UDC 004.93'12
DOI: 10.15587/2706-5448.2020.220428

Melnyk R.,
Tushnytskyy R.,
Kvit R.,
Salo T.

PRELIMINARY DATA CLASSIFICATION BY MULTILEVEL SEGMENTATION OF HISTOGRAMS FOR CLUSTERING OF HYPERCUBES

The object of research is an algorithm for the classification of large data based on the hierarchical clustering algorithm. The nonlinear complexity of the clustering algorithm does not allow for data samples of 5–10 thousand and above. To classify data, it is necessary to pre-group them. Therefore, the subject of research is the data segmentation algorithm based on piecewise linear approximation.

In the course of the study, let's use hierarchical clustering algorithms, the method of piecewise linear approximation of the cumulative histogram, calculated by a special procedure, and the procedure for searching for segmentation thresholds.

The computational complexity of the classical hierarchical algorithm reaches the value of $O(N^3)$, and certain steps to limit the search can achieve the value of $O(N^2)$, which is confirmed by experiments to study the dependence of the hierarchical tree on the initial sample. An approximate approach to key clustering with partitioning of a set of basic keys is implemented. To reduce further the complexity of the hierarchical clustering algorithm, a decomposition approach based on splitting the initial sample of large data into a number of subsets is proposed. In this article to use the hierarchical clustering algorithm for big data classification the preliminary decomposition method is proposed. It is based on multilevel segmentation of cumulative or ordinary histograms obtained for every feature coordinate characterizing object of data. Thresholds of multilevel segmentation are obtained by piecewise linear approximation of histogram functions. Build hypercubes of data are being accepted as objects for three stages clustering algorithm.

Powerful tool for data classification is obtained, the use of which allows carrying out many experiments with data of various types to identify patterns among the data features. Its application is intended for the processing of patient data, molecular structures, economic problems for making optimal treatment decisions, diagnostics and modeling. Thanks to this approach, data classification can be performed online to obtain the results of direct analysis when data is received, for example, from spacecraft.

Keywords: *cumulative histogram, multilevel segmentation, piecewise linear approximation, hierarchical clustering, decomposition of data space.*

Received date: 10.08.2020
Accepted date: 10.08.2020
Published date: 31.12.2020

1. Introduction

Cluster analysis methods are widely used for decomposition, research, indexing and search, image recognition. In particular, work [1–3] contains a classification of clustering methods and a method for forming the contours of the selected clusters. Works [4–6] devoted to the clustering of graphs of models, which reflect parts of images. In papers [7–9] recent trends in practical algorithms for balanced graph partitioning point to applications and discuss future research directions are surveyed. The hierarchical clustering algorithm in [10] has one stage of the folding procedure and proposes a new criterion for combining to reduce computational costs from $O(N^3)$ complexity to $O(N^2)$.

For large amounts of data with many characteristics, this complexity is also an obstacle to grouping. Therefore, the work proposes several approaches to reduce complexity based on the decomposition of the cluster search space. The first attempts of the authors were demonstrated in works [11–13]. In the study [14] an empirical analysis involving a multi-variable quantitative analysis was used to examine the factors that influence the performance of the innovation cluster.

Presented investigations are actual because nowadays there are a big amount of patients with symptoms of coronavirus whose tests need to be processed as well as characteristics of the decease to find patterns, reasons and medicaments. In practice a number of patients is millions and clustering of hypercubes is a good instrument for these purposes.

The object of research is an algorithm for the classification of large data based on the hierarchical clustering algorithm. The classification algorithm having two stages: preliminary segmentation of the object space to form their groups with similar characteristics and clustering of hypercubes having common coordinates and without them. *The aim of research* is to reduce radically a complexity of the clustering algorithm which are planned to be applied for big data.

ISSN 2664-9969

2. Methods of research

Clustering is one of the instruments to analyze data to know its structure and perform their classification. To build clusters of objects, an algorithm for grouping them by similar characteristics must be realized. An illustration of image pixel clusters having the same colors is shown in Fig. 1.

a *b*

Fig. 1. Examples of clustered images: *a* – for 5 clusters; *b* – for 9 clusters

One of wide spread methods is the agglomerative hierarchical clustering algorithm. It includes the following steps:

– S0. For all elements of the input set $x_i, x_j \in X$.

– S1. Searching of leaf pairs by the similarity function:

$$\forall (x_i, x_j) \quad \text{calculate} \quad F(x_i, x_j). \tag{1}$$

– S2. Comparison and selection of pairs with the smallest distance value:

$$F^*(x_i, x_j) = \min F(x_i, x_j), \quad i, j \in I. \tag{2}$$

For merging the clusters x_i, x_j in a new cluster x_n.

