UDC 004.93

DATA CLUSTERING BASED ON INDUCTIVE LEARNING OF NEURO-FUZZY NETWORK WITH DISTANCE HASHING

Subbotin S. A. – Dr. Sc., Professor, Head of the Department of Software Tools, National University "Zaporizhzhia Polytechnic", Zaporizhzhia, Ukraine.

ABSTRACT

Context. Cluster analysis is widely used to analyze data of various nature and dimensions. However, the known methods of cluster analysis are characterized by low speed and are demanding on computer memory resources due to the need to calculate pairwise distances between instances in a multidimensional feature space. In addition, the results of known methods of cluster analysis are difficult for human perception and analysis with a large number of features.

Objective. The purpose of the work is to increase the speed of cluster analysis, the interpretability of the resulting partition into clusters, as well as to reduce the requirements of cluster analysis to computer memory.

Method. A method for cluster analysis of multidimensional data is proposed, which for each instance calculates its hash based on the distance to the conditional center of coordinates, uses a one-dimensional coordinate along the hash axis to determine the distances between instances, considers the resulting hash as a pseudo-output feature, breaking it into intervals, which matches the labels pseudo-classes – clusters, having received a rough crisp partition of the feature space and sample instances, automatically generates a partition of input features into fuzzy terms, determines the rules for referring instances to clusters and, as a result, forms a fuzzy inference system of the Mamdani-Zadeh classifier type, which is further trained in the form of a neuro-fuzzy network to ensure acceptable values of the clustering quality functional. This makes it possible to reduce the number of terms and features used, to evaluate their contribution to making decisions about assigning instances to clusters, to increase the speed of data cluster analysis, and to increase the interpretability of the resulting data splitting into clusters.

Results. The mathematical support for solving the problem of cluster data analysis in conditions of large data dimensions has been developed. The experiments confirmed the operability of the developed mathematical support have been carried out.

Conclusions. . The developed method and its software implementation can be recommended for use in practice in the problems of analyzing data of various nature and dimensions.

KEYWORDS: cluster analysis, neuro-fuzzy network, hash, fuzzy inference, data analysis.

ABBREVIATIONS

NFN is a neuro-fuzzy network.

NOMENCLATURE

α is a user-specified coefficient;

β is a user-specified coefficient;

ε is an error threshold value specified by the user;

$< a_{j,q},\ b_{j,q},\ c_{j,q},\ d_{j,q} >$ are adjustable parameters of fuzzy term membership function;

C is a set of adjustable parameters;

C^k is a k-th cluster parameters;

E is a classification error;

F is a criterion of a model quality (user-specified)**;**

$F_{1,2}$ is a quotient from division of F for the first and second models;

$G_{1,2}$ is a quotient from division of I_G coefficients for first and second models;

$GF_{1,2}$ is a quotient from division of I_{GF} coefficients for first and second models

i is a feature number;

I_j. is an indicator of individual informativity of the j-th feature;

$I_{i,p,j,q}$ is a pairwise equivalence estimate for the terms of different input features;

$\bar{I}^t$ is an average estimate of the relationship between the terms of different input features;

$\bar{I}$ is an average estimate of the relationship of features;

$I(i,j)$ is an estimate of the pairwise relationship of the i-th and j-th input features;

$I_{j,q}$ is an indicator of the individual informativity of the q-th term of j-th feature for the entire set of classes;

I_G is a generalization indicator;

I_{GF} is a generalization of features indicator;

j is a feature number;

K is a number of subsets (clusters, pseudo-classes);

$l_{j,q}$ is a left boundary of the q-th interval of j-th feature;

$L_{j,q}$ is a length of the q-th interval of values of the j-th feature;

N is a number of features characterizing instances;

N' is a number of features after reduction;

n is a sample dimensionality;

N_j is a number of intervals into which the range of values of the j-th feature is divided;

$N_{j,q}$ is a number of times the q-th term of j-th feature was used in the rules;

$N_{j,q,k}$ is a number of instances of k-th class belonging to the q-th term of j-th feature;

N_w is a number of parameters of the clustering model;

opt is a formal designation of the optimum;

q^s_* is a number of the interval, in which s-th instance hit according to the hash value x^s_*;

q^s_j is a is number of the term of j-th feature to which s-th instance belongs (hit in);

q^s_j is a number of the interval in which s-th instance hit on j-th feature;

DOI 10.15588/1607-3274-2022-4-6

R_c is a conflict set of rules;

$|R_c|$ is a power (the number of rules in) of the conflict set;

$r_{j,q}$ is a right boundary of the q-th interval of j-th feature;

s is an instance number;

(s) is a rule number;

S' is a number of instances in a reduced set;

S is a number of instances in the sample;

t is a running time of the method;

$t_{1,2}$ is a quotient from division of t for the first and second methods;

$m_{1,2}$ is a quotient from division of m for the first and second methods;

w^s is a weight of s-th rule;

x is a sample of observations;

x^s is a s-th instance of a sample;

x^s_j is a value of j-th feature for s-th instance of a sample;

x^s_* is a hash value for the instance x^s;

y^s is a label of the output feature (pseudo-class or cluster) for s-th instance;

y^s_* is a calculated class number for the recognized instance x^s.

INTRODUCTION

The cluster analysis [1–17] is widely used to analyze data of various nature and dimensions. The purpose of cluster analysis is to split the initial sample of observations (instances) into compactly located groups of instances or to identify compact areas of grouping instances – clusters in the feature space that describe the sample instances.

The **object of study** is the process of cluster analysis of data samples.

There are two groups of cluster analysis methods: crisp [1–17] and fuzzy methods [18–26]. Unlike crisp methods, which provide a coarser separation of instances, fuzzy methods allow more adaptive selection of clusters in the feature space.

The **subject of study** is the methods of fuzzy cluster data analysis.

Fuzzy cluster analysis methods [18–26] are highly adaptive and require a large number of parameters to be adjusted. Also, the well-known methods of fuzzy and crisp cluster analysis are characterized by low speed and are demanding on computer memory resources due to the need to calculate pairwise distances between instances in a multidimensional feature space. In addition, the results of known methods of cluster analysis [1–29] are difficult for human perception and analysis with a large number of features.

The **purpose of the work** is to increase the speed of cluster analysis, the interpretability of the resulting partition into clusters, as well as to reduce the requirements of cluster analysis to computer memory.

1 PROBLEM STATEMENT

Suppose given the sample of observations $x=\{x^s\}$, $s = 1, 2,..., S$, $x^s = \{x^s_j\}$, $j = 1, 2, ..., N$, then the problem of cluster analysis of sample x is to determine the splitting of the sample x into K subsets (clusters, pseudo-classes) with parameters $C=\{C^k\}$:

$$x = \bigcup_{k=1}^{K} \{x^s \mid x^s \in C^k, s = 1,2,...,S\},$$

$$F(x, C) \to opt.$$

As a rule, such a criterion should ensure the minimization of distances between instances within the same cluster and the maximization of inter-cluster distances of instances [1, 12–14]. Here, the distance between instances in the feature space is considered as a measure of their similarity.

That is, for a given sample x, we need to constructively determine F and to find the optimal (or acceptable) values of C for it.

2 REVIEW OF THE LITERATURE

According to the type of membership functions used for instances to clusters, the known methods of cluster analysis are divided into crisp and fuzzy methods.

Well-known methods of crisp cluster analysis [1–17, 27–29] assume that the initial sample of observations is divided into clusters in such a way that each instance belongs to only one cluster, and the partition is formed iteratively from the initial random or user-defined partition to the final partition that satisfies the specified criterion quality. In fact, the main differences between the well-known methods of crisp cluster analysis [1–17, 27–29] are the method of calculating the distance, the quality criterion of the partition, the method of generating the initial partition (set of initial partitions), the method of generating a new partition (set of partitions) of the sample based on the existing (or previously considered), search termination criteria. In this case, the partition quality criterion is a function determined on the basis of pairwise distances between sample instances, as well as distances from instances to cluster centers in the feature space. The calculation of such distances for samples of large dimensions is a computationally expensive task and also requires significant memory costs to load the entire sample of observations into memory. Additionally, the task is complicated by the need for pairwise enumeration of distances between instances.

The well-known methods of fuzzy cluster analysis [18–26] assume that each sample instance belongs to all clusters, but with different values of the membership function, the splitting of instances into clusters is formed iteratively from the initial random or user-specified split to the final split that satisfies given quality criteria. In fact, the main differences between the known methods of fuzzy cluster analysis, as well as for the methods of crisp cluster analysis, are the method of calculating the distance, the quality criterion of the partition, the method of

DOI 10.15588/1607-3274-2022-4-6

generating the initial partition (set of initial partitions), the method of generating a new partition (set of partitions) of the sample based on the existing (or previously considered), criteria for terminating the search. At the same time, in contrast to crisp cluster analysis, fuzzy methods operate with cluster memberships calculated on the basis of the distance function of instances to cluster centers. Like crisp methods in fuzzy cluster analysis, the partition quality criterion is a function determined on the basis of pairwise distances between sample instances in the feature space. The calculation of such distances for samples of large dimensions is a computationally expensive task and also requires significant memory costs to load the entire sample of observations into memory. Additionally, the task is complicated by the need for pairwise enumeration of distances between instances, as well as the need to recalculate the belonging of instances to clusters.

Crisp methods of cluster analysis [1–17, 27–29] are obviously more accurate (provide a specific result), but at the same time coarse and less adaptive. Fuzzy methods [18–26] give a fuzzy assessment of the membership of an instance to a cluster and are less accurate (specific in the assessment of membership), but at the same time they are more adaptive compared to crisp methods, but also more expensive in terms of the amount of calculations and the required memory.

