to the next step, i.e. to select the next best hyperellipse (generalized attribute), for which it is necessary to return to the first stage of the algorithm.

This sequence of actions will be carried out until such a sequence of hyperellipses is generated that completely approximates the class H_0 (the number of steps of the algorithm will coincide with their number). However, a second option is possible, when such elements will belong to the class H_0 from which it is impossible to construct such hyperellipses that will not cover the objects of the neighboring class H_1. The appearance of such objects is explained by the complex arrangement of classes (images) in space; and will result in classification failures that require some other recognition algorithm to recognize.

It is clear that after finding the sequence E of hyperellipses that will cover the class H_0, the hyperellipses are constructed in a similar manner to approximate all other classes of training selection (H_1 in our case). The process of recognizing new objects in the hyperellipse algorithm corresponds by its structure to the same stage in the previous hypersphere algorithm. That is, each w_i of the unknown classification that enters the input of the recognition system should be checked sequentially for compliance with the conditions of hyperellipse. The condition of the test will be to verify the fact that the distance between w_i and the centers of the hyperellipse should not exceed its radius. If w_i is absorbed by one of the hyperellipses, it will be assigned to a class that approximates this hyperellipse. If none of the conditions are met, an object of unknown classification (failure) will be generated for the object w_i.

The weakest point in such an algorithm scheme in terms of time consumption is the construction of all possible hyperellipses, which will simply be reduced to the usual search. It is clear that in many cases a significant part of them will be rejected at the stage of verification either due to overlapping of the class area (incorrectness) or due to relatively insignificant capacity values (duplication of hyperellipses). A cardinal solution of this problem is to find and select the so-called “boundary” points and build a manifold E of hyperellipses based on them. Note only that the scheme of this process will be considered in the following geometric algorithm of hyperparallelepipeds, and can proceed in several stages.

It is clear that the relatively high cost of computing capacity with constant each-step (cyclic) processing of the training selection is also an important disadvantage of such an algorithm scheme for constructing hyperellipses. One of the possible solutions of this problem may be to work only with those hyperellipses that are obtained at the first step of the algorithm. The downside in this case is that there is a high probability of non-optimal or incomplete approximation of the training selection, first of all, this is due to the fact that the elements of the manifold E do not vary with each step. It should be emphasized that in the software implementation of real recognition problems, it is important to find the middle ground between the volume of calculations to elaborate the resulting classification rule. That is, the choice of, for example, the scheme of initial processing of the training selection should depend, first of all, on its volume.

Also note that since the hypercsphere algorithm is a partial case of the hyperellipse algorithm, that is, if the hyperellips turns into a hypersphere, then the number of hyperellipses will not exceed that of hyperspheres, which approximate this class based on the hyperphere algorithm. It does not follow from this that such an algorithm scheme will always lead us to the desired coverage of the class region (at least, by hyperspheres).

4 EXPERIMENTS

Note that the tree algorithm method uses one of the four simple geometric recognition algorithms as the attributes, i.e. the hypersphere algorithm, the hyperplane algorithm, the hyperellipse algorithm and the hyperparallelepiped algorithm. Their operation principle is to approximate the training selection by appropriate geometric objects [11]. The result of each of these algorithms is one or more generalized attribute(s)

DOI 10.15588/1607-3274-2021-3-10

(corresponding geometric objects), which describe a certain part of the training selection. Moreover, there may be the cases when the algorithm fails to construct a generalized attribute due to the complex arrangement of classes in the *n*-dimensional space. It is also possible that the constructed attributes do not approximate fully the selection (objects that do not fall under this approximated region are called classification failures).

Let a training selection of 2.000 elements be the only information about the nature of the manifold *M* partition into the classes H_i. Note that here we deal with the objects w_i described by three attributes and grouped into the four classes. So, the problem is to distribute objects of unknown classification over one of four classes. The presented selection contains the data of chemical analysis of the diesel fuel content (the task of assessing the quality of fuel) in a simplified version (the number of attributes is reduced from fifteen to three) to demonstrate the very concept of the algorithmic tree. Note that in this case there is no test sample, i.e. we will not assess the efficiency of the constructed system, but will only approximate the data with a set of generalized attributes (algorithms).

At the first stage, we shall assess the efficiency of each of the algorithms, on the basis of which the general classification scheme will be built, with respect to the initial training selection (Table 1).

Table 1 – Assessment of the efficiency of elementary geometric algorithms for classifying discrete objects with respect to the initial selection

Class number Algorithm type	Class 1	Class 2	Class 3	Class 4
Hypersphere algorithm	0/32	0/16	0/18	0/11
Hyperellipse algorithm	0/12	2/4	0/10	15/3
Hyperparallelepi-ped algorithm	0/6	1/5	0/7	9/6
Hyperplane algorithm	21/9	14/6	0/2	12/6

The cells present the efficiency of each of the algorithms with respect to the classes of the initial training selection. The first number is responsible for the quantity of objects that are denied in classification by the appropriate algorithm, and the second one is responsible for that of generalized attributes (geometric objects), which approximate the corresponding selection class. Depending on the initial choice of the algorithm as the recognition tree vertex, the process of constructing the resulting classification scheme can be completed with a different number of steps. Possible classification scheme is presented in (Fig. 1).