– S3. Recalculate the cluster's centers x_i^0 $(i = 1,...,k)$.

– S4. Remove clusters x_i, x_j from the list of candidates.

– S5. End of the procedure (for all $x_i, x_j \in X$).

In this algorithm for the measure of approximation between two clusters is used the distance between their centers of gravity or their centers of coordinates:

$$F(A,B) = d(i_c, j_c), \tag{3}$$

where a hypothetical object i_c is a centroid of the cluster A, and j_c is a centroid of the cluster B.

The centroid has as many features as every object in the input set or a cluster. So, the function F is being accepted as a weighted sum of modules of differences between the characteristics of centroids:

$$F_{ij} = w_1 |a_i - a_j| + w_2 |b_i - b_j| + w_3 |c_i - c_j| + ..., \tag{4}$$

or weighted sum of squares of the differences between the characteristics of centroids:

$$F_{ij} = w_1 [a_i - a_j]^2 + w_2 [b_i - b_j]^2 + w_3 [c_i - c_j]^2 + ... \tag{5}$$

When two clusters are being merged coordinated of new centroid are recalculated by the following:

$$C_k = C_k \left(\frac{k^* a_i + r^* a_j}{k+r}, \frac{k^* b_i + r^* b_j}{k+r}, ... \right). \tag{6}$$

The algorithm builds a binary hierarchical convolution tree (dendrogram) of the object clusters by the proximity function. An example of the tree for a clustering process as a dendrogram is shown in Fig. 2.

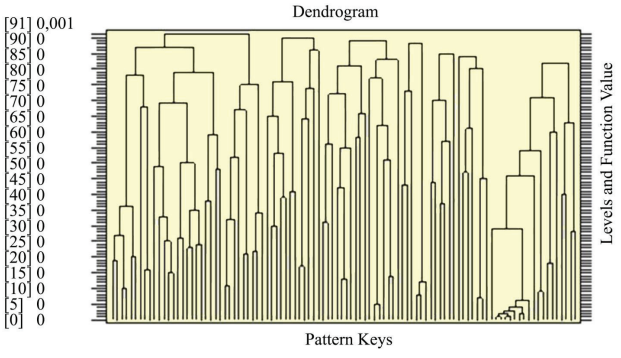


Fig. 2. Dendrogram of clustering process

In this dendogram the vertical axis shows the value between the nodes merged together to create a new node. Each node in the graph besides leaves marks a cluster with a corresponding number of elementary objects.

Algorithmic complexity $O(N^2)$ of the classic hierarchical algorithm does not allow to classify a large data sample, for example, images of practical sizes. For these purposes some approaches based on decomposition technique could be considered.

3. Research results and discussion

3.1. Space decomposition to hypercubes. To reduce the algorithmic complexity the initial set of objects $H(Q_1, Q_2, Q_3, ..., Q_N)$ is being divided into p subsets $H_1(Q_1, Q_2, Q_3, ..., Q_z)$, $H_2(Q_{z+1}, Q_{z+2}, Q_{z+3}, ..., Q_t)$, ..., $H(Q_{t+1}, Q_{t+2}, Q_{t+3}, ..., Q_N)$.

In common, the main input data hypercube must be divided into a number of smaller hypercubes. For the 3-dimension case in Fig. 3 $m \times k \times n$ hypercubes are shown, obtained by dividing corresponding coordinate intervals into m, k, n parts.

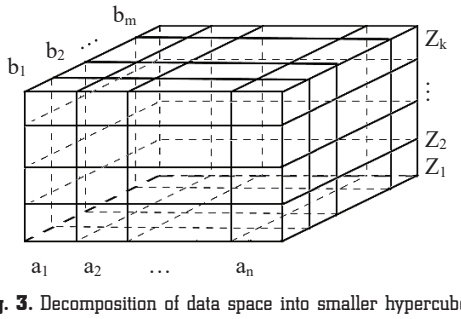


Fig. 3. Decomposition of data space into smaller hypercubes

As examples, consider two possible types of decomposition: hypercubes of different sizes, but with the same numbers of data objects and small hypercubes having the same size and different numbers of objects. These cases are illustrated by division of coordinate intervals for 2-dimension space in Fig. 4.

