http://doi.org/10.35784/iapgos.6957

received: 17.12.2024 | revised: 22.08.2025 | accepted: 27.08.2025 | available online: 30.09.2025

METHOD OF HYBRID LOGICAL CLASSIFICATION TREES BASED ON GROUP SELECTION OF DISCRETE FEATURES

Igor Povkhan[1], Andrii Leheza[1], Oksana Mulesa[1], Olena Melnyk[1], Aliya Kintonova[2]

[1]Uzhhorod National University, Department of Systems Software, Uzhgorod, Ukraine, [2] L.N. Gumilyov Eurasian National University, Department of Artificial Intelligence Technologies, Astana, Kazakhstan

Abstract. *The paper considers the problems of automating the construction of classification trees based on the scheme of branched feature selection. The object of research is classification trees. The subject of research is methods, algorithms, and schemes for constructing classification trees. The aim of this work is to build an effective method (scheme) for synthesizing classification tree models based on a group assessment of the importance of discrete features within a branched attribute selection. A method for constructing classification trees is proposed, which for a given training sample determines the individual information content (importance) of groups of features (and their combinations) in relation to the initial value of the classification function (data from the training sample). The developed logical tree method, when constructing the next node of the classification tree, tries to identify a group of the most closely interrelated discrete features, this reduces the overall structural complexity of the model (the number of levels of the classification tree), speeds up calculations when recognizing objects based on the model, and also increases the generalizing properties of the model and its enterprise. The proposed scheme for selecting groups of discrete traits allows using the constructed decision tree to assess the informative value (importance) of traits. The developed classification tree method is implemented programmatically and studied when solving the problem of classifying discrete objects represented by a set of features. The conducted experiments confirmed the operability of the proposed mathematical support and allow us to recommend it for use in practice in solving applied problems of classification of discrete objects based on logical classification trees. Prospects for further research may consist in creating a modified method of the logical classification tree by effectively iterating and evaluating sets of elementary features based on the proposed method, optimizing its software implementations, and experimentally studying the proposed method on a wider set of applied problems.*

Keywords: decision trees, classifier, discrete feature, branching criterion

METODA HYBRYDOWYCH LOGICZNYCH DRZEW KLASYFIKACYJNYCH OPARTYCH NA GRUPOWYM WYBORZE CECH DYSKRETNYCH

Streszczenie. *W artykule rozważono problemy związane z automatyzacją tworzenia drzew klasyfikacyjnych w oparciu o schemat rozgałęzionego wyboru cech. Przedmiotem badań są drzewa klasyfikacyjne. Tematem badań są metody, algorytmy i schematy tworzenia drzew klasyfikacyjnych. Celem niniejszej pracy jest opracowanie skutecznej metody (schematu) syntezy modeli drzew klasyfikacyjnych w oparciu o grupową ocenę znaczenia cech dyskretnych w ramach rozgałęzionego wyboru atrybutów. Zaproponowano metodę konstruowania drzew klasyfikacyjnych, która dla danej próbki szkoleniowej określa indywidualną zawartość informacyjną (znaczenie) grup cech (i ich kombinacji) w odniesieniu do wartości początkowej funkcji klasyfikacyjnej (dane z próbki szkoleniowej). Opracowana metoda drzewa logicznego, podczas konstruowania kolejnego węzła drzewa klasyfikacyjnego, próbuje zidentyfikować grupę najbardziej powiązanych ze sobą cech dyskretnych, co zmniejsza ogólną złożoność strukturalną modelu (liczbę poziomów drzewa klasyfikacyjnego), przyspiesza obliczenia podczas rozpoznawania obiektów na podstawie modelu, a także zwiększa właściwości uogólniające modelu i jego przedsiębiorstwa. Proponowany schemat wyboru grup cech dyskretnych pozwala na wykorzystanie skonstruowanego drzewa decyzyjnego do oceny wartości informacyjnej (znaczenia) cech. Opracowana metoda drzewa klasyfikacyjnego została zaimplementowana programowo i zbadana podczas rozwiązywania problemu klasyfikacji obiektów dyskretnych reprezentowanych przez zbiór cech. Przeprowadzone eksperymenty potwierdziły funkcjonalność proponowanego wsparcia matematycznego i pozwalają nam polecić je do praktycznego zastosowania w rozwiązywaniu problemów stosowanych związanych z klasyfikacją obiektów dyskretnych w oparciu o logiczne drzewa klasyfikacyjne. Perspektywy dalszych badań mogą polegać na stworzeniu zmodyfikowanej metody logicznego drzewa klasyfikacyjnego poprzez skuteczne iterowanie i ocenę zestawów cech elementarnych w oparciu o proponowaną metodę, optymalizację jej implementacji programowych oraz eksperymentalne badanie proponowanej metody na szerszym zestawie problemów stosowanych.*

Słowa kluczowe: drzewa decyzyjne, klasyfikator, cecha dyskretna, kryterium rozgałęzienia

Introduction

The complex of discrete object classification tasks arises in many applied fields of both economic and social human activities. The main purpose of creating classical recognition systems (RS) is the need to automate a group of processes related to perception, search, extraction, and classification of patterns based on certain operations on real data obtained in one way or another [1, 2]. Information technologies based on mathematical models of pattern recognition (namely, models of decision trees of various types) are widely used in various systems for collecting and processing discrete information, as they have the ability to eliminate the shortcomings of classical neural network recognition methods and achieve fundamentally new results by rationally using the power of computing systems [3, 8, 24].