It can be seen from Table 1 that the efficiency of all the algorithms, except for the hyperplane one, with respect to the *Class 1* is 100%, so it can be applied to any algorithm (of course, except for the hyperplane algorithm). At any further stages of recognition scheme construction, this algorithm is selected again, and it has been proven to be the most efficient and economical one with respect to all other classes of the initial selection. In addition, each of the generalized attributes generated by it represents the coordinates of the center of the hypersphere and its radius and requires a minimum amount of memory for its storage.

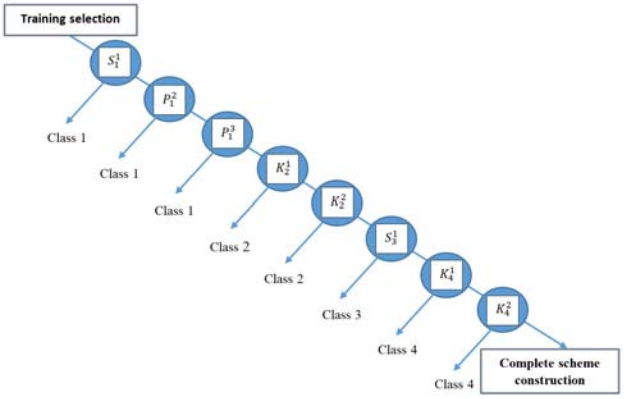

Figure 1 – Example of RS constructed

The recognition scheme (Fig. 1) is built of algorithms, the efficiency of which was evaluated with respect to the number of generalized attributes, which they use to describe the training selection. Thus, two algorithms were used to approximate the *Class 1*: first, the hyperplane algorithm had constructed the GA S_1^1, which described it only partially. At the second stage, the hyperparallelepiped algorithm is applied, i.e. the attributes P_1^2 and P_1^3 that finally completed the recognition of this class. At the following stages of recognition the hypersphere (attributes $K_2^1, K_2^2, K_4^1, K_4^2$) and the hyperplane algorithms (attribute S_3^1) are again applied. Note that in order to build this scheme three different recognition algorithms are used not directly affecting each other's operation. That is, completely different in principle and ideology algorithms could be applied instead of them, allowing one to construct a recognition scheme with arbitrary complexity and efficiency. Only the efficiency of each of them for a fixed selection and the information capacity of the generalized attributes generated by them are important. That is, the tree method operates with ready-made (constructed) attributes only, and it may not be interested in what algorithm or method they were obtained.

This recognition scheme is constructed on the basis of a logical tree method and can be represented as a certain algorithmic scheme (operator) built by some algorithm to minimize or maximize the corresponding functionality, based on which the importance of the attribute, group of attributes or the efficiency of the recognition algorithm uniquely related to classification errors are assessed (it crosses the method of attribute branched selection).

Note that the tree method based on the input (training selection) data and the range of algorithms for generalized attributes stored in its library constructs (generates) a

DOI 10.15588/1607-3274-2021-3-10

certain scheme optimal by memory costs (complexity) and recognition efficiency (system). Under the scheme in this case we mean a set of numerical parameters for elementary attributes that best approximate the initial data array [14] (meaning the definition of the decisive scheme and the I. Vitenko's generalized attribute). Thus, in our case, the arguments of the constructed recognition scheme are the class attributes (hypercoils, hyperellips, etc.) or interclass attributes (hyperplanes). The parameters of the specified attributes and the general structure of the system (scheme) are stored in the computer memory.

Each of the schemes constructed by the tree method will be a general recognition system that can be used for practical work (processing large arrays of experimental data in the form of data arrays). Note also that the resulting scheme will be to some extent a new recognition algorithm (of course, synthesized from known algorithms and methods). In addition, for these classification systems it is not necessary to store in the computer memory the objects of selection on which it was constructed, i.e. large information arrays. The latter, in turn, leads to the fact that the process of constructing recognition system based on the tree method is largely similar to the process of information compression (meaning methods of information compression with losses) or encoding.

5 RESULTS

Thus, based on the classification tree method and modularity principle, Uzhhorod National University has developed a software package Orion III to generate autonomous recognition systems. The algorithmic library of this system has 11 recognition algorithms, among which are the geometric algorithmic implementations suggested above. This system allows other autonomous algorithms to be connected given the provision of the data exchange interface with the module for generating recognition schemes (the open architecture principle). Due to the use of external algorithms, which can be based on arbitrary concepts, a high versatility of the software for a wide range of recognition tasks is provided. Note also that the system allows generation of autonomous recognition systems in two modes – the automatic (with a step-by-step assessment of the efficiency of a set of algorithms with respect to the training selection) and interactive (choice of vertices and algorithms of the algorithmic tree depend on the operator) ones. This approach provides to a great extent high versatility in solving application tasks (it shifts responsibility to the implementation of an autonomous recognition algorithm), requires less attention to the task specifics (interpretation of attributes), and, on the other hand, imposes high requirements on the completeness and adequacy (quality) of training selections.