Depending on the method of forming a partition into clusters, cluster analysis methods can be divided into non-hierarchical [1–17], in which clusters are not subordinated, and hierarchical [27–29], in which partitioning is carried out sequentially by forming nested clusters. In fact, non-hierarchical methods implement breadth-first search, while hierarchical methods implement depth-first search.

It should also be noted that most of the known methods of cluster analysis are dependent on a set of features specified by the user and do not allow one to evaluate their significance. This leads to an excessive partition, an increase in the number of calculations, and also reduces the possibility for the perception of the resulting partition by a person.

Also, if a set of features contains interrelated, duplicate, similar features, or features that are discrete of a time-distributed value, traditional cluster analysis methods will generate an extremely complex, redundant, and uninterpretable partition. However, they will not be able to identify such signs and eliminate or use them more effectively.

Therefore, there is a need to eliminate the shortcomings of crisp and fuzzy methods by developing a fuzzy clustering method taking into account crisp partitioning and search acceleration heuristics.

3 MATERIALS AND METHODS

Unlike most cluster analysis methods [1–29], which involve calculating the distances between all instances in the feature space, it is proposed to calculate the hash distance from it to the conditional common center of coordinates for each instance, replacing the N-dimensional instance coordinate vector with one coordinate, and then determine the distance between instances in one-dimensional space. This will allow for large samples to load into memory only individual instances (minimally – one by one in turn), reducing the amount of calculations and the minimum amount of memory required.

Also, unlike the traditional methods of cluster analysis [1–29], it is proposed to consider the obtained hash feature [30–49] as a pseudo-output feature, dividing it into intervals, which can be compared with labels of pseudo-classes – clusters. This will allow replacing the enumeration of pairs of compared feature distances with an ordered set of one-dimensional coordinates of instances along the hash axis, thus reducing the amount of calculations.

Further, having received a rough crisp splitting of the sample instances, it is proposed for them to set the splitting of the input features into fuzzy terms, to determine on their basis and splitting the instances the rules for referring instances to clusters.

In contrast to the traditional metric approach to cluster analysis [1–17], which involves the use of the entire set of initial features, it is proposed to evaluate the informativity of features [50, 51] and fuzzy terms [52, 53] and exclude non-informative terms, as well as non-informative features, while maintaining an acceptable level of quality criterion.

Than we may determine the fuzzy inference system of the Mamdani-Zadeh classifier type, which in the form of a neuro-fuzzy network can be further trained by means of optimization methods [54–56] to adjust the parameters of membership functions to fuzzy terms and weights of rules that provide acceptable values of the clustering quality functional.

The above mentioned feature and term reduction will reduce the complexity of calculations, reduce the amount of memory, the complexity of the neuro-fuzzy system, reduce the number of configurable network parameters and, as a result, increase its level of data generalization, as well as interpretability.

Formally, a method for constructing a NFN for data cluster analysis that implements the ideas described above can be represented as follows.

The initialization stage. Specify a sample of observations $x=\{x^s\}$ and used defined values $\varepsilon \geq 0$, $0 < \alpha \leq 1$, and $0 < \beta \leq 1$.

The hash calculation stage. Using one of the hash calculation methods [30–49] determine the hashes $\{x_*^s\}$ for the sample instances. Order sample instances along the hash value axis x_*^s.

The stage of a crisp division of the feature space. Split the range of hash values x_*^s into intervals, numbering them sequentially. For each sample instance x^s, fix the number of the interval $q^s{}_*$, in which it fell according to the hash value x_*^s, as the label of the output feature – the pseudo-class (cluster) $y^s = q^s{}_*$.

DOI 10.15588/1607-3274-2022-4-6

A crisp division of the ranges of feature values into intervals (selection of terms) can be done in various ways.

The simplest way is to divide the range of values of each feature into an equal number of intervals that have the same length for the corresponding feature. With such a partition, the space of input features will be divided by a uniform grid, the parameters of the intervals of which are easy to determine the number of intervals into which the range of values of the j-th feature is divided N_j ($N_j \geq 2$). It is also reasonable to provide the restriction $N_j \geq K$. It is also desirable that $N_j << S$. For a given number of intervals N_j define $L_{j,q}$ the length of the q-th interval of values of the j-th feature:

$$L_{j,q} = \frac{x_j^{\max} - x_j^{\min}}{N_j},$$

$$x_j^{\min} = \min_{s=1,2,...,S} \{x_j^s\},$$

$$x_j^{\max} = \max_{s=1,2,...,S} \{x_j^s\},$$

on the basis of which we calculate the left $l_{j,q}$ and right $r_{j,q}$ boundaries of the intervals:

$$l_{j,q} = x_j^{\min} + (q-1)L_{j,q},$$

$$r_{j,q} = x_j^{\min} + qL_{j,q}.$$

For the s-th instance, the number of the interval q, in which it falls according to the j-th feature, we determine as:

$$q_j^s = 1 + \left\lfloor \frac{x_j^s - x_j^{\min}}{L_{j,1}} \right\rfloor$$

or as

$$q_j^s = \{q \mid l_{j,q} \leq x_j^s \leq r_{j,q}, q = 1,2,..., N_j\}.$$

Since the dimension of the sample is $n = NS$, then to ensure the generalizing properties of the model, it is important that $3N_jN \leq n$, that is, $3N_j \leq S$. As a result, we can recommend setting: $K \leq N_j \leq S/3$. If $S/3$ is less than K, then set $N_j = K$.

In the cells of such a grid, in the general case, instances of different pseudo-classes will fall, since the hash feature is not taken into account. Also unknown is the number of intervals into which it is necessary to divide the ranges of feature values in order to achieve an acceptable accuracy of approximation of the cluster boundaries. This partition will be computationally the fastest and simplest, but it will contain an uncertainty in the choice of the number of intervals, and it will also not allow us to accurately select clusters.

It is more difficult to divide the range of feature values into intervals, when a different number of intervals are allocated on the axis of each feature and the pseudo-class number determined by the hash feature [30–49] is taken into account. To do this, we need to project labels of instances (pseudo-class numbers) one by one onto the axis of the j-th feature in ascending order of its values.

In this case, a situation may arise when for the same value of the coordinate along the axis of the j-th feature, there are several instances with different labels of pseudo-classes. In this case, instances with a label equal to the label of the instance with a lower coordinate preceding the group should be placed first, and instances with a label equal to the label of the instance with a larger coordinate following the group should be placed last. After that, it is necessary to select intervals of values of the j-th feature $\{<l_{j,q}, r_{j,q}>\}$ such that within one interval there are instances with the same value of the hash pseudoclass number, and instances of adjacent intervals of feature values have different hash pseudoclass numbers. Here the situations are possible when adjacent intervals can touch and partially overlap if the left and right boundaries of adjacent intervals have the same coordinates. It is also possible that there will be voids between adjacent intervals in which there are no instances.

After the formation of such a partition, the number of intervals into which the range of values of the feature is divided N_j can be used to determine the information content of the features.

The more intervals of changing the class number the range of the feature is divided into, the more complex and non-linear the classification is, i.e. the lower the individual informativity of this feature. The fewer intervals of feature values (ideally, one) correspond to a specific pseudo-class number, the more valuable this interval is for this pseudo-class. This can be quantified by the indicator of individual informativity of the j-th feature:

$$I_j = \frac{K}{N_j}, N_j \geq K.$$

This indicator will tend to zero with a lower individual information content of the feature and to one – with a higher one.

The rule formation stage. Convert each instance of the sample into a crisp rule of the form:

$$(s): \text{if } \bigcup_{j=1}^{N} \left\{ \bigcap_{q=1}^{N_j} x_j^s \in [l_{j,q}, r_{j,q}] \right\},$$

$$\text{then } y^s = q^s{}_* \text{ with a weight } w^s = 1.$$

Here (s) is a rule number.

To simplify program processing, such rules can be represented as a set R: $\{(s): \{q^s{}_j\} \rightarrow q^s{}_*, w^s\}$.

The stage of assessing the quality of a partition and a set of rules. To assess the quality of the generated partition and set of rules, it is possible to use the classification error E.

DOI 10.15588/1607-3274-2022-4-6

To do this, for each s-th instance of the sample x^s, $s=1, 2, ..., S$:

– determine its belonging to each term $\{q^s_j\}$;

– from the set of rules R select those rules that correspond to the recognized instance on the left side (form a conflict set of rules R_c and estimate its power (the number of rules in the conflict set $|R_c|$));

– define as the calculated class number for the recognized instance x^s and the conflict set R_c the value:

$$y^s_* = \arg \max_{q=1,2,..,K} \left\{ \sum_{p=1}^{|R_c|} w^p \right\}.$$

For a sample of recognized instances, the classification error can be estimated as:

$$E = \sum_{s=1}^{S} \{1 \mid y^s \neq y^s_*\}.$$

If the error value is unacceptable ($E > \varepsilon$), then it is possible to revise the generated partition by increasing the number of intervals into which the feature value ranges are divided and / or change the hash calculation method.

The rule reduction stage. All rules should be sorted by the value $y=\{q^s_*\}$, then by the values $\{q^s_j\}$. Set $S' = S$. Looking through the rules sequentially $s= 1, 2, ..., S'–1$: for two rules (s) and (s+1) consecutive in y, if their right parts are the same ($q^s_* = q^p_*$) and the left parts are the same ($\forall j = 1,2,....,N : q^s_j = q^{s+1}_j$), then keep the first ($s$-th) rule, increasing its weight by the weight of the (s+1)-th rule: $w^s=w^s+ w^{s+1}$, then remove the second ((s+1)-th) rule, and decrease S': $S'= S'–1$.