The central task in the theory of pattern recognition is the construction, based on certain theoretical and experimental studies, of effective computing systems for assigning objects to corresponding classes. It is clear that for such a class of applied problems, logical classification tree models (LCT structures based on a branched feature selection) are a fairly effective tool, representing inductively constructed models that are trained based on an array of data of some fixed information [16, 17].

The object of research is a class of classification models in the form of LCT structures of all types and structures.

Thus, the class of problems that are reduced to the theory of pattern recognition is extremely diverse, and, most importantly, there is no universal approach to solving them that would provide comparatively high speed and acceptable quality (efficiency) when solving arbitrary problems. The reason for this situation is the central problem of the theory of pattern recognition – the problem of selecting (constructing) features (vertices in the LCT structure). A universal solution to this problem would allow the automatic creation of a universal RS, effective for any pattern recognition tasks [11].

The logical classification tree method (feature selection branching method) presented in the paper represents a promising basis for further research in the field of mathematical design and optimization of the resulting classification models in the form of LCT structures with vertices based on generalized features. This particular feature in the methods of branched feature selection gives them the most important advantage over other approaches. The flexibility of systems on feature selection branching lies in the wide possibility of manipulating features (elementary and generalized features) that are selected at each step of generating the resulting classification rule. The ability to regulate recognition quality in feature selection branching schemes lies in the fact that the logical recognition tree can be built indefinitely long on a limited information, thereby achieving a certain degree of approximation accuracy (that is, recognition quality) [5, 27].

The subject of research is the methods, schemes, and algorithms for constructing logical classification trees.

The purpose of the work is to create a simple and effective method for constructing classification models based on decision trees, with individual groups (sets) of elementary (discrete) features as vertices.

1. Problem statement

Let there $H_1, H_2, ..., H_k$ be a system of classes (images) defined on a set G consisting of objects $x_i, (i = 1,...,m)$. The nature of the partitioning of the set G into corresponding classes is specified by a training sample (TS) of the following form:

$$((x_1, f_R(x_1)), (x_2, f_R(x_2)), ..., (x_m, f_R(x_m))) \quad (1)$$

Note that here $x_h \in G$, $f_R(x) \in \{0,1,...,k-1\}$, $(h = 1, 2, ..., m)$, k – the number of classes in the TS, m – the total number of objects in the TS, and is $f_R(x)$ – a certain finite function that defines the partition of the set G into corresponding images (basic classes). The relationship is $f_R(x_h) = l, (l = 0,1,...,k-1)$ signifies that $x_h \in H_l$. It is noteworthy that each TS of form (1) can be associated (via some algorithm or representation method) with a fully defined classification decision tree (LCT), which assigns the values of the function to the objects $x_h, (h = 1,...,m)$ in TS (1), defining $f_R(x_h)$ the partitioning R on the set G. Therefore, the task will be to construct a classification tree (LCT/ACT) with a structure that would be optimal relative $f_R(x_j) \to opt$ to the initial TS data [8, 9].

2. Review of the literature

In the works [6], a method for synthesizing classification models based on the assessment of the informativeness of individual discrete features (attributes) has been developed. The central problem of this classification tree generation scheme is the high structural complexity of the constructed model and the issue of overfitting based on training data. In the works [23], a solution to the structural limitations of decision logic tres (LCT) is provided through a modular scheme for building classifiers, which overcomes the constraints of traditional decision tree methods. This raises the important task of building an efficient scheme for generating generalized features for such structures (models of ACT) [22]. The disadvantage of this approach lies in the limitations on the computational resources of the system when constructing ACT models and its lack of universal applicability.

The principal limitation of decision tree methods, associated with assessing the structural complexity of built models and methods of their final pruning (minimization of LCT structures), is the subject of the works [21]. From the works [20], it is known that the constructed LCT model (it can be constructed by any method or heuristic algorithm) has a tree-like logical structure. This then raises the central question of decision tree theory – the problem of choosing a quality branching criterion. The LCT structure consists of nodes grouped by levels, obtained at a certain step in constructing the recognition tree [19, 28]. One possible way to modernize decision tree methods (standard LCT structures) is through the synthesis of recognition trees, which essentially represent a tree of algorithms [23].

An important case arises when discrete features are individually low-informative [28]. In this situation, known decision tree methods will generate very deep and structurally complex classification trees that may not provide the necessary accuracy or complexity for the model and will have a low level of data generalization. This will lead to high hardware resource consumption (system memory and processing time) [21]. Thus, in many tasks, the features available for measurement are individually low-informative, yet collectively possess sufficient group informativeness for constructing some generalized feature (a node of the decision logic classification tree) [22, 27]. This is why the issue of building LCT structures for cases of low-informative discrete features arises.

The important question of generating and decomposing classification rules within decision tree structures is raised in the work [13]. The task of effectively assessing the informativeness of features [12] when constructing classification tree nodes remains central. The work [28] addresses fundamental issues regarding the generation of decision trees for low-informative features. A possible way to improve this work could be the use of combinations and sets of features to generate informative nodes in LCT structures [22].

This task is the focus of the presented work.