The principal task here was to construct an autonomous recognition system based on geological data (the problem of oil-bearing bed separation). The mathematical model of recognition objects in this case is presented in the form of attributes of the $x_1, x_2, ..., x_n$ - sets. Their following main properties were used to recognize objects (12 basic elementary attributes and 10 additional ones):

– bed thickness;
– clay solution resistance;
– resistance ρ_k on the standard potential probe;
– resistance on the gradient probes at the lateral logging sounding with the $A_0 = 0.5$ m size;
– resistance on the gradient probes at the lateral logging sounding with the $A_0 = 1$ m size;
– resistance on the gradient probes at the lateral logging sounding with the $A_0 = 2$ m size;
– resistance on the gradient probes at the lateral logging sounding with the $A_0 = 4$ m size;
– resistance on the gradient probes at the lateral logging sounding with the $A_0 = 8$ m size;
– inverse probe;
– rock resistance;
– well diameter;
– clay cake thickness.

The training sample provides information about the objects of two classes (the oil-bearing bed class and the water-bearing bed class). At the stage of examination, the constructed classification system should ensure effective recognition of objects of unknown classification with respect to these two classes. Before starting work, the training selection was automatically checked for correctness (searching and deleting the same objects of different belonging, i.e. the first-kind errors), although the system has implemented a scheme for additional training and error correction in the classification tree (the ATEC algorithm), since generation took place automatically, then this algorithm was not used.

Note that the training selection consisted of 1.250 objects (of which 756 were the oil-bearing ones), and the efficiency of the constructed recognition system was assessed on a test sample of 240 objects. The data from training and test selections were obtained on the basis of geological exploration in the Transcarpathian region during the period from 2001 to 2013. The methods of training selection approximation on the basis of hyperspheres and hyperellipses were used as the fixed algorithms (only geometric algorithms were selected from the library, and their algorithmic schemes were described above). The test use of a set of other library algorithms provided generation of an algorithmic tree of much greater complexity. It should also be noted that increasing of the set of algorithms negatively affects the total generation time of the system (in automatic mode) due to their step-by-step assessment with respect to the training selection.

Also, the resulting number of generalized features was 18 per 756 objects of the oil-bearing bed class and 22 per 494 objects of the water-bearing bed class. The constructed autonomous classification system is based on the recognition scheme presented in Fig. 2.

DOI 10.15588/1607-3274-2021-3-10

Note that if one assesses the efficiency (with respect to compression – the description of the training selection data) of the constructed scheme of the system of discrete object classification by the formula $\rho = 100 - (100 \cdot \frac{q}{m})$ and amounted to 96.8%.

Note that this recognition system was constructed for two different configurations:

1) (*Conf No. 1* Intel I5 8500 / Ram 8GB);

2) (*Conf No. 2* AMD FX8370 / Ram 16GB).

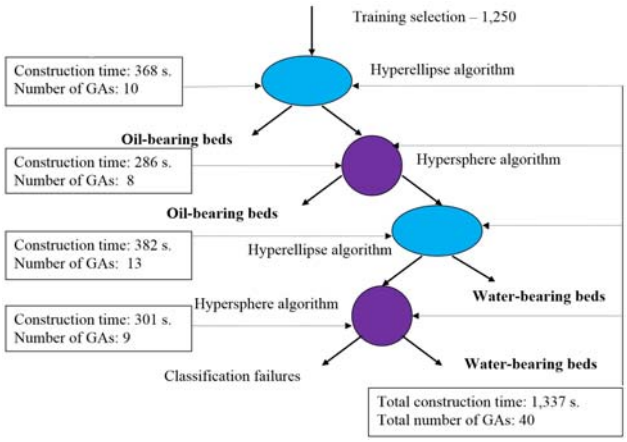

Figure 2 – The layout of the RS constructed on the basis of geological data

The whole process took 1.586 seconds and 1.337 seconds, respectively, what is largely due to the increase in the processor frequencies, disk subsystem speed and assembler optimization.

Note also that the hyperplane and hyperparallelepiped algorithms appeared not very suitable for this recognition system, since they 'refused' to work on this array of geological data (training selection). It should also be noted that the hyperellipse algorithm 'failed' to perform a complete approximation of the class regions, and was used in conjunction with another algorithm to simplify the resulting recognition scheme. At the stage of evaluating the effectiveness of the system constructed, 221 of 240 objects of the test sample were successfully classified (5 objects were denied classification). This pretty good result (certainly, 1250 objects are not enough to build a normal highly efficient classification system volume) is mainly due to the use of the hypersphere algorithm to approximate the training selection. This allowed one to ensure effective separation of the class region even at an incomplete (inadequate) selection due to increasing the number of generalized features.

6 DISCUSSION

Note that the structure of the algorithm tree constructed within this work (the ACT model) operates only with ready-made (constructed) sets of generalized attributes (elementary geometric classification algorithms), and it may not be interested in any general algorithm or method (scheme, rule, method) they are obtained, and each of the schemes constructed by the algorithmic tree method will be a general recognition system that can be used for practical work (processing large arrays of experimental data in the form of discrete sets of arbitrary nature). An important point is that the resulting classification scheme (tree of algorithms) will be to some extent a new recognition algorithm (of course, synthesized from known algorithms and methods), and the resulting ACT structure (new classification scheme) is characterized by high versatility with respect to the application and relatively compact structure of the model itself (within the scope of the problem presented in the work, the GA sets of only two types were used). However, the ACT structure requires relatively high hardware costs for storing the generalized attributes (or their sets) and the initial assessment of the quality of classification algorithms according to TS. Moreover, the ACT models, in comparison with the LCT structures, have a high-speed classification rules, comparable hardware costs for storage and operation of the tree structure and high quality of classification.