The stage of reduction of terms and features. For each term of each input feature, using the set of generated rules for each k-th pseudo-class, determine the number of times the term was used in the rules, taking into account their weights:

$$N_{j,q,k} = \sum_{s=1}^{S'} \{w^s \mid q^s_j = q, q^s_* = k\},$$

$$N_{j,q} = \sum_{s=1}^{S'} \{w^s \mid q^s_j = q\}.$$

The greater the value of $N_{j,q,k}$,, the more strongly the q-th term of the j-th feature is involved in making decisions about assigning an instance to the k-th pseudoclass.

Let's define the indicator of the individual informativity of the term for the entire set of classes:

$$I_{j,q} = \frac{\max\limits_{k=1,2,...,K} \{N_{k,j,q}\}}{N_j}.$$

This indicator will take values in the range from zero to one. The smaller its value, the less informative is the q-th term of the j-th feature. The greater its value, the more significant is the q-th term of the j-th feature.

In ascending order, the estimates of the individual informativities of features I_j are sequentially for the current considered j-th feature:

– exclude it (all its terms) from all rules;

– estimate the error in determining the class number for sample instances according to the current set of rules and terms E;

– if the error E is acceptable ($E \le \varepsilon$), then consider the j-th feature as non-informative and remove it from further consideration, and also remove its terms and their membership functions;

– if the error E is unacceptable ($E > \varepsilon$), then return the deleted feature and terms to the rules and stop further revision of the features.

Looking through the terms of the features in order from the least frequently used in the rules to the most frequently used (i.e. in ascending order of the value of $I_{j,q}$), sequentially for each term:

– exclude it from all rules;

– estimate the error in determining the class number for sample instances according to the current set of rules and terms E;

– if the error is acceptable ($E \le \varepsilon$), then consider the q-th term of the j-th feature as non-informative and remove it and its membership function from further consideration.

– if the error is unacceptable ($E > \varepsilon$), then return the deleted term to the rules and stop further revision of the terms.

The stage of identifying similarities and reducing similar features. For i=1, 2, ..., N, $j = i$+1, i+2,...,N, determine the estimate of the pairwise relationship of the i-th and j-th input features $I(i,j) = I(j,i)$. It is possible to implement this on the basis of indicators of individual informativity of features [48–51], meaning the input feature is the i-th feature, and the output feature is the j-th feature.

Based on pairwise estimates $\{I(i,j)\}$, determine the average estimate of the relationship of features:

$$\bar{I} = \frac{1}{0.5N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} I(i,j).$$

Divide features into groups, such that the features of one group have an estimate of the relationship greater than the average, multiplied by a user-specified coefficient α, $0 < \alpha \le 1$. To do this, first enter into the set of unconsidered features all the features that are present in the rules of the current set of rules. Then, while the set of unconsidered features is non-empty, repeat:

– choose from the set of individually unconsidered the most informative feature (in relation to the output feature) and form a new group of features for it;

– from the set of unconsidered features, select all the features that are related with the feature of the new group stronger than the average relation, taking into account the coefficient α: $I(i,j) \ge \alpha \bar{I}$, transfer them to the group of this feature.

DOI 10.15588/1607-3274-2022-4-6

From the features of each group containing two or more features:

– leave in the rules only one feature that is most closely related to the output feature (has the highest estimate of individual informativity I_j).

– estimate the classification error E for the current set of rules and terms;

– if the error is acceptable ($E \le \varepsilon$), then remove the excluded features and their terms from further consideration, otherwise, return all the features of this group.

An alternative option is to sequentially remove from each group individually the least significant feature (with a lower value of I_j) until the error remains acceptable and the number of remaining features in the group is at least one.

Also, in a number of problems where it is assumed that a number of input features are indirect observations of a hidden factor or some of the input features are discrete samples of a distributed signal, then instead of or in addition to this stage, it is possible to include not the removal, but the combination of the primary features into the calculated one. In this case, after selecting groups of features for each group, based on all its features, it is necessary to calculate the values of artificial convolution features (in the simplest case, this can be the sum, average value, maximum, minimum, and the product of the values of the primary features of the group), and then evaluate for each convolution its connection with the output feature. If not a single convolution exceeds the individual informativity of the primary features of the group in terms of the value of individual informativity, then for this group it should be limited to choosing one primary feature with the highest individual informativity, otherwise all primary features should be excluded from the group, replacing them into an artificial convolution feature, determine the terms and their parameters for this feature, adjusting the generated partition and its parameters, as well as the set of rules.

The stage of revealing the similarity and reduction of similar terms of different features. Determine the pairwise equivalence estimate for the terms of different input features:

$$I_{i,p,j,q} = I_{j,q,i,p} = \frac{\sum_{s=1}^{S} \{1 \mid l_{i,p} \le x_i^s \le r_{i,p}, l_{j,q} \le x_j^s \le r_{j,q}\}}{\max\{N_{i,p}, N_{j,q}\}},$$
$$i = 1, 2, ..., N,\ j = i+1, i+2, ..., N,$$
$$p = 1, 2, ..., N_i, q = 1, 2, ..., N_j.$$

Determine the average estimate of the relationship between the terms of various input features:

$$\bar{I}^t = \frac{\sum_{i=1}^{N} \sum_{j=i+1}^{N} \sum_{p=1}^{N_i} \sum_{q=1}^{N_j} I_{i,p,j,q}}{0{,}5N(N-1)\left(\sum_{i=1}^{N} N_i\right)^2}.$$

Divide the terms into groups, such that the terms of different features of the same group have an estimate of the relationship greater than the average multiplied by a user-specified coefficient β, $0 < \beta \le 1$.

To do this, first enter into the set of unconsidered terms all the terms that are present in the rules of the current set of rules. Then, while the set of unconsidered terms is non-empty, repeat:

– choose from the set of individually unconsidered the most informative term and form a new group of terms for it;

– from the set of unconsidered terms excluding other terms of the feature same as feature of a group forming term, select all the terms that are related with the term of the new group stronger than the average relation, taking into account the coefficient β: $I_{i,p,j,q} \ge \beta \bar{I}^t$, transfer them to the group of this term.

From the terms of each group containing two or more terms:

– leave in the rules only one term, the feature of which is most closely related to the output feature (it has the biggest value of individual informativity I_j);

– estimate the classification error E for the current set of rules and terms;

– if the error is acceptable ($E \le \varepsilon$), then remove the excluded terms from further consideration, otherwise, return all the terms of this group.

An alternative option is to sequentially remove from each group individually the least significant term (with a lower value of I_j) until the error remains acceptable and the number of remaining terms in the group is at least one.

The stage of fuzzy terms formation. On the basis of the parameters of the intervals of the values of the features selected during the formation of a crisp partition and the terms and features selected in the process of reduction, it is possible to determine the membership functions for fuzzy terms. For this it is possible to use different types of elementary membership functions [21, 22]. For crisp intervals in which two or more instances fell into, it is proposed to use the following functions: trapezoidal, bell-shaped, Gaussian, Π-shaped. For point intervals, where only one instance fell, it is proposed to use the following functions: triangular, bell-shaped, Gaussian, Π-shaped function. Each of these functions $\mu_{j,q}(x_j^s)$ for a specific fuzzy term (the q-th interval of values on the axis of the j-th feature) will have adjustable parameters $< a_{j,q}, b_{j,q}, c_{j,q}, d_{j,q} >$, such that: $a_{j,q} \le b_{j,q} \le c_{j,q} \le d_{j,q}$. The values of the parameters of the membership functions can be determined based on the parameters of a crisp partition.

For example, for trapezoidal and Π-shaped functions, the parameters can be defined as:

– for splitting into intervals equal in length:

$$a_{j,q} = b_{j,q} = l_{j,q},$$

DOI 10.15588/1607-3274-2022-4-6

$$c_{j,q} = d_{j,q} = r_{j,q};$$

– for splitting into intervals of different classes:

$$a_{j,q} = \begin{cases} l_{j,q}, q = 1; \\ \dfrac{r_{j,q-1} + l_{j,q}}{2}, q > 1; \end{cases}$$

$$b_{j,q} = l_{j,q},$$

$$c_{j,q} = r_{j,q},$$

$$d_{j,q} = d_{j,q} = \begin{cases} r_{j,q}, q = N_j; \\ \dfrac{r_{j,q} + l_{j,q+1}}{2}, q < N_j. \end{cases}$$

The stage of NFN formation for clustering. Map the generated knowledge base into a fuzzy logical inference system, which is conveniently represented in the neural network basis as a NFN.

The network structure can be determined based on the Mamdani-Zadeh approximator [57]. The nodes of the input layer of the network will correspond to the input features x_j for the recognized instance $x^s = \{x_j^s\}$. Thus, the input layer will have N nodes (hereinafter, we mean not the initial values of the number of features and terms, but after reduction).

The nodes of the first hidden layer of the network will correspond to fuzzification blocks, i.e. will determine the values of membership functions for terms of input features $\mu_{j,q}(x_j^s)$. On the first hidden layer there will be $\sum_{j=1}^{N} N_j$ nodes. The input of each node of the first hidden layer receives a value from the output of only the input layer node corresponding to its feature.

The nodes of the subsequent second hidden layer will combine the membership functions of terms into the membership functions of the antecedents (left parts) of the rules, combining the outputs of the nodes of the first layer, the corresponding terms of which are included in the corresponding antecedents. The second hidden layer will have S' fuzzy "AND" nodes.

The nodes of the third layer will combine the rules into pseudo-classes, implementing a fuzzy "OR". The third hidden layer will have K nodes.