3. Branched feature selection in decision tree method

General scheme of the branched feature selection method. All algorithms based on the method of branched selection of discrete features follow the following scheme: first, a certain elementary feature ϕ is selected. It is required that its importance concerning the training sample is maximal compared to other features. Feature importance is understood as the following value:

$$W(\phi) = \sum_{i=1}^{m} (b_i / h) * \rho_i, \ \rho_i = \max(q_i^m / b_i) \quad (2)$$

Similarly, the importance of other features can be assessed [28]. The value (q_i^m / b_i) can be interpreted as the probability that function $f(x)$ takes the value O_m under the condition that the value of feature ϕ equals i. The value ρ_i represents the maximum of these probabilities. One can say that the value ρ_i represents the information that can be obtained about the value of function $f(x)$, knowing that on the set d, the value of feature ϕ equals i. The value $W(\phi)$, defined by this formula, characterizes the information that can be obtained about function $f(x)$ if the value of feature ϕ is known on the set d. It is clear that the feature for which this information is the greatest is considered the most important feature concerning $f(x)$.

Formula (2) can be used as a branching criterion in branched feature selection methods (BFS). Thus, all steps of the branched feature selection scheme can be represented using the following tree (Fig. 1).

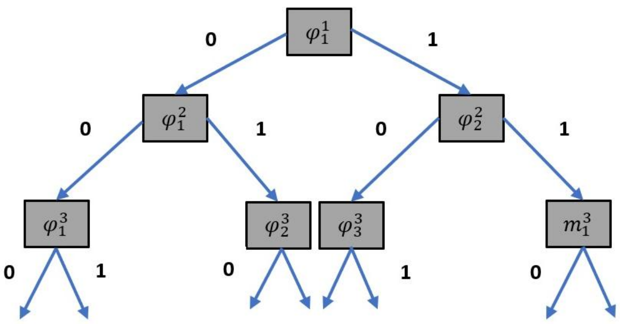


Fig. 1. General scheme of the classification tree based on BFS

Note that at each node of this tree (structure LDC), there is either a certain feature ϕ_i^j or a number m_i^j (this number characterizes membership in a certain class $H_{m_i^j}$). The node where the number m_i^j (the value of the recognition function) stands is called the terminal node of the tree. Two arrows branch off from each node where feature ϕ_i^j stands, marked as 0 and 1.

The arrow marked as 0 corresponds to the value $\phi_i^j = 0$, and the one marked as 1 corresponds to the value $\phi_i^j = 1$ (the feature takes only two values; otherwise, the number of arrows increases). The tree consists of layers. In the $j-$th layer, there are features $\phi_1^j, \phi_2^j, ...,$. All the features that stand in all layers of the tree, starting from the first and ending with the $n-$th, represent those features obtained after conducting $n-$ steps of the BFS scheme. Moreover, the features in the $n-$th layer represent all those features obtained in the $n-$th step of this process. In this example, only three steps of the BFS process are presented, and $\phi_1^1, \phi_1^2, \phi_2^2, \phi_1^3, \phi_2^3, \phi_3^3$ – are all the features obtained as a result of these steps (stages of constructing the LCT structure). The mathematical justification of this LCT scheme can be taken from the works [23, 28].

Scheme of the BFS method based on group selection of elementary features. Similarly to the BFS scheme described above, the structure of LCT can be built based on the selection of groups of elementary features. To do this, we introduce an informativeness assessment of an arbitrary group of discrete features $\phi_{i_1}, \phi_{i_2}, ..., \phi_{i_s} (1 \le i_1, i_2, ..., i_s \le h)$. For example, we fix a certain group of elementary features $\phi_1, \phi_2, ..., \phi_y (2 \le y \le h)$. Here as well, let the G – set of elementary features $\phi_1, ..., \phi_h$ be defined on which of recognition function (RF) $f_R(x), (x = \phi_1, ..., \phi_h)$ is set. The hypothesis of adequacy is imposed on the set G (G approaches the domain of definition of $f_R(x)$).

Let us fix $\Delta = t_1, t_2, ..., t_y$ as an arbitrary set of values of elementary features. Then denote U_Δ the number of all sets $\phi_1, ..., \phi_h$ from G for which the relationship holds $\phi_j = t_j$. Then denote U_Δ^m the number of sets $\phi_1, ..., \phi_h$ from G for which the relationships $\phi_j = t_j$ and $f_R(\phi_1, ..., \phi_h) = O_m$ (here h the cardinality of the set G), and $C-$ the set of all sets of features $\phi_1, \phi_2 ..., \phi_y$. Then the informativeness $W(\phi_1, \phi_2 ..., \phi_y)$ of a certain group (set) of elementary features is defined by formula (3).

$$W(\phi_1, \phi_2 ..., \phi_y) = \sum_{\Delta \in C} (U_\Delta / h) * \theta_\Delta \tag{3}$$

Note that here $\theta_\Delta = \max_m (U_\Delta^m / U_\Delta)$. For the justification of (3), we can fix $y, (1 \le y \le h)$ and among all groups $\phi_{i_1}, \phi_{i_2}, ..., \phi_{i_y}$ find the most informative group of elementary features. Clearly, for large y and h, the calculation of the most informative group of features $\phi_{i_1}, \phi_{i_2}, ..., \phi_{i_y}$ is associated with a large volume of computations, namely, iterations, which can be calculated by formula (4).

$$P_h^y = \frac{h!}{y!(h-y)!} \tag{4}$$

It should be noted that formula (3) represents a branching criterion in the LCT structure, the vertices of which are some generalized features (sets or groups of discrete features). The idea of a generalized feature can be borrowed from the work [23]. Emphasize that such an LCT structure with vertices in the form of generalized features is significantly approximated to ACT structures from works [22, 23]. The informativeness assessment (3) of a group (set) of discrete features will be referred to as the importance functional of a group of features (IFGF).