CONCLUSIONS

New simple algorithmic implementations for approximating an array of geological data by a set of generalized features (elementary geometric algorithms of algorithms) have been suggested in this paper. The ACT structure is a graph-schematic structure with a dimension of 40 generalized attributes and generation time of 1.337 seconds, and for the tree synthesis two elementary geometric algorithms have been used.

An approach to the synthesis of new recognition algorithms based on a library (set) of already known algorithms and methods has been developed. That is, an effective scheme for recognizing discrete objects based on the tree method is presented with a step-by-step assessment and selection of generalized features at each step of the scheme synthesis. The efficiency of the developed discrete object classification scheme (geological data) was 96%.

The **scientific novelty** of the obtained results is based on the fact that the proposed method of the ACT structures on the basis of a set of autonomous recognition and classification algorithms assessment and ranking for generating the classification tree structure (the ACT model). Moreover, at each step of the classification tree branching a certain part of the TS (or its submanifold) is approximated.

The **practical value** of the obtained results is that the method of constructing the ACT models (the LCT/ACT structure) was implemented in the library of algorithms of the universal software system ORION III to solve various practical classification (recognition) problems for different arrays of discrete objects. The practical tests confirmed the efficiency of mathematical software, proposed ACT models and developed software that allows one to make recommendations on the use of this approach (the algorithm tree models) and its software implementation

DOI 10.15588/1607-3274-2021-3-10

for a wide range of applications for discrete object classification and recognition.

Prospects for further research may be directed towards the development of algorithmic classification trees methods, i.e. the methods of cutting-off and minimizing the ACT structures, optimizing software implementations of the ACT construction method, as well as its practical testing on the manifold of real classification and recognition problems.

ACKNOWLEDGEMENTS

The work was carried out within the framework of the scientific topic of the Department of software systems of the state higher educational institution "Uzhgorod National University" – methods and tools of software engineering, implementation of large data analysis processes based on information platforms (State Registration Number 0119U100703).

REFERENCES

1. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning. Stanford, 2008, 768 p.
2. Quinlan J. R. Induction of Decision Trees, *Machine Learning*, 1986, No. 1, pp. 81–106.
3. Mitchell T. Machine learning. New York, McGraw-Hill, 1997, 432 p.
4. Dietterich T. G., Kong E. B. Machine learning bias, statistical bias, and statistical variance of decision tree algorithms [Electronic resource]. Corvallis, Oregon State University, 1995, 14 p. Access mode : http://www.cems.uwe.ac.uk/~irjohnso/coursenotes/uqc832/tr bias.pdf
5. Breiman L. L., Friedman J. H., Olshen R. A., Stone C. J. Classification and regression trees. Boca Raton, Chapman and Hall/CRC, 1984, 368 p.
6. Vtogoff P.E. Incremental Induction of Decision Trees, *Machine Learning*, 1989, No. 4, pp. 161−186.
7. Vasilenko Y. A., Vasilenko E. Y., Kuhayivsky A. I., Papp I. O. Construction and optimization of recongnizing systems, *Scientific and technical journal "Information technologies and systems",* 1999, No. 1, pp. 122–125.
8. Subbotin S.A. Construction of decision trees for the case of low-information features, *Radio Electronics, Computer Science, Control*, 2019, No. 1, pp. 121–130.
9. Povhan I.F. Logical recognition tree construction on the basis a step-to-step elementary attribute selection, *Radio Electronics, Computer Science, Control*, 2020, No. 2, pp. 95–106.
10. Povkhan I. F. The general concept of the methods of algorithmic classification trees, *Radio Electronics, Computer Science, Control,* 2020, No. 3, pp. 108–121.
11. Povhan I. F. Limited method for the case of algorithmic classification tree, *Radio Electronics, Computer Science, Control,* 2020, No. 4, pp. 106–118.
12. Povhan I. Designing of recognition system of discrete objects, *2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP), 2016, Lviv, Ukraine.* Lviv, 2016, pp. 226–231.
13. Vasilenko Y. A., Vashuk F. G., Povkhan I. F. Automating the construction of classification systems based on agent – schemes, *Mathematical modeling, optimization and information technologies : International Joint Conference MDIF-2012, Kisheneu, Moldova, 2012*. Kisheneu, 2012, pp. 444–446.
14. Povkhan I.F. Features of synthesis of generalized features in the construction of recognition systems using the logical tree method, *Information technologies and computer modeling ITKM-2019 : materials of the international scientific and practical conference, Ivano-Frankivsk, May 20–25, 2019*. Ivano-Frankivsk, 2019, pp. 169–174.
15. Vasilenko Y. A., Vashuk F. G., Povkhan I. F. The importance of discrete signs, *XX International Conference Promising ways and directions of improving the educational system, Uzhgorod, November 16–19, 2010.* Uzhgorod, 2010, Vol. 21, No. 1, pp. 217–222.
16. Alpaydin E. Introduction to Machine Learning. London, The MIT Press. 2010, 400 p.
17. De Mántaras R. L. A distance-based attribute selection measure for decision tree induction, *Machine learning*, 1991, Vol. 6, No. 1, pp. 81–92.
18. Painsky A., Rosset S. Cross-validated variable selection in tree-based methods improves predictive performance, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, Vol. 39, No. 11, pp. 2142–2153. DOI:10.1109/tpami.2016.2636831.
19. Miyakawa M. Criteria for selecting a variable in the construction of efficient decision trees, *IEEE Transactions on Computers*, 1989, Vol. 38, No. 1, pp. 130–141.
20. Kotsiantis S.B. Supervised Machine Learning: A Review of Classification Techniques, *Informatica*, 2007, No. 31, pp. 249–268.
21. Deng H., Runger G., Tuv E. Bias of importance measures for multi-valued attributes and solutions, *Proceedings of the 21st International Conference on Artificial Neural Networks (ICANN), Espoo, Finland, Jun 14–Jun 17, 2011*. Espoo, 2011, pp. 293–300.
22. Dietterich T. G. An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization, *Machine learning*, 2000, Vol. 40, No. 2, pp. 139–157.
23. Subbotin S., Oliinyk A. *eds. R. Szewczyk, M. Kaliczyńska*.The dimensionality reduction methods based on computational intelligence in problems of object classification and diagnosis, *Recent Advances in Systems, Control and Information Technology.* Cham, Springer, 2017, pp. 11–19. (Advances in Intelligent Systems and Computing, vol. 543).
24. Subbotin S. A. Methods and characteristics of localitypreserving transformations in the problems of computational intelligence, *Radio Electronics, Computer Science, Control,* 2014, No. 1, pp. 120–128.
25. Subbotin S.A. Methods of sampling based on exhaustive and evolutionary search, *Automatic Control and Computer Sciences,* 2013, Vol. 47, No. 3, pp. 113–121. DOI: 10.3103/s0146411613030073
26. Koskimaki H., Juutilainen I., Laurinen P., Roning J. Two-level clustering approach to training data instance selection: a case study for the steel industry, *Neural Networks : International Joint Conference (IJCNN-2008), Hong Kong, 1–8 June 2008, proceedings*. Los Alamitos, IEEE, 2008, pp. 3044–3049. DOI: 10.1109/ijcnn.2008.4634228
27. Srikant R., Agrawal R. Mining generalized association rules *Future Generation Computer Systems*, 1997, Vol. 13, No. 2, pp. 161–180.
28. Amit Y., Geman D., Wilder K. Joint induction of shape features and tree classifiers, *IEEE Transactions on Pattern*