The single node of the output layer will defuzzify the result, giving the number of the cluster (pseudo-class) according to the formula:

$$y^s = \left\lfloor \frac{\sum_{k=1}^{K} k\mu_k(x^s)}{\sum_{k=1}^{K} \mu_k(x^s)} \right\rfloor \text{ or}$$

$$y^s = \arg \max_{k=1,2,...,K} \{\mu_k(x^s)\}.$$

The network parameters will be determined on the basis of the previously formed crisp partition and set of rules.

The stage of additional training and optimization of the neuro-fuzzy clusterizer. Let evaluate the performance quality of a NFN (the quality of data clustering) based on a given functional F, which can be determined based on a wide class of metrics [1, 2, 7, 8, 12–15]. Using the methods of evolutionary optimization [54, 55], we can select such values of the network term parameters that will improve the value of the optimized functional F. The final model will be a neuro-fuzzy clusterizer optimized by the number of features used and the functional F.

4 EXPERIMENTS

To study the practical applicability of the proposed method, it was implemented in software and used to solve a set of practical problems of different nature and dimension. The characteristics of the initial data samples for practical tasks are given in Table 1.

Table 1 – Characteristics of initial samples for cluster analysis

Task name	Task acronym	Source	N	S	n
Low Resolution Spectrometer	LRS	https://archive.ics.uci.edu/ml/datasets/Low+Resolution+Spectrometer	102	531	54162
Musk (Version 2)	Mv2	https://archive.ics.uci.edu/ml/datasets/Musk+%28Version+2%29	168	6598	1108464
Urban Land Cover	ULC	https://archive.ics.uci.edu/ml/datasets/Urban+Land+Cover	148	168	24864
Iris	IRIS	https://archive.ics.uci.edu/ml/datasets/Iris	4	150	600
Heart Disease	HD	https://archive.ics.uci.edu/ml/datasets/Heart+Disease	75	303	22725
Breast Cancer Wisconsin (Diagnostic)	BCWD	https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29	32	569	18208
Arrhythmia	ART	https://archive.ics.uci.edu/ml/datasets/Arrhythmia	279	452	126108
Crop mapping using fused optical-radar	CMFOR	https://archive.ics.uci.edu/ml/datasets/Crop+mapping+using+fused+optical-radar+data+set	175	325834	57020950
Sensorless Drive Diagnosis	SDD	https://archive.ics.uci.edu/ml/datasets/Dataset+for+Sensorless+Drive+Diagnosis	49	58509	2866941

For each task in the experiments, various hash generation methods and various values of the parameters that regulate the operation of the proposed method were used.

To evaluate the results of the cluster analysis methods, we used the error E, the F value for the resulting model, the

DOI 10.15588/1607-3274-2022-4-6

running time of the method *t*, the memory size required by the method *m*, the number of parameters of the clustering model N_w, and the number of features after reduction *N*'. It is also suggested that, in addition to the characteristics described above, the following metrics to be used.

The generalizing properties of the resulting clustering models compared to the original data dimension can be characterized by the generalization coefficient:

$$I_G = \frac{NS}{N_w} = \frac{n}{N_w}, N_w \geq 1.$$

This coefficient will take a value in the range from zero to *n*. The more parameters the model has, the lower the level of generalization it has relative to the dimension of the initial data, the lower the value of the generalization coefficient.

Alternatively, the generalization may be characterized at the equal number of instances as the ratio of the number of features in the primary set *N* to the number of features used in the reduced set of the final model, *N*':

$$I_{GF} = \frac{N}{N'}, N \geq N' \geq 1.$$

This coefficient will take a value in the range from zero to *n*. The more parameters the model has, the lower the level of generalization it has relative to the dimension of the initial data, the lower the value of the generalization coefficient.

When comparing pairwise the resulting models 1 and 2 for the same initial data sample, their generalization with acceptable values of the criterion *F* may be characterized by the relations:

$$G_{1,2} = \frac{I_{G1}}{I_{G2}} = \frac{NS}{N_{w1}} \frac{N_{w2}}{NS} = \frac{N_{w2}}{N_{w1}}, N_{w1} \geq 1, N_{w2} \geq 1, N \geq 1, S \geq 1.$$

$$GF_{1,2} = \frac{I_{GF1}}{I_{GF2}} = \frac{N}{N_1'} \frac{N_2'}{N} = \frac{N_2'}{N_1'}, N \geq 1, N_1' \geq 1, N_2' \geq 1.$$

For two models we also can define:

$$t_{1,2} = t_1 / t_2,$$

$$m_{1,2} = m_1 / m_2,$$

$$F_{1,2} = F_1 / F_2.$$

5 RESULTS

Table 2 presents the values of the indicators $G_{1,2}$, $GF_{1,2}$, $t_{1,2}$, $m_{1,2}$, and $F_{1,2}$ to compare the proposed method with the fuzzy c-means method [24–26] , in which the initial partition is formed randomly.

Table 2 – Resulting values of indicators for clustering model comparison

Task acronym	$GF_{1,2}$	$G_{1,2}$	$t_{1,2}$	$m_{1,2}$	$F_{1,2}$
LRS	1.3	1.5	0.8	0.7	0.9
Mv2	1.4	2.4	0.9	0.4	1.1
ULC	1.5	1.7	0.9	0.6	0.9
IRIS	1.3	1.4	1	0.8	1
HD	1.2	1.5	0.9	0.6	1.1
BCWD	1.1	1.7	1.1	0.6	0.9
ART	1.3	2.2	0.9	0.5	0.9
CMFOR	1.2	1.9	0.8	0.5	1.1
SDD	1.2	1.7	0.9	0.6	0.9

It is easy to see from the Table 1 and Table 2, that the proposed method allows for the same data sample to significantly improve the generalizing properties of the model, to reduce time and computer memory costs, and also provide a better or acceptable value of the quality functional. This is explained by the fact that the proposed method non-randomly generates a partition of the feature space, selects and reduces non-informative terms and features, seeking to reduce the complexity of the model. At the same time, the proposed method does not require the calculation of distances between instances due to the use of a locally sensitive hash.

The generalized dependences between key characteristics of the proposed method obtained in experiments are schematically shown in Fig. 1–Fig. 9.

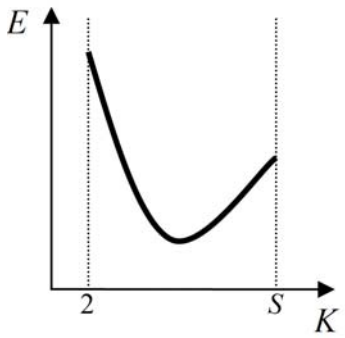

Figure 1 – Schematic graph of the averaged dependence *E* from *K*

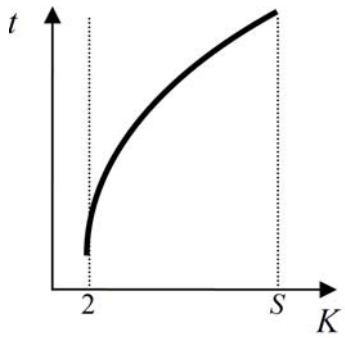

Figure 2 – Schematic graph of the averaged dependence *t* from *K*

DOI 10.15588/1607-3274-2022-4-6

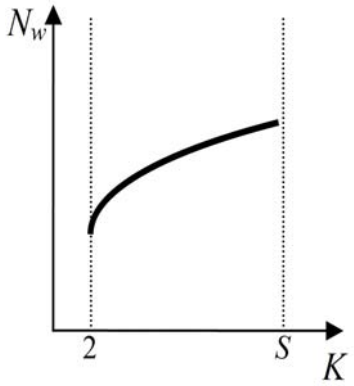

Figure 3 – Schematic graph of the averaged dependence N_w from K

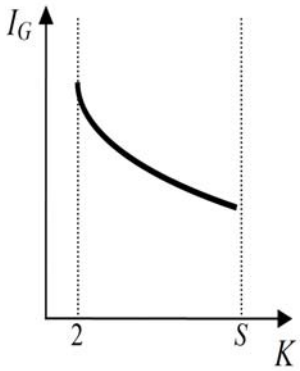

Figure 4 – Schematic graph of the averaged dependence I_G from K

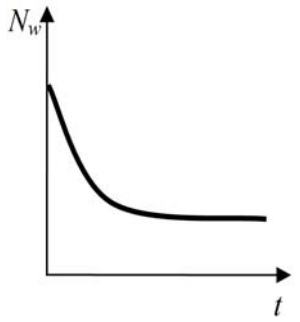

Figure 5 – Schematic graph of the averaged dependence N_w from t

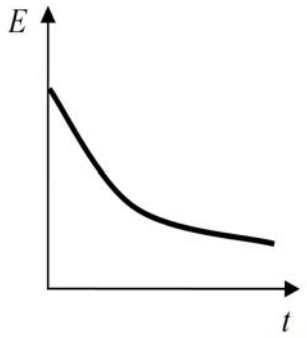

Figure 6 – Schematic graph of the averaged dependence E from t

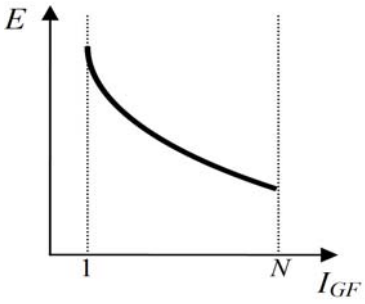

Figure 7 – Schematic graph of the averaged dependence E from I_{GF}

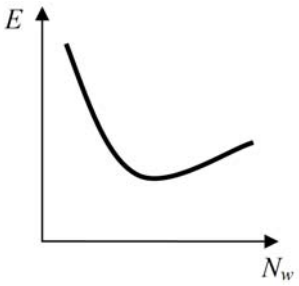

Figure 8 – Schematic graph of the averaged dependence E from N_w

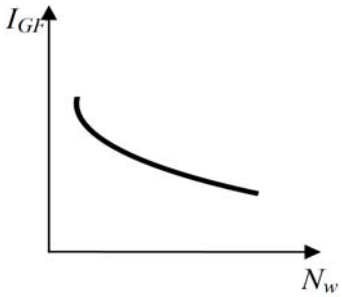

Figure 9 – Schematic graph of the averaged dependence I_{GF} from N_w

As can be seen from the Fig. 1, with an increase in the value of K, the value of the error E decreases to a certain level, after which it begins to grow. The decrease in the error E with increasing K is explained by the increase in the accuracy of approximating the boundaries of clusters due to the refinement of the partition of the output feature. An increase in the error E with a further increase in the value of K is explained by an increase in the uncertainty in the approximation of the boundaries of clusters with an excessively detailed partition of the output feature due to the selection of a large number of small clusters. Therefore, it is recommended to iteratively select such a value of K in the process of executing the method, when using a uniform partition of the feature space, at which the smallest error value will be achieved. The use of non-uniform partitioning into intervals with different class numbers makes it possible to automate this process and avoid the growth of the error. Generally, with an increase in the number of clusters, the accuracy of the partition increases,

DOI 10.15588/1607-3274-2022-4-6

but the cost of calculations and memory requirements also increase.