It should be noted that (3) evaluates the informativeness of a group of discrete features relative to $W(\phi_1, \phi_2 ..., \phi_h)$ and has an unambiguous connection with the classification errors of TS objects. That is, the numerical value of the IFGF coincides with the proportion of TS objects that can be correctly classified into TS classes (naturally, using only the selected group of elementary features when classifying discrete objects). It should also be noted that based on (3), an effective procedure for ranking subsets of features can be carried out, allowing the construction of sets of generalized features (vertices of LCT structures of a new type).

An important point of the IFGF is that an arbitrary subset of the set $\phi_{i_1}, \phi_{i_2}, ..., \phi_{i_s}, (1 \le s \le h)$ of discrete features $\phi_1, ..., \phi_h$ forms some test of the initial TS if and only if:

$$W(\phi_{i_1}, \phi_{i_2}, ..., \phi_{i_s}) = 1 \tag{5}$$

Here, by a test, we mean some set of elementary features $\phi_{i_1}, \phi_{i_2}, ..., \phi_{i_s}$ that allows an unambiguous separation of the classes of the initial TS.

It should also be noted that for (3), the following relationship will hold:

$$\frac{1}{k} \le IFGF \le 1 \tag{6}$$

It should be noted that in (6) $k-$ the total number of classes of the initial TS.

Fast calculation scheme for the importance of groups of discrete features in the LCT method. Thus, for the calculation IFGF, it is not necessary to store an array of discrete features in the computer memory (to remember all sets of the TS set). The value $W(\phi_i)$ can be calculated sequentially when training pairs $(x_1, f_R(x_1))$ are fed. That is, the following algorithm for calculation $W(\phi_i)$ can be proposed.

Let at some step of calculating IFGF, pairs z of TS of the form have already been presented, and for these training pairs $(x_1, f_R(x_1))$, the values b_{ij} and q_{ji}^m are calculated (here $j = 0,1,..., k_j - 1; i = 1,2,..., h; m = 0,1,..., k-1$).

The value b_{ij} is the number of all sets from training TS pairs in which feature ϕ_i took the value j.

The value q_{ji}^m is the number of all sets from training TS pairs in which feature ϕ_i took the value j, and the function $f_R(x)$ took the value O_m.

Then, when receiving the next ($h+1$) pair $(l_1, l_2, ..., l_n, O_v)$, where $0 \le l_i \le k_i - 1; (0 \le v \le k-1)$, the values z, b_{ij}, q_{ji}^m change as follows:

1. The value z increases ($z+1$).
2. The value $b_{ji} = b_{ij}$ if $j \ne l_i$; $b_{ji}^` = b_{ji} + 1$, if $j = l_i$.
3. The value $(q_{ji}^m)^` = q_{ji}^m$ if $j \ne l_i$ or $m \ne v$; $(q_{ji}^m)^` = q_{ji}^m + 1$, if $j = l_i$ and $m \ne v$.

Thus, with this scheme for calculating the importance of discrete features, only the values v, b_{ij}, q_{ji}^m are stored in the computer memory. It is clear that according to this algorithm, these values should be calculated until the value $W(\phi_i)$ begins to stabilize near some values l_i. Then the set of values l_i can be accepted as an estimate of the importance of the corresponding elementary features ϕ_i.

Hence, sets of IFGF can be calculated according to this scheme. The application of such algorithms is particularly effective in conditions where there are some limitations on the volumes of RAM.

Attention is drawn to the fact that the functional (3) allows relatively simple (in terms of saving system computational resources) to find sets of tests (primarily minimal tests are of interest), as well as the importance of sets and groups of discrete features in the case when the initial TS consists of more than two classes. Minimal tests are fundamentally important in terms of constructing the optimal structure of LCT (both conventional and modified based on vertices – generalized features), which do not require final trimming or structure minimization.

In some cases, in applied problems, it is very difficult (or completely impossible) to find a test. In such cases, a possible solution is to search for a set of discrete features that would allow effectively distinguishing objects of one class from objects of another. Such a group of features is called a quasi-test, and formula (3) allows finding quasi-tests directly based on the given TS.

It should be noted that based on (3), an effective mechanism for finding sets of elementary features can be organized, ensuring the recognition of the initial TS with a predetermined accuracy. That is, for LCT structures, this allows building decision trees of fixed complexity (structural complexity, structural depth) and fixed recognition accuracy. On the other hand, the most informative group of discrete features can be calculated (relative to the function that specifies the division of the initial TS) with a fixed number of features in the group.

An important property of functional (3) is that the importance of discrete features (groups or sets of features) and finding tests is calculated in parallel. This allows highlighting a group of features for which $W(\phi_1, \phi_2, ..., \phi_k) < 1; (k \geq 2)$, as some quasi-test recognizing TS objects with some accuracy W. In various test search problems, subtasks for determining the informativeness of elementary features are solved sequentially, increasing the resource and processing time costs of the information system.

It should be noted that by analogy with LCT structures, the functional (3) can be used to select attributes in the vertices of graph-schematic structures of boolean and multivalued logical functions in tasks of their minimization.