DOI 10.15588/1607-3274-2021-3-10

Analysis and Machine Intelligence, 1997, Vol. 19, No. 11, pp. 1300–1305.
29. Mingers J. An empirical comparison of pruning methods for decision tree induction, *Machine learning*, 1989, Vol. 4, No. 2, pp. 227–243.
30. Karimi K., Hamilton H. J. Generation and Interpretation of Temporal Decision Rules, *International Journal of Computer Information Systems and Industrial Management Applications*, 2011, Vol. 3, pp. 314 –323.
31. Kamiński B., Jakubczyk M., Szufel P. A framework for sensitivity analysis of decision trees, *Central European Journal of Operations Research*, 2017, Vol. 26 (1), pp. 135–159.
32. Lupei M., Mitsa A., Repariuk V., Sharkan V. Identification of authorship of Ukrainian-language texts of journalistic style using neural networks, *Eastern-European Journal of Enterprise Technologies*, 2020, Vol. 1 (2 (103)), pp. 30–36. DOI: https://doi.org/10.15587/1729-4061.2020.195041
33. Bodyanskiy Y., Vynokurova O., Setlak G. and Pliss I. Hybrid neuro-neo-fuzzy system and its adaptive learning algorithm, *Computer Sciences and Information Technologies (CSIT), Xth Scien. and Tech. Conf., Lviv, 2015*. Lviv, 2015, pp. 111–114.
34. Subbotin S. The neuro-fuzzy network synthesis and simplification on precedents in problems of diagnosis and pattern recognition, *Optical Memory and Neural Networks (Information Optics)*, 2013, Vol. 22, No. 2, pp. 97–103. DOI: 10.3103/s1060992x13020082

Received 15.06.2021.
Accepted 13.08.2021.

УДК 001.891:65.011.56

ЗАДАЧА АПРОКСИМАЦІЇ МАСИВУ ДИСКРЕТНИХ ДАНИХ НАБОРОМ ЕЛЕМЕНТАРНИХ ГЕОМЕТРИЧНИХ АЛГОРИТМІВ

Повхан І. Ф. – д-р техн. наук, доцент, доцент кафедри програмного забезпечення систем ДВНЗ Ужгородський національний університет, м. Ужгород, Україна.

Міца О. В. – д-р техн. наук, доцент, зав. кафедри інформаційних управляючих систем і технологій. ДВНЗ Ужгородський національний університет, м. Ужгород, Україна.

Мулеса О. Ю. – д-р техн. наук, доцент, доцент кафедри кібернетики та прикладної математики ДВНЗ Ужгородський національний університет, м. Ужгород, Україна.

Мельник О. О. – канд. техн. наук, доцент, доцент кафедри програмного забезпечення систем ДВНЗ Ужгородський національний університет, м. Ужгород, Україна.