As can be seen from the Fig. 2 and Fig. 3, with an increase in the number of classes K, the running time of the method t and the number of parameters of the resulting model N_w are increased significantly. This is explained by the fact that with an increase in the number of pseudo-classes (pseudo-clusters) K, the number of selected terms and their parameters will increase, which will require a significant amount of time to calculate their information content indicators and reduction, as well as optimization tuning of the increased number parameters of the NFN model.

As can be seen from the Fig. 4, with an increase in the number of pseudo-clusters K, a decrease in the generalization indicator of the model is observed. This is explained by a significant increase in the number of model parameters due to a more detailed approximation of the cluster boundaries due to an increase in their number.

As can be seen from the Fig. 5, with an increase in the time spent in the process of the method's work t, the number of model parameters N_w is reduced. This is explained by the fact that the number of features and terms is reduced due to the removal of non-informative and duplicate features and terms. The more iterations will be in the process of identifying and reducing non-informative terms and features, the more time will be spent by the method, but the smaller will be the number of parameters of the resulting model.

As can be seen from the Fig. 6, with an increase in the time of the method t, a decrease in the model error E is observed. This is explained by the fact that the adjustment of the model parameters makes it possible to increase the accuracy (reduce the error) of the model. Also, the increase in time costs can be explained by an iterative search for the optimal partition of the feature space, which will eventually lead to a decrease in the error of the resulting model.

As can be seen from the Fig. 7, with an increase in the value of the I_{GF} indicator, a decrease in the error of the model E is observed. This is explained by the fact that with a very high generalization, the number of features used will be less and, accordingly, the approximation of the partition of the feature space will be rougher, which will lead to an increase in the error E. With the lowest value of the feature generalization index, more features will be used and the feature space will be split in more detail, which will reduce the error value.

As can be seen from the Fig. 8, with an increase in the number of model parameters N_w, there is a drop in the error value to a certain value. This is explained by the fact that the detailing of the division of the feature space due to the increase in the number of pseudoclusters makes it possible to more accurately approximate the boundaries of the clusters. The further growth of the error in the process of increasing the value of N_w is explained by the fact that the excessive selection of clusters leads to a lack of generalization, which is gradually reflected in the growth of the error value E. However, this is typical mainly for the uniform partition of the feature space.

As can be seen from the Fig. 9, with an increase in the number of model parameters N_w, a decrease in the generalization index of features I_{GF} is observed. This is explained by the fact that the detailing of the division of the feature space leads to the forming of a larger number of non-informative terms and features, which makes it possible to exclude non-informative features. On the other hand, an increase in the number of model parameters N_w can be explained by an increase in the number of features in the original feature set for the task N, which in turn may indicate a greater proportion of non-informative features, i.e. about the reduction in the number of features.

At the Fig. 10 the schematic graphs of averaged dependencies of E, I_G, t, and N_w from the α and β values are shown.

As it can be seen from the Fig. 10 the bigger the value of α or β the bigger will be values of t and N_w and the lower the E and I_G values. If it is assumed that the features are of a different nature, then the value of the coefficients α and β is recommended to be set the bigger. If it is assumed that the features are ordered readings of a certain value, then the values of the coefficients α and β are recommended to be set smaller.

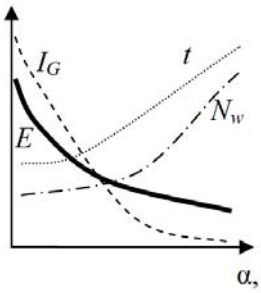

Figure 9 – Schematic graphs of the averaged dependences E, I_G, t, and N_w from the α and β values

6 DISCUSSION

The operability and practical applicability of the proposed method and the developed software were confirmed as a result of the analysis of experimentally obtained data.

The proposed method combines the ideas of crisp and fuzzy cluster analysis. At first, it forms a crisp partition of the feature space, but then, due to fuzzification, it transforms it into a fuzzy partition. The crisp partition is used to automate the selection of the number of clusters, as well as to speed up the selection of feature terms. The proposed method generates a crisp partition automatically (without human intervention), and the proposed method is faster than the traditional iterative (optimization) formation of a fuzzy partition [13, 21–23] , which is inherent in most fuzzy clustering methods.

Unlike the traditionally used metric methods of cluster analysis [1–3], which involve the use of the entire primary set of features in the final model, the proposed method

DOI 10.15588/1607-3274-2022-4-6

selects the minimum subset of features necessary for clustering, thereby reducing the structural and parametric complexity of the model, increasing its generalization. properties and interpretability (explainability), and can also reduce the number of features by combining primary features into artificial calculated ones, thereby further increasing the generalizing properties of the model and its interpretability. In addition, the proposed method is more adaptive due to fuzzification, does not require the initial setting of the number of clusters and the initial splitting of the sample into clusters, as well as the user setting metrics for clusters.

Unlike hierarchical methods of cluster analysis [27–29], which subordinate features and form a splitting hierarchy, which can have a large depth, the proposed method does not subordinate features, but at the same time, removes non-informative features and terms, and the hierarchy of its depth checks does not exceed three levels. At the same time, the proposed method significantly exceeds hierarchical methods in terms of parallelization of calculations, which is achieved due to a smaller depth compared to hierarchical methods. However, the proposed method makes it possible to obtain as an additional result the estimates of the informativity of features and terms, to form artificial features by replacing the original ones, to adaptively adjust the shape and parameters of clusters due to membership functions, and also to automatically generate the number of clusters.

CONCLUSIONS

The problem of multidimensional data cluster analysis is considered. Within this problem the cluster formation speed is increased, the complexity of the clustering model is reduced, and its interpretability is increased.

The scientific novelty of obtained results is that for the first time a method of cluster analysis of multidimensional data is proposed, which for each instance calculates its hash based on the distance to the conditional center of coordinates, uses a one-dimensional coordinate along the hash axis to determine the distances between instances, considers the resulting hash as a pseudo output a feature, dividing it into intervals, to which it compares labels of pseudo-classes-clusters, having received a rough crisp partition of the feature space and sample instances, automatically generates a partition of input features into fuzzy terms, determines the rules for referring instances to clusters and, as a result, forms a fuzzy inference system of the Mamdani-Zadeh classifier, which is retrained in the form of a neuro-fuzzy network to ensure acceptable values of the clustering quality functional. This makes possible to reduce the number of terms and features used, to evaluate their contribution to making decisions about assigning instances to clusters, to increase the speed of data cluster analysis, and to increase the interpretability of the resulting data splitting into clusters.

The practical significance of obtained results is that mathematical support allowing to solve the problem of cluster data analysis in conditions of large data dimensionality has been developed. The experiments confirmed the operability of the developed software have been carried out. They allow to recommend it for use in practice in problems of data analysis of various nature and dimensions.

The prospects for further research are to study the application of the proposed method on a wide range of practical problems of various dimensions and nature, to study the influence of various metrics on the results of the method (accuracy and speed of building NFN, computational complexity), to develop a parallel implementation of the method, to study questions of method integration with evolutionary and multi-agent search methods.

DOI 10.15588/1607-3274-2022-4-6

ACKNOWLEDGEMENTS

The work is supported by the state budget scientific research project “Intellectual methods for diagnosing and prediction the state of the complex objects” (state registration number 0122U000972) of the National University “Zaporizhzhia Polytechnic” with partial information support of international projects “Cross-domain competences for healthy and safe work in the 21st century” (WORK4CE, Project Reference: 619034-EPP-1-2020-1-UA-EPPKA2-CBHE-JP) of the Erasmus+ program of the European Union and “EuroPIM Virtual Master School Ukraine” (EU-ViMUk) of the program “Ukraine digital: Ensuring study success in times of crisis (2022)” of the German Academic Exchange Service DAAD.