In the problem of assessing the completeness of a set of elementary features $\phi_1, \phi_2, ..., \phi_k$, the value $U(\phi_1, \phi_2, ..., \phi_k) = \min_{\Delta \in C} \theta_\Delta$ is of great importance, here θ_Δ and Δ have the same meaning as in (3). It should be noted that this assessment is stricter than $W(\phi_1, \phi_2, ..., \phi_k)$, and therefore it should be applied in problems where stricter accuracy requirements for object classification $x_i; (i = 1, ..., m)$ are imposed, when it is necessary to minimize the number of classification errors (of all types). Similarly, the functional $W^{`}(\phi_i) = W(\phi_1, ..., \phi_{i-1}, \phi_{i+1}, ..., \phi_k)$ can be introduced. In this case, the most important discrete feature under the constraint is considered $f_R$ to be such a feature $\phi$ for which the value $W^{`}(\phi_i)$ will be minimal.

4. Experiments

Based on a fixed training dataset, using a branched feature selection, it is possible to construct decision tree structures of various types with a set of different constraints regarding structural complexity (number of structure levels) and the accuracy of the final classification model. For hybrid decision tree structures based on generalized features as vertices (groups and sets of discrete features), functional (3) is used as the branching criterion.

It is noted that constructing such decision tree structures requires significantly more hardware resources for the classification system compared to classical decision tree models [18, 30]. In this case, the question arises of evaluating the constructed classification models (decision trees) to choose the best one relative to the current training dataset.

After the final trimming (minimization) procedure of the classification model, the next stage involves the analysis and comparison of synthesized tree-like recognition models (decision tree structures of different types and structural complexities). At this stage, it is necessary to determine sets of local indicators (evaluation characteristics) that most informatively describe the basic properties of the obtained classification models. In the vast majority of cases, the evaluation stage of constructed decision tree models is conducted based on an overall integral quality indicator – the central criterion for model comparison (decision tree structures). Based on the works [21–23], the following basic characteristics (indicators) of synthesized decision tree structures can be highlighted in the following composition:

V_{TR} – the number of vertices of all types in the decision tree structure. Logically, the minimum value of this quantity will be one more than the power of the initial training dataset.

N_{TR} – the number of elementary features on which the decision tree is based.

C_{TR} – the number of transitions, connections between levels in the decision tree structure.

O_{TR} – the number of final vertices (recognition function values) in the decision tree structure. In hybrid decision tree structures, this also includes unfinished transitions in the tree structure.

En_{TR} – the generalized number of errors of all types in the classification model on the training dataset.

Et_{TR} – the generalized number of errors of all types in the classification model on the test dataset.

Er_H – the number of classification errors for each of the classes of the initial training dataset.

The most important in this list are the indicators of all types of classification errors, which directly affect the integral quality assessment of the constructed decision tree model. Introducing a set of indicators of the decision tree classification model concerning its basic characteristics:

$C_{AVG} = (C_{TR} / V_{TR})$ – the average number of connections (transitions from other levels) per vertex in the decision tree structure.

$N_{TR}^V = (N_{TR} / n)$ – the total share of discrete features used in the decision tree structure.

$O_{TR}^V = (O_{TR} / V_{TR})$ – the total share of resulting recognition function values (tree leaves) in the decision tree structure.

$Q_{AVG} = (m / O_{TR})$ – the average number of initial training dataset sets per resulting recognition function value (tree leaves) in the decision tree structure.

At the next stage, a general indicator of the logical tree model for generalizing the data of the initial training dataset can be introduced in the following form:

$$I_{MAIN} = (m * n) / (V_{TR} + 2C_{TR}) \tag{7}$$

The most important task remains the problem of reducing the structural complexity of the constructed decision tree model (final pruning procedure LCT). This highlights indicators N_{TR} such as the number of features in the decision tree structure, V_{TR} the number of vertices in the decision tree model, and C_{TR} the total number of transitions in the decision tree structure, as well as memory μ and processor time τ expenses. Minimizing these parameters allows for good results in terms of model classification performance, structural complexity of the final decision tree model, and resource consumption of the information system.

Considering all the mentioned decision tree model parameters, the greatest resource savings of the system can be achieved by increasing the indicators O_{TR} and O_{TR}^V. This approach will significantly reduce the overall classification time for the logical tree model (total decision-making time based on the classification rule) and save processor time.

Given the characteristics of parameter I_{MAIN} (the generalization indicator of the decision tree model), it is clear that it should also be maximized, which will ensure the most optimal (trimmed) decision tree structure. Maximizing parameter I_{MAIN} should also provide maximum compression of the initial training dataset. This allows for the representation of the initial data array with a minimally structurally complex decision tree (decision tree structure of any type). It is clear that the parameter I_{MAIN} should be considered together with the parameter Q_{AVG} (average number of training dataset sets per resulting recognition function value), which allows evaluating the degree of generalization of the training dataset by the decision tree structure.

At the next stage, it is fixed that the central quality indicator of the constructed decision tree classification model (decision tree structure) considering the above-mentioned parameters is the integral quality indicator of the model in the following form:

$$Q_{MAIN} = (O_{TR} * e^{\lambda}) / (N_{TR} * V_{TR} * C_{TR}) \qquad (8)$$

It should be noted that here:

$$\lambda = -\frac{\left| En_{TR} * Et_{TR} - \delta^2 \right|}{M * M_{TS}} \qquad (9)$$

From the given formulas, it is clear that this integral quality indicator of the decision tree model (8) makes sense only if $(En_{TR} / M) \le \delta$. In the case of $(En_{TR} / M) > \delta$, it will equal zero. Here represents M_{TS} the overall power of the test dataset. Thus, it can be concluded that an increase in this indicator indicates an improvement in the quality of the decision tree model, and vice versa, a decrease indicates a deterioration in classification (recognition) quality.