АНОТАЦІЯ

Актуальність. В роботі розв’язана задача апроксимації масиву дискретних даних набором елементарних геометричних алгоритмів і представлення побудованої моделі розпізнавання у вигляді алгоритмічного дерева класифікації. Об‘єктом даного дослідження є концепція дерева класифікації у вигляді дерева алгоритмів. Предметом дослідження є актуальні моделі, методи, алгоритми та схеми побудови різнотипних дерев класифікації.

Мета. Метою даної роботи є створення простого та ефективного методу та алгоритмічної схеми побудови деревоподібних моделей розпізнавання та класифікації на основі дерев алгоритмів для навчальних вибірок дискретної інформації великого об‘єму, який характеризується модульною структурою з незалежних алгоритмів розпізнавання оцінених на основі даних початкової начальної вибірки для широкого спектру прикладних задач.

Метод. Пропонується схема синтезу дерев класифікації (дерев алгоритмів) на основі апроксимації масиву даних набором елементарних геометричних алгоритмів, яка для заданої початкової навчальної вибірки довільного розміру будує деревоподібну структуру (модель АДК), яка складається з набору автономних алгоритмів класифікації та розпізнавання оцінених на кожному кроці, етапі побудови АДК за даною початковою вибіркою. Розроблений метод побудови алгоритмічного дерева класифікації основна ідея якого полягає в по кроковій апроксимації начальної вибірки довільного об‘єму та структури набором елементарних геометричних алгоритмів класифікації. Даний метод при формуванні поточної вершини дерева алгоритмів, вузла, узагальненої ознаки, забезпечує виділення найбільш ефективних, якісних елементарних алгоритмів класифікації з початкового набору та добудову лише тих шляхів в структурі АДК де відбувається найбільша кількість помилок класифікації. Розроблена схема синтезу результуючого дерева класифікації, моделі АДК дозволяє значно скоротити розмір та складність дерева. Структурна складність конструкції АДК оцінюється на основі кількості переходів, вершин та ярусів структури АДК, що дозволяє підвищити якість його наступного аналізу, забезпечити ефективний механізм декомпозиції, та будувати структури АДК в умовах фіксованих наборів обмежень. Метод синтезу дерев алгоритмів дозволяє будувати різнотипні деревоподібні моделі розпізнавання з різними початковими наборами елементарних класифікаторів з наперед заданою точністю для широкого класу задач теорії штучного інтелекту.

Результати. Розроблений та представлений в даній роботі метод апроксимації дискретних навчальних вибірок набором елементарних геометричних алгоритмів отримав програмну реалізацію та був досліджений і порівняний з методами логічних дерев класифікації на основі селекції елементарних ознак при розв’язку задачі розпізнавання реальних даних геологічного типу.

Висновки. Проведені в даній роботі загальний аналіз та експерименти підтвердили працездатність розробленого механізму побудови структур дерев алгоритмів та показують можливість його перспективного використання для розв‘язку широкого спектру практичних задач розпізнавання та класифікації. Перспективи подальших досліджень та апробацій можуть полягати в створенні методів алгоритмічного дерева класифікації інших типів з іншими початковими наборами елементарних класифікаторів, оптимізації його програмних реалізацій, а також експериментальних дослідженнях даного методу на більш широкому колі практичних задач.

DOI 10.15588/1607-3274-2021-3-10

КЛЮЧОВІ СЛОВА: алгоритмічне дерево класифікації, розпізнавання образів, класифікація, алгоритм класифікації, критерій розгалуження, геометричний алгоритм.

УДК 001.891:65.011.56

ЗАДАЧА АППРОКСИМАЦИИ МАССИВА ДИСКРЕТНЫХ ДАННЫХ НАБОРОМ ЭЛЕМЕНТАРНЫХ ГЕОМЕТРИЧЕСКИХ АЛГОРИТМОВ

Повхан И. Ф. – д-р техн. наук, доцент, доцент кафедры программного обеспечения систем ГВУЗ Ужгородский национальный университет, г. Ужгород, Украина.

Мица А. В. – д-р техн. наук, доцент, зав. кафедры информационных управляющих систем и технологий ГВУЗ Ужгородский национальный университет, г. Ужгород, Украина.

Мулеса О. Ю. – д-р техн. наук, доцент, доцент кафедры кибернетики и прикладной математики ГВУЗ Ужгородский национальный университет, г. Ужгород, Украина.

Мельник Е. А. – канд. техн. наук, доцент, доцент кафедры программного обеспечения систем ГВУЗ Ужгородский национальный университет, г. Ужгород, Украина.

АННОТАЦИЯ

Актуальность. В работе решена задача аппроксимации массива дискретных данных набором элементарных геометрических алгоритмов и представления построенной модели распознавания в виде алгоритмического дерева классификации. Объектом данного исследования является концепция деревьев классификации в виде дерева алгоритмов. Предметом исследования являются актуальные модели, методы, алгоритмы и схемы построения разнотипных деревьев классификации.

Цель. Целью данной работы является создание простого и эффективного метода и алгоритмической схемы построения древовидных моделей распознавания и классификации на основе деревьев алгоритмов для обучающих выборок дискретной информации большого объема. Причем они характеризуется модульной структурой из независимых алгоритмов распознавания, оцененных на основе данных начальной обучающей выборки для широкого спектра прикладных задач.