REFERENCES

1. Everitt B., Landau S., Morven L. et al. Cluster analysis. Chichester, Wiley, 2011, 330 p.
2. Aggarwal C., Reddy C., Chandan K.. eds. Data Clustering : Algorithms and Applications. New York, Chapman and Hall/CRC, 2016, 652 p.
3. Huang Z. Extensions to the k-means algorithm for clustering large data sets with categorical values, *Data Mining and Knowledge Discovery,* 1998, Vol. 2, Issue 3, pp. 283–304. DOI:10.1023/A:1009769707641.S2CID 11323096.
4. Ng R., Han J. Efficient and effective clustering method for spatial data mining, *20th International Conference on Very Large Data Bases (VLDB'94), September 12–15, 1994, Santiago, Chile, proceedings*. Burlington, Morgan Kaufmann, 1994, pp. 144–155.
5. Bailey K. D. Typologies and Taxonomies: An Introduction to Classification Techniques. London, Sage Publications, 1994, 96 p.
6. Gordon A.D. Classification. Boca Raton, Chapman & Hall/CRC, 1999, 256 p.
7. Romesburg C. H. Cluster Analysis for Researchers. Belmont, Lifetime Learning Publications, 1984, 334 p.
8. Aldenderfer M. S., Blashfield R. K. Cluster Analysis. London, Sage Publications, 1984, 88 p.
9. Meilă, M. Comparing Clusterings by the Variation of Information, *Lecture Notes in Computer Science*, 2003, Vol. 2777, pp. 173–187. DOI:10.1007/978-3-540-45167-9_14.
10. Kraskov A., Stögbauer H., Andrzejak R. G., Grassberger P. Peter Hierarchical Clustering Based on Mutual Information, [Electronic resource]. Access mode: https://arxiv.org/abs/q-bio/0311039.

11. Frey B. J., Dueck D. Clustering by Passing Messages Between Data Points, *Science,* 2007,Vol. 315, № 5814, pp. 972–976. DOI: 10.1126/science.1136800.
12. Pfitzner D., Leibbrandt R., Powers D. Characterization and evaluation of similarity measures for pairs of clusterings, *Knowledge and Information Systems*, 2009, Vol. 19, № 3, pp. 361–394. DOI:10.1007/s10115-008-0150-6.
13. Dunn J. Well separated clusters and optimal fuzzy partitions, *Journal of Cybernetics*, 1974, № 4, pp. 95–104. DOI:10.1080/01969727408546059.
14. Rand W. M. Objective criteria for the evaluation of clustering methods, *Journal of the American Statistical Association*, 1971, Vol. 66 (336), pp. 846–850. DOI:10.2307/2284239.
15. Hubert L., Arabie P. Comparing partitions, *Journal of Classification*, 1985, Vol. 2, pp. 193–218. DOI:10.1007/BF01908075.S2CID189915041
16. .Di Marco A., Navigli R. Clustering and Diversifying Web Search Results with Graph-Based Word Sense Induction, *Computational Linguistics,* 2013, Vol. 39, № 3, pp. 709–754. DOI:10.1162/COLI_a_00148. S2CID 1775181.
17. Arnott R. D. Cluster Analysis and Stock Price Comovement, *Financial Analysts Journal*,1980,Vol. 36, № 6, pp. 56–62. DOI: 10.2469/faj.v36.n6.56. ISSN 0015-198X.
18. Dunn J. C. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters, *Journal of Cybernetics,* 1973, Vol. 3, Issue 3, pp. 32–57. DOI:10.1080/01969727308546046.
19. Ahmed M., Yamany S., Mohamed N., Farag A., Moriarty T. A Modified Fuzzy C-Means Algorithm for Bias Field Estimation and Segmentation of MRI Data, *IEEE Transactions on Medical Imaging*, 2002, Vol. 21, № 3, pp. 193–199. DOI:10.1109/42.996338.
20. Abonyi J., Feil B. Cluster Analysis for Data Mining and System Identification. Berlin, Birkhäuser Verlag, 2007, 306 p. DOI: 10.1007/978-3-7643-7988-9
21. Höppner F., Klawonn F., Kruse R., Runkler T. Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition. Chichester, John Wiley & Sons, 1999, 304 p.
22. Miyamoto S. Fuzzy Sets in Information and Retrieval and Cluster Analysis. Dordrecht, Kluwer Academic Publishers, 1990, 274 p.
23. Banerjee T. Day or Night Activity Recognition From Video Using Fuzzy Clustering Techniques, *IEEE Transactions on Fuzzy Systems*, 2014, Vol. 22, Issue 3, pp. 483–493. DOI: 10.1109/TFUZZ.2013.2260756.
24. Valente de Oliveira J., Pedrycz W. eds. Advances in Fuzzy Clustering and its Applications. Chichester, John Wiley & Sons, 2007, 454 p.
25. Bezdek J. C. Pattern Recognition with Fuzzy Objective Function Algorithms. New York, Plenum Press, 1981, 272 p. DOI: 10.1007/978-1-4757-0450-1
26. Dumitrescu D., Lazzerini B., Jain L. C. Fuzzy Sets and Applications to Clustering and Training. Boca Raton, CRC Press, 2000, 664 p.
27. Achtert E., Böhm C., Kriegel H.-P., Kröger P., Müller-Gorman I., Zimek A. Finding Hierarchies of Subspace Clusters, *Lecture Notes in Computer Science,* 2006, Vol. 4213, pp. 446–453. DOI:10.1007/11871637_42. ISBN 978-3-540-45374-1.
28. Achtert E., Böhm C., Kriegel H. P., Kröger P., Müller-Gorman I., Zimek A. Detection and Visualization of Subspace Cluster Hierarchies, *Lecture Notes in Computer Science*, 2007, Vol. 4443, pp. 152–163. DOI:10.1007/978-3-540-71703-4_15.
29. Johnson S. C. Hierarchical clustering schemes, Psychometrika, 1967, Vol. 32, № 3, pp. 241–254. DOI:10.1007/BF02289588
30. Jafari O., Maurya P., Nagarkar P., Islam K. M., Crushev C. A Survey on Locality Sensitive Hashing Algorithms and their Applications [Electronic resource]. Access mode: https://arxiv.org/pdf/2102.08942
31. Buhler J. Efficient large-scale sequence comparison by locality-sensitive hashing, *Bioinformatics,* 2001, Vol. 17, № 5, pp. 419–428.
32. Zhao K., Lu H., Mei J. Locality Preserving Hashing, *Twenty-Eighth AAAI Conference on Artificial Intelligence, 27–31 July 2014, Québec, proceedings.* Palo Alto, AAAI Press, 2014, pp. 2874–2880.
33. Tsai Y.-H., Yang M.-H. Locality preserving hashing, *2014 IEEE International Conference on Image Processing (ICIP), Paris, 27–30 of October 2014, proceedings.* Los Alamitos, IEEE, 2014, pp. 2988–2992. DOI: 10.1109/ICIP.2014.7025604.
34. Weinberger K., Dasgupta A., Langford J., et al. Feature Hashing for Large Scale Multitask Learning, *26th Annual International Conference on Machine Learning (ICML '09) Montreal, June 2009, proceedings*. New York, ACM, 2009, pp. 1113–1120. DOI: 10.1145/1553374.1553516
35. Wolfson H. J., Rigoutsos I. Geometric Hashing: An Overview, *IEEE Computational Sci-ence and Engineering*, 1997, Vol. 4, No 4, pp. 10–21.
36. Fast supervised discrete hashing / [J. Gui, T. Liu, Z. Sunet et al.] // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2017. – Vol. 40, No 2. – P. 490–496. DOI: 10.1109/TPAMI.2017.2678475
37. Aluç, G., Özsu M., Daudjee K. Building self-clustering RDF databases using Tunable-LSH, *The VLDB Journal,* 2018, Vol. 28, № 2, pp. 173–195. DOI:10.1007/s00778-018-0530-9
38. Pauleve L., Jegou H., Amsaleg L. Locality sensitive hashing: A comparison of hash function types and querying mechanisms, *Pattern Recognition Letters*, 2010, Vol. 31, № 11, pp. 1348–1358. DOI:10.1016/j.patrec.2010.04.004.
39. Andoni A., Indyk P. Near-Optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions, *Communications of the ACM*, 2008, Vol. 51, № 1, pp. 117–122. DOI:10.1145/1327452.1327494.
40. Salakhutdinov R., Hinton G. Semantic hashing, *International Journal of Approximate Reasoning,* 2008, Vol. 50, № 7, pp. 969–978. DO:10.1016/j.ijar.2008.11.006.
41. Dahlgaard S., Knudsen M., Thorup M. Fast similarity sketching, *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS), 15–17 October 2017, Berkeley*. Los Alamitos: IEEE, 2017, pp. 663–671. DOI: 10.1109/FOCS.2017.67
42. Chin A. Locality-preserving hash functions for general purpose parallel computation, *Algorithmica,* 1994, Vol. 12, Issue 2–3, pp. 170–181. DOI: 10.1007/BF01185209. S2CID 18108051.
43. Subbotin S. A. Methods and characteristics of locality-preserving transformations in the problems of computational intelligence, *Radio Electronics, Computer Science, Control,* 2014, № 1, pp. 120–128. DOI: 10.15588/1607-3274-2014-1-17
44. Subbotin S. A., Blagodarev A. Yu., Gofman Ye. A. The neuro-fuzzy diagnostic model synthesis with hashed transformation in the sequence and parallel mode, *Radio Elec-*