5. Results

Verification of the proposed methods (schemes for constructing LCT structures based on sets and groups of elementary features as vertices) was carried out using the "Orion III" software complex (a system for generating autonomous recognition and classification systems). The "Orion III" software complex was developed at Uzhhorod National University, and its library of autonomous classifiers includes 15 recognition algorithms. Among them are simple geometric recognition algorithms as well as the algorithmic implementation of the hybrid LCT structure construction proposed above (based on generalized features).

The method for constructing the hybrid LCT structure was tested on the problem of geological data classification – specifically, the task of identifying oil-bearing layers in geological research. The classification task utilized 13 primary elementary features and 11 additional ones (the feature space dimension is 24). The LCT structure is characterized by objects of two classes. The total volume of the training sample consisted of 1395 objects (of which 809 are oil-bearing), and the test sample volume was 462 objects.

It is worth noting that the training and test sample data were obtained based on geological surveys conducted in the Zakarpattia region (Ukraine) from 2001 to 2021. The general parametric characteristics of the geological object classification task are presented in (table 1). The main results of the above experiments (various decision tree models – logical and algorithmic classification trees) are presented in (table 2), respectively. It should be noted that the constructed LCT models within this work ensured the required accuracy level specified by the task conditions (the classification accuracy level can be effectively adjusted), decision-making speed, and memory efficiency of the information system.

Taking all of the above into account, it can be concluded that the proposed quality assessments of the model (integral quality indicator of the LCT structure) of the logical decision tree reflect the most important characteristics (parameters) of the decision tree. Thus, the proposed quality indicators of the LCT model can be applied as an optimality criterion in the synthesis tasks of classification models based on logical decision trees (tasks of pruning LCT structures, constructing, and selecting LCT sets).

The construction of decision tree models was carried out on the following hardware configuration – Intel i7-13800h. In (table 2) presents general information about the type of classification tree model, its structural complexity indicator S_{MAIN} (number of vertices, attributes, and structural transitions), the number of classification errors (failures) of all types Er_{All}, the time to generate classification models, and the overall integral quality indicator of the decision tree model (LCT/ACT structures).

Table 1. Initial parameters of the classification problem

Description of classes H_l tasks	The dimension of the feature space N	The power of data array of the primary TS – M	The total number of classes by data splitting TS – l	Relation of test sample classes (H_i/TS)	Relation of objects of different classes TS – (H_i/M)
Oil-bearing layers (H_1)	(13/11)	1395	2	284/462	809 / 1395
Aquifers (H_2)	(13/11)	1395	2	178/462	586 / 1395

Table 2. Comparative table of constructed ACT/LCT models

Decision tree-based classification model (LCT/ACT structure)	Integral feature of model quality (LCT/ACT) Q_{Main}	Indicator of structural complexity of the model (LCT/ACT) S_{Main}	Model classification errors / failure rate (LCT/ACT) Er_{All}	Time to build (model) the decision tree T_{All}
LCT method based on peer-to-peer branched feature selection [21–23]	0.002219	140	11	44 s
Constraint-based LCT method (Z = 5) [21–23]	0.003016	99	15	39 s
Algorithmic tree method (type I) [21–23]	0.005286	51	11	64 s
Algorithmic tree method (type II) [21–23]	0.003032	63	10	81 s
ACT method based on constraints (Z = 4) [21–23]	0.002661	57	15	62 s
Hypersphere-based ACT method	0007219	30	7	51 s
LCT method based on selection of sets (groups) of elementary traits (S = 2)	0.002356	125	9	49 s
LCT method based on the selection of sets (groups) of elementary traits (S = 3)	0.002531	112	10	54 s
LDK method based on the selection of sets (groups) of elementary traits (S = 4)	0.003014	114	11	58 s

During the experiments, the hybrid LCT scheme based on sets (groups) of elementary features as vertices (generalized features) was compared with both LCT methods (schemes of branched selection of features with single and stepwise informativeness assessment) and algorithmic trees (two types of ACT structures) and showed acceptable results within the scope of the task.

It should be noted that algorithmic tree methods are based on an approximation scheme using a set of algorithms for the initial training sample. For LCT methods, the scheme is based on the approximation by a set of elementary features (a group of discrete features for hybrid LCT structures) of the initial training sample.

It can be seen that compared to LCT structures, algorithmic trees have significant advantages in terms of universality and relatively compact model structure. It is clear that such classification model advantages require significantly greater resource expenditures for storing generalized features (sets of classifiers) and stepwise evaluation of classification algorithm efficiency. However, LCT structures of all types have high classification rule speed, reduced resource requirements of the information system for storing LCT/ACT structures, and adjustable classification accuracy.

6. Conclusion

The method for constructing hybrid LCT structures (based on the selection of groups of elementary features) and the constructed decision tree models demonstrated the equired level of classification accuracy compared to other classification tree methods. The basis of such hybrid structures (LCT models) lies in an effective scheme for evaluating the informativeness of sets (groups) of discrete features. The main drawback of this approach remains the need to iterate over feature sets to highlight the most informative groups (sets) of discrete features. It is clear that direct iteration of TS relative to elementary features is accompanied by significant resource costs for the information system and may only be appropriate with restrictions on the length of the set of elementary features (S).