Метод. Предлагается схема синтеза деревьев классификации (деревьев алгоритмов) на основе аппроксимации массива данных набором элементарных геометрических алгоритмов, которая для заданной исходной обучающей выборки произвольного размера строит древовидную структура (модель АДК), которая состоит из набора автономных алгоритмов классификации и распознавания, оцененных на каждом этапе построения АДК по данной исходной выборке. Разработан метод построения алгоритмического дерева классификации основная идея которого заключается в по шаговой аппроксимации начальной выборки произвольного объема и структуры набором элементарных геометрических алгоритмов классификации. Данный метод при формировании текущей вершины дерева алгоритмов, узла, обобщенного признака, обеспечивает выделение наиболее эффективных, качественных элементарных алгоритмов классификации из начального набора и достройку только тех путей в структуре АДК где происходит наибольшее количество ошибок классификации. Разработана схема синтеза результирующего дерева классификации, модели АДК позволяет значительно сократить размер и сложность дерева. Структурная сложность конструкции АДК оценивается на основе количества переходов, вершин и ярусов структуры АДК, что позволяет повысить качество его последующего анализа, обеспечить эффективный механизм декомпозиции, и строить структуры АДК в условиях фиксированных наборов ограничений. Метод синтеза деревьев алгоритмов позволяет строить разнотипные древовидные модели распознавания с различными начальными наборами элементарных классификаторов с заранее заданной точностью для широкого класса задач теории искусственного интеллекта.

Результаты. Разработанный и представленный в данной работе метод аппроксимации дискретных обучающих выборок набором элементарных геометрических алгоритмов получил программную реализацию и был исследован и сравнен с методами логических деревьев классификации на основе селекции элементарных признаков при решении задачи распознавания реальных данных геологического типа.

Выводы. Проведенные в данной работе общий анализ и эксперименты подтвердили работоспособность разработанного механизма построения структур деревьев алгоритмов и показывают возможность его перспективного использования для решения широкого спектра практических задач распознавания и классификации. Перспективы дальнейших исследований и апробаций могут заключаться в построении методов алгоритмического дерева классификации других типов с другими начальными наборами элементарных классификаторов, оптимизации его программных реализаций, а также экспериментальных исследованиях данного метода в более широком спектре практических задач.

КЛЮЧЕВЫЕ СЛОВА: алгоритмическое дерево классификации, распознавания образов, классификация, алгоритм классификации, критерий ветвления, геометрический алгоритм.

ЛІТЕРАТУРА / ЛИТЕРАТУРА

1. Hastie T. The Elements of Statistical Learning / T. Hastie, R. Tibshirani, J. Friedman. – Stanford, 2008. – 768 p.
2. Quinlan J.R. Induction of Decision Trees / J. R. Quinlan // Machine Learning. – 1986. − №1. – P. 81–106.
3. Mitchell T. Machine learning / T. Mitchell. – New York : McGrawHill, 1997. – 432 p.
4. Dietterich T. G. Machine learning bias, statistical bias and statistical variance of decision tree algorithms [Electronic resource] / T. G. Dietterich, E. B. Kong. – Corvallis : Oregon State University, 1995. – 14 p. Access mode : http://www.cems.uwe.ac.uk/~irjohnso/coursenotes/uqc832/tr bias.pdf
5. Classification and regression trees / L. L. Breiman, J. H. Friedman, R. A. Olshen et al.]. – Boca Raton : Chapman and Hall/CRC, 1984. – 368 p.
6. Vtogoff P. E. Incremental Induction of Decision Trees / P. E. Vtogoff // Machine Learning. – 1989. – № 4. – P. 161−186.