DOI 10.15588/1607-3274-2022-4-6

tronics, Computer Science, Control, 2017, No. 1, pp. 56–65. DOI: 10.15588/1607-3274-2017-1-7
45. Subbotin S. A. The polar coordinates based hashing for data dimensionality reduction, *Radio Electronics, Computer Science, Control*, 2020, № 4, pp. 118–128. DOI: 10.15588/1607-3274-2020-4-12
46. Subbotin S. A., Oleynik A. A. Analiz preobrazovaniy dlya proyetsi-rovaniya dannykh na obobshchonnuyu os' v zadachakh raspoznavaniya obrazov, *Shtuchniy íntelekt,* 2010, № 1, pp. 114–121.
47. Subbotin S. A. Constructed features for automatic classification of stationary timing signals, *Radio Electronics, Computer Science, Control,* 2012, № 1, pp. 96–103. DOI: 10.15588/1607-3274-2012-1-19
48. Subbotin S. A. The complex data dimensionality reduction for diagnostic and recognition model building on precedents, *Radio Electronics, Computer Science, Control*, 2016, No. 4. pp. 70–76. DOI: 10.15588/1607-3274-2016-4-9
49. Subbotin S. A. Evaluation of informativity and selection of instances based on hashing, *Radio Electronics, Computer Science, Control,* 2020, № 3, pp. 129–137. DOI: 10.15588/1607-3274-2020-3-12
50. Oliinyk A., Subbotin S., Lovkin V., Blagodariov O., Zaiko T. The system of criteria for feature informativeness estimation in pattern recognition, *Radio Electronics, Computer Science, Control*, 2017, № 4, pp. 85–96. DOI: 10.15588/1607-3274-2017-4-10
51. Subbotin S. eds.: Bris R., Majernik J., Pancerz K., Zaitseva E. The instance and feature selection for neural network based diagnosis of chronic obstructive bronchitis, *Applications of Computational Intelligence in Biomedical Technology*. Cham, Springer, 2016, pp. 215–228. (Studies in Computational Intelligence, Vol. 606).
52. 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, № 2, pp. 97–103.
53. Subbotin S., Oliinyk A. The Fully-Defined Neuro-Fuzzy Model Synthesis, *Data Stream Mining & Processing (DSMP), 2016 IEEE First International Conference,* Lviv, 23–27 August 20016, proceedings. Lviv: NU "Lvivska Politeckhnika", 2016. – P. 9–14.
54. Oliinyk A. O., Skrupsky S. Yu., Subbotin S. A. Using parallel random search to train fuzzy neural networks, *Automatic Control and Computer Sciences,* 2014, Vol. 48, №. 6, pp. 313–323.
55. Subbotin S. The method of a structural-parametric synthesis of neuro-fuzzy diagnostic model based on the hybrid stochastic search, *The experience of designing and application of CAD systems in microelectronics : XI International conference CADSM–2011, Polyana–Svalyava, 23–25 February 2011, proceedings.* Lviv, NU "Lviv Polytechnic", 2011, pp. 248–249.
56. Subbotin S. A. Building a fully defined neuro-fuzzy network with a regular partition of a feature space based on large sample, *Radio Electronics, Computer Science, Control,* 2016, № 3, pp. 47–53. DOI: 10.15588/1607-3274-2016-3-6
57. Halgamuge S. K. A trainable transparent universal approximator for defuzzification in Mamdani-type neuro-fuzzy controllers, *IEEE Transactions on Fuzzy Systems*, Vol. 6, № 2. pp. 304–314. DOI: 10.1109/91.669031.

Received 10.09.2022.
Accepted 01.11.2022.

УДК 004.93

НЕЙРО-НЕЧІТКА МЕРЕЖА ДЛЯ КЛАСТЕРИЗАЦІЇ ДАНИХ З ХЕШУВАННЯМ ВІДСТАНЕЙ ТА САМОНАВЧАННЯМ

Субботін С. О. – д-р техн. наук, професор, завідувач кафедри програмних засобів Національного університету «Запорізька політехніка», Запоріжжя, Україна.

АНОТАЦІЯ

Актуальність. Для аналізу даних різної природи та розмірності широко застосовують кластерний аналіз. Однак відомі методи кластер-аналізу характеризуються низькою швидкістю та є вимогливими до ресурсів пам'яті ЕОМ внаслідок необхідності розрахунку попарних відстаней між екземплярами у багатовимірному просторі ознак. Крім того, результати відомих методів кластер-аналізу складні для сприйняття та аналізу людиною при великій кількості ознак.

Мета – підвищення швидкості кластер-аналізу, інтерпретабельності одержуваного розбиття на кластери, а також зниження вимог кластер-аналізу до пам'яті ЕОМ.

Метод. Запропоновано метод кластер-аналізу багатовимірних даних, який для кожного екземпляра обчислює його хеш на основі відстані до умовного центру координат, використовує одновимірну координату по осі хешу для визначення відстаней між екземплярами, розглядає отриманий хеш як псевдовихідну ознаку, розбивши її на інтервали, яким співставляє мітки псевдокласів-кластерів, отримавши грубе чітке розбиття простору ознак і екземплярів вибірки, автоматично формує розбиття вхідних ознак на нечіткі терми, визначає правила віднесення екземплярів до кластерів і в результаті формує систему нечіткого виведення типу класифікатора Мамдані-Заде, який у вигляді нейро-нечіткої мережі донавчається для забезпечення прийнятного значення функціоналу якості кластеризації. Це дозволяє скоротити кількість використовуваних термів і ознак, оцінити їх внесок у прийняття рішень про віднесення екземплярів до кластерів, підвищити швидкість кластер-аналізу даних, а також підвищити інтерпретабельність отримуваного розбиття даних на кластери.

Результати. Розроблено математичне забезпечення, що дозволяє вирішувати завдання кластерного аналізу даних в умовах великої розмірності даних, проведено експерименти, що підтвердили працездатність розробленого математичного забезпечення.

Висновки. Розроблений метод та його програмна реалізація можуть бути рекомендовані для використання практиці у завданнях аналізу даних різної природи та розмірності.

КЛЮЧОВІ СЛОВА: кластер-аналіз, нейро-нечітка мережа, хеш, нечітке виведення, аналіз даних.

DOI 10.15588/1607-3274-2022-4-6

ЛІТЕРАТУРА / ЛИТЕРАТУРА

1. Everitt B. Cluster analysis / [B. Everitt, S. Landau, L. Morven et al.]. – Chichester : Wiley, 2011. – 330 p.
2. Data Clustering : Algorithms and Applications / eds.: C. Aggarwal, C. Reddy, K. Chandan. – New York : Chapman and Hall/CRC, 2016. – 652 p.
3. Huang Z. Extensions to the k-means algorithm for clustering large data sets with categorical values / Z. Huang // Data Mining and Knowledge Discovery. – 1998. – Vol. 2, Issue 3. – P. 283–304. DOI:10.1023/A:1009769707641.S2CID 11323096.
4. Ng R. Efficient and effective clustering method for spatial data mining / R. Ng, J. Han //20th International Conference on Very Large Data Bases (VLDB'94), September 12–15, 1994, Santiago, Chile : proceedings. –Burlington : Morgan Kaufmann,1994. – P. 144–155.
5. Bailey K. D. Typologies and Taxonomies: An Introduction to Classification Techniques / K. D. Bailey. – London: Sage Publications, 1994. – 96 p.
6. Gordon A. D. Classification / A. D. Gordon. – Boca Raton : Chapman & Hall/CRC, 1999. – 256 p.
7. Romesburg C. H. Cluster Analysis for Researchers / C. H. Romesburg. – Belmont : Lifetime Learning Publications, 1984. – 334 p.
8. Aldenderfer M. S. Cluster Analysis / M. S Aldenderfer, R. K. Blashfield. – London : Sage Publications, 1984. – 88 p.
9. Meilă M. Comparing Clusterings by the Variation of Information / M. Meilă // Lecture Notes in Computer Science. – 2003. – Vol. 2777. – P. 173–187. DOI:10.1007/978-3-540-45167-9_14.
10. Hierarchical Clustering Based on Mutual Information / [A. Kraskov, H. Stögbauer, R. G. Andrzejak, P. Grassberger, Peter]: [Electronic resource]. – Access mode: https://arxiv.org/abs/q-bio/0311039.
11. Frey B. J. Clustering by Passing Messages Between Data Points / B. J. Frey, D. Dueck // Science. – 2007. –Vol. 315, № 5814. – P. 972–976. DOI: 10.1126/science.1136800.
12. Pfitzner D. Characterization and evaluation of similarity measures for pairs of clusterings / D. Pfitzner, R. Leibbrandt, D. Powers // Knowledge and Information Systems. – 2009. . – Vol. 19, № 3. – P. 361–394. DOI:10.1007/s10115-008-0150-6.
13. Dunn J. Well separated clusters and optimal fuzzy partitions / J. Dunn // Journal of Cybernetics. – 1974. – № 4. – P. 95–104. DOI:10.1080/01969727408546059.
14. Rand W. M. Objective criteria for the evaluation of clustering methods / W. M. Rand // Journal of the American Statistical Association. – 1971. – Vol. 66 (336). – P. 846–850. DOI:10.2307/2284239.
15. Hubert L. Comparing partitions / L. Hubert, P. Arabie // Journal of Classification. 1985. – Vol. 2. – P. 193–218. DOI:10.1007/BF01908075.S2CID189915041
16. .Di Marco A. Clustering and Diversifying Web Search Results with Graph-Based Word Sense Induction / A. Di Marco, R. Navigli // Computational Linguistics. – 2013. – Vol. 39, № 3. – P. 709–754. DOI:10.1162/COLI_a_00148. S2CID 1775181.
17. Arnott R. D. Cluster Analysis and Stock Price Comovement / R. D. Arnott // Financial Analysts Journal. –1980. –Vol. 36, № 6. – P. 56–62. DOI: 10.2469/faj.v36.n6.56. ISSN 0015-198X.
18. Dunn J. C. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters / J. C. Dunn // Journal of Cybernetics. – 1973. – Vol. 3, Issue 3. – P. 32–57. DOI:10.1080/01969727308546046.
19. A Modified Fuzzy C-Means Algorithm for Bias Field Estimation and Segmentation of MRI Data / [M. Ahmed S. Yamany, N. Mohamed, A. Farag, T. Moriarty] // IEEE Transactions on Medical Imaging. – 2002. – Vol. 21, № 3. – P. 193–199. DOI:10.1109/42.996338.
20. Abonyi J. Cluster Analysis for Data Mining and System Identification / J. Abonyi, B. Feil. – Berlin : Birkhäuser Verlag, 2007. – 306 p. DOI: 10.1007/978-3-7643-7988-9
21. Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition / [F. Höppner, F. Klawonn, R. Kruse, T. Runkler]. – Chichester: John Wiley & Sons, 1999. – 304 p.
22. Miyamoto S. Fuzzy Sets in Information and Retrieval and Cluster Analysis / S. Miyamoto. – Dordrecht : Kluwer Academic Publishers, 1990. – 274 p.
23. Banerjee T. Day or Night Activity Recognition From Video Using Fuzzy Clustering Techniques / T. Banerjee // IEEE Transactions on Fuzzy Systems. – 2014. – Vol. 22, Issue 3. – P. 483–493. DOI:10.1109/TFUZZ.2013.2260756.
24. Advances in Fuzzy Clustering and its Applications / eds.: J. Valente de Oliveira, W. Pedrycz. – Chichester : John Wiley & Sons, 2007. – 454 p.
25. Bezdek J. C. Pattern Recognition with Fuzzy Objective Function Algorithms / J. C. Bezdek. – New York : Plenum Press, 1981. – 272 p. DOI: 10.1007/978-1-4757-0450-1
26. Dumitrescu D. Fuzzy Sets and Applications to Clustering and Training / D. Dumitrescu, B. Lazzerini, L. C. Jain. – Boca Raton: CRC Press, 2000. – 664 p.
27. Finding Hierarchies of Subspace Clusters / [E. Achtert, C. Böhm, H.-P. Kriegel et al.] // Lecture Notes in Computer Science. – 2006. – Vol. 4213. – P. 446–453. DOI:10.1007/11871637_42. ISBN 978-3-540-45374-1.
28. Detection and Visualization of Subspace Cluster Hierarchies / [E. Achtert, C. Böhm, H. P. et al.] // Lecture Notes in Computer Science. –2007. – Vol. 4443. – P. 152–163. DOI:10.1007/978-3-540-71703-4_15.
29. Johnson S. C. Hierarchical clustering schemes / S. C. Johnson // Psychometrika. – 1967. – Vol. 32, № 3. – P. 241–254. DOI:10.1007/BF02289588
30. A Survey on Locality Sensitive Hashing Algorithms and their Applications [Electronic resource] / [O. Jafari, P. Maurya, P. Nagarkar et al.]. – Access mode: https://arxiv.org/pdf/2102.08942
31. Buhler J. Efficient large-scale sequence comparison by locality-sensitive hashing / J. Buhler // Bioinformatics. – 2001. – Vol. 17, № 5. – P. 419–428.
32. Zhao K. Locality Preserving Hashing / K. Zhao, H. Lu, J. Mei // Twenty-Eighth AAAI Conference on Artificial Intelligence, 27–31 July 2014, Québec : proceedings. – Palo Alto: AAAI Press, 2014. –P. 2874–2880.
33. Tsai Y.-H. Locality preserving hashing / Y.-H. Tsai, M.-H. Yang // 2014 IEEE International Conference on Image Processing (ICIP), Paris, 27–30 of October 2014: proceedings. – Los Alamitos: IEEE, 2014. – P. 2988–2992. DOI: 10.1109/ICIP.2014.7025604.
34. Feature Hashing for Large Scale Multitask Learning / [K. Weinberger, A. Dasgupta, J. Langford et al.] // 26th Annual International Conference on Machine Learning (ICML '09) Montreal, June 2009 : proceedings. – New York : ACM, 2009. – P. 1113–1120. DOI: 10.1145/1553374.1553516
35. Wolfson H. J. Geometric Hashing: An Overview / H. J. Wolfson, I. Rigoutsos // IEEE Computational Science and Engineering. – 1997. – Vol. 4, No 4. – P. 10–21.