Among all the constructed decision tree models, the high efficiency of tree algorithm models in terms of classification accuracy and structural compactness (S_{Main}) should be noted. In these classification models, the simple geometric hypercube algorithm showed the greatest efficiency (the most effective classifier for all types of ACT structures). It should be noted that the constructed ACT models showed a relatively low number of classification errors of all types during operation (Er_{All}). For example, the first type of tree algorithm based on the hypercube (a model based on restrictions on classifier selection) demonstrated a good result ($Q_{Main} = 0.007219$). The LCT structure based on branched feature selection (a complete classification tree) showed a result of ($Q_{Main} = 0.002219$). In contrast, the second type of tree algorithm demonstrated acceptable accuracy and quality ($Q_{Main} = 0.003032$) on the one hand, but significant resource and time costs on the other ($T_{All} = 81$ s). The advantage of ACT structures is explained by the more complex design of the model (sets of generalized features – independent classifiers), which requires a longer generation time (model construction).

It should be noted that ACT models can effectively work in LCT structure tasks, but there are cases where it is economically feasible to use logical classification trees restrictions on complexity, model generation time, or computational resources). A noticeable disadvantage of the constructed tree algorithm models is the relatively high time expenditure at the stage of synthesizing classification tree models, especially compared to LCT structures. The difference in time for constructing ACT models (only models with a complete structure and without restrictions) compared to LCT trees was at least 30%.

The scientific novelty of the work lies in the development of a method for constructing hybrid LCT structures based on group selection (sets) of elementary features (tree vertices in the form of generalized features) for the first time. The results obtained are of significant practical importance. The proposed method for constructing hybrid LCT structures (classification tree models) allows building effective classifiers with predefined accuracy. The hybrid LCT structure method was integrated into the algorithm library of the "Orion" software system, enabling the solution of various applied classification tasks and demonstrating a high degree of versatility in various fields of application.

All constructed classification tree models and the developed software demonstrated their effectiveness in applied tasks, allowing the recommendation to expand the scope of hybrid LCT structures (decision trees based on group selection of elementary features). A possible future direction of research could be the further development of LCT/ACT methods (structures), including the introduction of new types and schemes of classification trees. Moreover, the optimization of software implementations (parallelization of the algorithm for calculating the informativeness of discrete features) of the proposed hybrid LCT method and its practical testing in various tasks of accurate classification and recognition could provide valuable information and improvements in this field.

References

[1] Bodyanskiy Y., et al.: Hybrid neuro-neo-fuzzy system and its adaptive learning algorithm. Xth International Conference on Computer Science and Information Technologies – CSIT 2015, Ukraine, 2015, 111–114.
[2] Bodyanskiy Y.V., Shafronenko A.Y., Pliss I.P.: Credibilistic fuzzy clustering based on evolutionary method of crazy cats. System Research and Information Technologies 3, 2021, 110–119 [https://doi.org/10.20535/SRIT.2308-8893.2021.3.09].
[3] Breiman L. L., et al.: Classification and regression trees. Chapman and Hall, 1984.
[4] Dang M., et al.: Computer Vision for Plant Disease Recognition: A Comprehensive Review. The Botanical Review 90(3), 2024, 251–311 [https://doi.org/10.1007/s12229-024-09299-z].
[5] De Mántaras R. L.: A distance-based attribute selection measure for decision tree induction. Machine learning 6(1), 1991, 81–92.
[6] Deng H., Runger G., Tuv E.: Bias of importance measures for multi-valued attributes and solutions. 21st International Conference on Artificial Neural Networks – ICANN 2011, Finland, 2011, Part II, 293–300 [https://doi.org/0.1007/978-3-642-21738-8_38].
[7] Ferreira J. J., de Souza Monteiro M.: Do ML Experts Discuss Explainability for AI Systems? A discussion case in the industry for a domain-specific solution. arXiv, 2002.12450, 2020.
[8] Hastie T., Tibshirani R., Friedman J.: The Elements of Statistical Learning. Springer, 2008.
[9] Harakannanavar S. S., et al.: Plant Leaf Disease Detection Using Computer Vision and Machine Learning Algorithms. Global Transitions Proceedings 3(1), 2022, 305–310 [https://doi.org/10.1016/j.gltp.2022.03.016].
[10] Imamovic D., Babovic E., Bijedic N.: Prediction of mortality in patients with cardiovascular disease using data mining methods. 19th International Symposium Infoteh-Jahorina – INFOTEH, Bosnia and Herzegovina, 2020, 1–4 [https://doi.org/10.1109/INFOTEH48170.2020.9066297].
[11] Jaman S. F. I., Anshari M.: Facebook as marketing tools for organizations: Knowledge management analysis. Dynamic perspectives on globalization and sustainable business in Asia. IGI Global 2019, 92–105 [https://doi.org/10.4018/978-1-5225-7095-0.ch007].
[12] Kamiński B., Jakubczyk M., Szufel P.: A framework for sensitivity analysis of decision trees. Central European Journal of Operations Research 26(1), 2017, 135–159.
[13] Karimi K., Hamilton H.: Generation and Interpretation of Temporal Decision Rules. International Journal of Computer Information Systems and Industrial Management Applications 3, 2011, 314–323.
[14] Kintonova A., et al.: Improvement of iris recognition technology for biometric identification of a person. Eastern-European Journal of Enterprise Technologies 6(2(120)), 2023, 60–69 [https://doi.org/10.15587/1729-4061.2022.269948].
[15] Koskimaki H., et al.: Two-level clustering approach to training data instance selection: a case study for the steel industry. IEEE International Joint Conference on Neural Networks – IJCNN 2008, Hong Kong, 2008, 3044–3049 [https://doi.org/10.1109/ijcnn.2008.4634228].
[16] Kotsiantis S. B.: Supervised Machine Learning: A Review of Classification Techniques. Informatica 31, 2007, 249–268.
[17] Miyakawa M.: Criteria for selecting a variable in the construction of efficient decision trees. IEEE Transactions on Computers 38(1), 1989, 130–141.