DOI 10.15588/1607-3274-2021-3-10

7. Construction and optimization of recongnizing systems / [Y. A. Vasilenko, E. Y. Vasilenko, A. I. Kuhayivsky, I. O. Papp] // Scientific and technical journal "Information technologies and systems". – 1999. – № 1. – P. 122–125.
8. Subbotin S. A. Construction of decision trees for the case of low-information features / S. A. Subbotin // Radio Electronics, Computer Science, Control. – 2019. – № 1. – P. 121–130.
9. Povhan I. F. Logical recognition tree construction on the basis a step-to-step elementary attribute selection / I. F. Povhan // Radio Electronics, Computer Science, Control. – 2020. – № 2. – P. 95–106.
10. Povkhan I. F. The general concept of the methods of algorithmic classification trees / I. F. Povkhan // Radio Electronics, Computer Science, Control. – 2020. – № 3. – P. 108–121.
11. Povhan I. F. Limited method for the case of algorithmic classification tree / I. F. Povhan // Radio Electronics, Computer Science, Control. – 2020. – № 4. – P. 106–118.
12. Povhan I. Designing of recognition system of discrete objects / I. Povhan // 2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP), Lviv, 2016, Ukraine. – Lviv, 2016. – P. 226–231.
13. Vasilenko Y. A. Automating the construction of classification systems based on agent-schemes / Y. A. Vasilenko, F. G. Vashuk, I. F. Povkhan // Mathematical modeling, optimization and information technologies : International Joint Conference MDIF-2012, Kisheneu, Moldova, 2012. – Kisheneu, 2012. – P. 444–446.
14. Povkhan I. F. Features of synthesis of generalized features in the construction of recognition systems using the logical tree method / I. F. Povkhan // Information technologies and computer modeling ITKM-2019 : materials of the international scientific and practical conference, Ivano-Frankivsk, May 20–25, 2019. – Ivano-Frankivsk, 2019. – P. 169–174.
15. Vasilenko Y. A. The importance of discrete signs / Y. A. Vasilenko, F. G. Vashuk, I. F. Povkhan // XX International Conference Promising ways and directions of improving the educational system, Uzhgorod, November 16–19, 2010. – Uzhgorod, 2010. – Vol. 21, № 1. – P. 217–222.
16. Alpaydin E. Introduction to Machine Learning / E. Alpaydin. – London : The MIT Press, 2010. – 400 p.
17. De Mántaras R. L. A distance-based attribute selection measure for decision tree induction / De Mántaras R. L. // Machine learning. – 1991. – Vol. 6, № 1. – P. 81–92.
18. Painsky A. Cross-validated variable selection in tree-based methods improves predictive performance / A. Painsky, S. Rosset // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2017. – Vol. 39, № 11. – P. 2142–2153. DOI:10.1109/tpami.2016.2636831
19. Miyakawa M. Criteria for selecting a variable in the construction of efficient decision trees / M. Miyakawa // IEEE Transactions on Computers. – 1989. – Vol. 38, № 1. – P. 130–141.
20. Kotsiantis S. B. Supervised Machine Learning: A Review of Classification Techniques / S. B. Kotsiantis // Informatica. – 2007. – № 31. – P. 249–268.
21. Deng H. Bias of importance measures for multi-valued attributes and solutions / H. Deng, G. Runger, E. Tuv // Proceedings of the 21st International Conference on Artificial Neural Networks (ICANN), Espoo, Finland, Jun 14–Jun 17, 2011. – Espoo, 2011. – P. 293–300.
22. Dietterich T. G. An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization / T. G. Dietterich // Machine learning. – 2000. – Vol. 40, № 2. – P. 139–157.
23. Subbotin S. The dimensionality reduction methods based on computational intelligence in problems of object classification and diagnosis / S. Subbotin, A. Oliinyk // Recent Advances in Systems, Control and Information Technology / eds.: R. Szewczyk, M. Kaliczyńska. – Cham : Springer, 2017. – P. 11–19. – (Advances in Intelligent Systems and Computing, vol. 543).
24. Subbotin S. A. Methods and characteristics of locality-preserving transformations in the problems of computational intelligence / S. A. Subbotin // Radio Electronics, Computer Science, Control. – 2014. – № 1. – P. 120–128.
25. Subbotin S. A. Methods of sampling based on exhaustive and evolutionary search / S. A. Subbotin // Automatic Control and Computer Sciences. – 2013. – Vol. 47, № 3. – P. 113–121. DOI: 10.3103/s0146411613030073
26. Two-level clustering approach to training data instance selection: a case study for the steel industry / [H. Koskimaki, I. Juutilainen, P. Laurinen, J. Roning] // Neural Networks : International Joint Conference (IJCNN-2008), Hong Kong, 1–8 June 2008 : proceedings. – Los Alamitos : IEEE, 2008. – P. 3044–3049. DOI: 10.1109/ijcnn.2008.4634228
27. Srikant R. Mining generalized association rules / R. Srikant, R. Agrawal // Future Generation Computer Systems. – 1997. – Vol. 13, №2. – P. 161–180.
28. Amit Y. Joint induction of shape features and tree classifiers / Y. Amit, D. Geman, K. Wilder // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 1997. – Vol. 19, № 11. – P. 1300–1305.
29. Mingers J. An empirical comparison of pruning methods for decision tree induction / J. Mingers // Machine learning. – 1989. – Vol. 4, № 2. – P. 227–243.
30. Karimi K. Generation and Interpretation of Temporal Decision Rules / K. Karimi, H. J. Hamilton // International Journal of Computer Information Systems and Industrial Management Applications. – 2011. – Vol. 3. – P. 314–323.
31. Kamiński B. A framework for sensitivity analysis of decision trees / B. Kamiński, M. Jakubczyk, P. Szufel // Central European Journal of Operations Research. – 2017. – Vol. 26 (1). – P. 135–159.
32. Identification of authorship of Ukrainian-language texts of journalistic style using neural networks / [M. Lupei, A. Mitsa, V. Repariuk, V. Sharkan] // Eastern-European Journal of Enterprise Technologies. – 2020. – Vol. 1 (2 (103)). – P. 30–36. DOI: https://doi.org/10.15587/1729-4061.2020.195041
33. Hybrid neuro-neo-fuzzy system and its adaptive learning algorithm / [Y. Bodyanskiy, O. Vynokurova, G. Setlak and I. Pliss] // Computer Sciences and Information Technologies (CSIT) : Xth Scien. and Tech. Conf., Lviv, 2015. – Lviv, 2015. – P. 111–114.
34. Subbotin S. The neuro-fuzzy network synthesis and simplification on precedents in problems of diagnosis and pattern recognition / S. Subbotin // Optical Memory and Neural Networks (Information Optics). – 2013. – Vol. 22, № 2. – P. 97–103. DOI: 10.3103/s1060992x13020082

DOI 10.15588/1607-3274-2021-3-10