DOI 10.15588/1607-3274-2022-4-6

36. Fast supervised discrete hashing / [J. Gui, T. Liu, Z. Sunet et al.] // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2017. – Vol. 40, No 2. – P. 490–496. DOI: 10.1109/TPAMI.2017.2678475
37. Aluç, G. Building self-clustering RDF databases using Tunable-LSH / G. Aluç, M. Özsu, K. Daudjee // The VLDB Journal. – 2018. – Vol. 28, № 2. – P. 173–195. DOI:10.1007/s00778-018-0530-9
38. Pauleve L. Locality sensitive hashing: A comparison of hash function types and querying mechanisms / L. Pauleve, H. Jegou, L. Amsaleg // Pattern Recognition Letters. – 2010. – Vol. 31, № 11. – P. 1348–1358. DOI:10.1016/j.patrec.2010.04.004.
39. Andoni A. Near-Optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions / A. Andoni, P. Indyk // Communications of the ACM. – 2008. – Vol. 51, № 1. – P. 117–122. DOI:10.1145/1327452.1327494.
40. Salakhutdinov R. Semantic hashing / R. Salakhutdinov, G. Hinton // International Journal of Approximate Reasoning. – 2008. – Vol. 50, № 7. – P. 969–978. DO:10.1016/j.ijar.2008.11.006.
41. Fast similarity sketching / [S. Dahlgaard, M. Knudsen, M. Thorup] // 2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS), 15–17 October 2017, Berkeley. – Los Alamitos : IEEE, 2017. – 663–671. DOI: 10.1109/FOCS.2017.67
42. Chin A. Locality-preserving hash functions for general purpose parallel computation / A. Chin // Algorithmica. – 1994. – Vol. 12, Issue 2–3. – P. 170–181. DOI:10.1007/BF01185209. S2CID 18108051.
43. 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. – С. 120–128. DOI: 10.15588/1607-3274-2014-1-17
44. Субботин С. А. Синтез нейро-нечетких диагностических моделей с хэширующим преобразованием в последовательном и параллельном режимах / С. А. Субботин, А. Ю. Благодарев, Е. А. Гофман // Radio Electronics, Computer Science, Control. – 2017. – № 1. – С. 56–65. DOI: 10.15588/1607-3274-2017-1-7
45. Субботин С.А. Хэширование на основе полярных координат для сокращения размерности данных / С.А.Субботин // Radio Electronics, Computer Science, Control. – 2020. – № 4. – P. 118–128. DOI: 10.15588/1607-3274-2020-4-12
46. Субботин С. А. Анализ преобразований для проецирования данных на обобщённую ось в задачах распознавания образов / С. А. Субботин, А. А. Олейник // Штучний інтелект. – 2010. – № 1. – С. 114–121.
47. Субботин С. А. Конструируемые признаки для автоматической классификации распределенных во времени стационарных сигналов / С. А. Субботин // Radio Electronics, Computer Science, Control. – 2012. – № 1. – С. 96–103. DOI: 10.15588/1607-3274-2012-1-19
48. Субботин С. А. Комплексное сокращение размерности данных для построения диагностических и распознающих моделей по прецедентам / С. А. Субботин // Radio Electronics, Computer Science, Control. – 2016. – № 4. – С. 70–76. DOI: 10.15588/1607-3274-2016-4-9
49. Субботин С.А. Оценка информативности и отбор экземпляров на основе хэширования / С.А. Субботин / Radio Electronics, Computer Science, Control. – 2020. – № 3. – С. 129–137. DOI: 10.15588/1607-3274-2020-3-12
50. Oliinyk A. The system of criteria for feature informativeness estimation in pattern recognition / [A. Oliinyk, S. Subbotin, V. Lovkin et al.] // Radio Electronics, Computer Science, Control. – 2017. – № 4. – С. 85–96. DOI: 10.15588/1607-3274-2017-4-10
51. Subbotin S. The instance and feature selection for neural network based diagnosis of chronic obstructive bronchitis / S. Subbotin // Applications of Computational Intelligence in Biomedical Technology / eds.: R. Bris, J. Majernik, K. Pancerz, E. Zaitseva. – Cham : Springer, 2016: – P. 215–228. – (Studies in Computational Intelligence, Vol. 606).
52. 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.
53. Subbotin S. The Fully-Defined Neuro-Fuzzy Model Synthesis / S. Subbotin, A. Oliinyk // Data Stream Mining & Processing (DSMP): 2016 IEEE First International Conference, Lviv, 23–27 August 20016 : proceedings. – Lviv : NU “Lvivska Politeckhnika”, 2016. – P. 9–14.
54. Oliinyk A. O. Using parallel random search to train fuzzy neural networks / A. O. Oliinyk, S. Yu. Skrupsky, S. A. Subbotin // Automatic Control and Computer Sciences. – 2014. – Vol. 48. – №. 6. – P. 313–323.
55. Subbotin S. The method of a structural-parametric synthesis of neuro-fuzzy diagnostic model based on the hybrid stochastic search / S. Subbotin // The experience of designing and application of CAD systems in microelectronics : XI International conference CADSM-2011, Polyana–Svalyava, 23–25 February 2011 : proceedings. – Lviv : NU “Lviv Polytechnic”, 2011. – P. 248 – 249.
56. Субботин С. А. Построение полностью определенных нейро-нечетких сетей с регулярным разбиением пространства признаков на основе выборок большого объема / С. А. Субботин // Radio Electronics, Computer Science, Control. – 2016. – № 3. – С. 47–53. DOI: 10.15588/1607-3274-2016-3-6
57. Halgamuge S. K. A trainable transparent universal approximator for defuzzification in Mamdani-type neuro-fuzzy controllers / S. K. Halgamuge // IEEE Transactions on Fuzzy Systems. – Vol. 6, № 2. – P. 304–314. DOI: 10.1109/91.669031.

DOI 10.15588/1607-3274-2022-4-6