[18] Nayak A., et al.: Leveraging BERT-Enhanced MLP Classifier for Automated Stress Detection in Social Media Articles. International Conference on Advances in Computing Research on Science Engineering and Technology –ACROSET, IEEE, 2024 [https://doi.org/10.1109/ACROSET62108.2024.10743857].
[19] Orazbayev B., et al.: Decision-making in the fuzzy environment on the basis of various compromise schemes. Procedia computer science 120, 2017, 945–952 [https://doi.org/10.1016/j.procs.2017.11.330].
[20] Painsky A., Rosset S.: Cross-validated variable selection in tree-based methods improves predictive performance. IEEE Transactions on Pattern Analysis and Machine Intelligence 39(11), 2017, 2142–2153 [https://doi:10.1109/tpami.2016.2636831].
[21] Povkhan I.: Classification models of flood-related events based on algorithmic trees. Eastern-European Journal of Enterprise Technologies 6(4), 2020, 58–68 [https://doi.org/10.15587/1729-4061.2020.219525].
[22] Povkhan I., et al.: The issue of efficient generation of generalized features in algorithmic classification tree methods. IEEE Third International Conference on Data Stream Mining & Processing – DSMP–2020, Ukraine, 2020, 98–113.
[23] Povkhan I., Lupei M.: The algorithmic classification trees. IEEE Third International Conference on Data Stream Mining & Processing (DSMP–2020), Ukraine, 2020, 37–44.
[24] Quinlan J. R.: Induction of Decision Trees. Machine Learning 1, 1986, 81–106.
[25] Rabcan J., et al.: Application of Fuzzy Decision Tree for Signal Classification. IEEE Transactions on Industrial Informatics 15(10), 2019, 5425–5434 [https://doi.org/10.1109/tii.2019.2904845].
[26] Shyshatskyi A.: Complex Methods of Processing Different Data in Intellectual Systems for Decision Support System. International Journal of Advanced Trends in Computer Science and Engineering 9(4), 2020, 5583–5590 [https://doi.org/10.30534/ijatcse/2020/206942020].
[27] Strilets V. E., et al.: Methods of machine learning in the problems of system analysis and decision making. Karazin Kharkiv National University, 2020.
[28] Subbotin S. A.: Construction of decision trees for the case of low-information features. Radio Electronics, Computer Science, Control 1, 2019, 121–130 [https://doi.org/10.15588/1607-3274-2019-1-12].
[29] Wang M., Deng W.: Deep face recognition: A survey. Neurocomputing 429, 2021, 215–244 [https://doi.org/10.1109/SIBGRAPI.2018.00067].
[30] Zhuravlev Y. I., Nikiforov V. V.: Recognition algorithms based on the calculation of estimates. Cybernetics 3, 1971, 1–11.

Prof. Igor Povkhan
e-mail: igor.povkhan@uzhnu.edu.ua

Igor Povkhan, worked as the head of the laboratory of software and hardware of the faculty of information technology in the period from 2003 to 2008. Since 2009, professor of the Department of software systems of Uzhgorod national University. The direction of scientific work-artificial intelligence, pattern recognition theory, low-level programming.

https://orcid.org/0000-0002-1681-3466

M.Sc. Andrii Leheza
e-mail: andrii.legeza@uzhnu.edu.ua

Lecturer at the Department of Software Systems of Uzhhorod National University. Author of scientific and methodical works. Research interests: research into the effectiveness of modern teaching methods, particularly in the context of online learning Researching the possibilities of using artificial intelligence in the fields of scientific activity: Web programming, programming for mobile platforms.

https://orcid.org/0000-0001-8416-8836

Prof. Oksana Mulesa
e-mail: mulesa.oksana@gmail.com

Oksana Mulesa is a professor and Doctor of Technical Sciences working at the University of Prešov (Slovakia) and Uzhhorod National University (Ukraine). Her research interests include decision-making, fuzzy analysis, artificial intelligence, neural networks, and forecasting. She actively participates in national and international research projects.

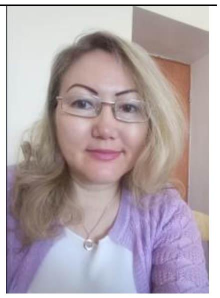
https://orcid.org/0000-0002-6117-5846

Ph.D. Olena Melnyk
e-mail: olena.melnyk@uzhnu.edu.ua

Ph.D., associate professor of Software Systems Department, Uzhhorod National University, Uzhhorod, Ukraine. She is the author/co-author of more than 50 scientific articles and conference abstracts, 8 training manuals. Scientific interests: information technologies, optimization, decision making theory, theory of scheduling, forecasting.

https://orcid.org/0000-0001-7340-8451

Ph.D. Aliya Kintonova
e-mail: Aliya_kint@mail.com



Kintonova Aliya, associate professor at the Artificial Intelligence Technologies Department of L. N. Gumilyov Eurasian National University, Ph.D. in Technical Sciences. Total publications: 187, including: articles published in Scopus: 13 (Hirsch index: 6); articles in Web of Science databases: 6; certificates of intellectual property: 17; textbooks: 3; study guides: 5.

https://orcid.org/0000-0002-8405-5038