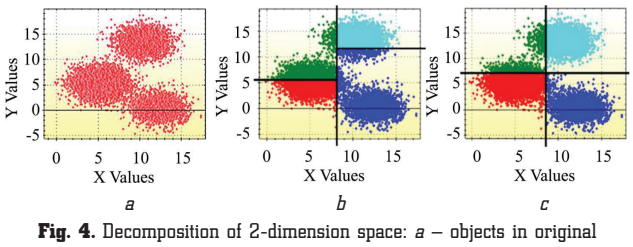


Fig. 4. Decomposition of 2-dimension space: *a* – objects in original 2-cube; *b* – equal numbers of objects; *c* – equal sizes of 2-cubes sides

In common the complexity of the algorithm will be reduced to the complexity of a small hypercube clustering for example $O(p \times n_i^2)$, where p is a number of hypercubes having n_i objects. The price of this complexity reduction is possible losses in resulting variants of data structure including the best ones. Decomposition is very simple for realization. But the first approach needs to apply the sorting procedure for every object feature which does the complexity as $O(N^2)$ and for big data it is unacceptable.

3.2. Multilevel segmentation for decomposition. Having the set of objects $H(Q_1,Q_2,Q_3,...,Q_N)$ to be classified by their features, a histogram is calculated. It contains in its intervals numbers of corresponding feature values:

$$G_i = \sum G_i(j), \quad j = 1,...,n_i, \quad i = 1,...,N. \tag{7}$$

Also the cumulative histogram is calculated:

$$G_{F_i}(s) = \sum G_i(j), \quad j = 1,...,s, \quad i = 1,...,N, \tag{8}$$

where G_i is the overall number of i-th feature values (in absolute or relative values), $G_i(j)$ is number of the object feature values or their frequency in the histogram interval, $G_{F_i}(s)$ is the accumulating frequency (or accumulative number) for the given histogram interval, n_i is the number of histogram and cumulative histogram intervals, s, i are interval numbers.

It is necessary to approximate a histogram function by piecewise-linear function $g(x)$ determined on the interval $a_i \le x_i \le a_{i+1}$ with a set of sections having the end points (α_i, β_i) and $(\alpha_{i+1}, \beta_{i+1})$. The function $g(x)$ minimizes the approximation error ε got by the following formula:

$$\varepsilon = \max_{a_i \le x \le a_{i+1}} |g(x) - f(x)|. \tag{9}$$

The approximation function is presented as:

$$g(x) = \sum_{i=0}^{N-1} g_i(x). \tag{10}$$

For piecewise linear approximation of the histogram function the realized Ramer-Douglas-Peucker (RDP) algorithm [15, 16] was taken as basic.

The algorithm defines maximum distance (tolerance) between the original and the approximating function which must be less than assigned maximum allowable tolerance.

Having no special data to hold experiments, one image is used, which pixel intensity values present values of the object's one feature.

In Fig. 5, a simple image is shown. It contains 900×600 pixels. It has five columns by 180 pixels with colors (imaginary) from one to five. So, there are 540,000 objects and consequently 540,000 values of the first feature for all objects. This feature values are equal to the corresponding pixel intensity values.

In Fig. 6 the cumulative histogram of the «gradient» image is presented. Applying the RDP algorithm to the function $G_{F_1}(s)$ of the cumulative histogram, the threshold values for multilevel segmentation of the function and the image giving this function are obtained. In Fig. 6 the piecewise linear function is marked by red color (above blue color of histogram) and intensity interval borders are marked by green color.

Fig. 5. «Gradient» image for data features

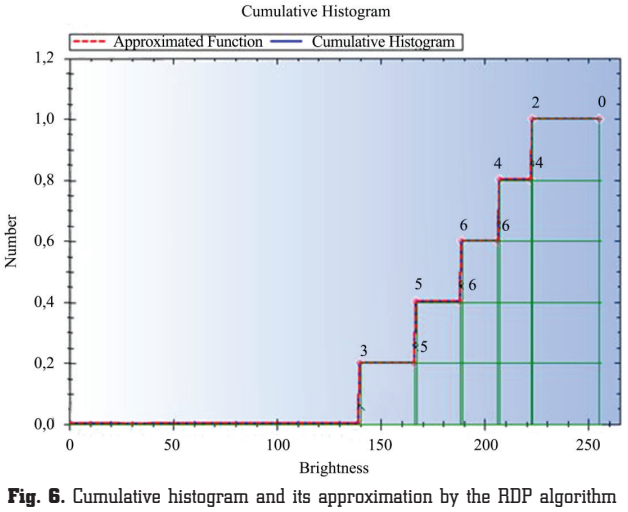


Fig. 6. Cumulative histogram and its approximation by the RDP algorithm

Coordinates of intensity intervals are the following: 140, 167, 189, 207, 222, 255 and corresponding sizes of intervals are as follows: 26, 22, 18, 16, 22, k(23).

Multilevel segmentation gives five intervals of pixel intensity, which correspond to five clusters grouping values of the first feature characterizing objects:

$$\begin{aligned} &F_1(1),...,F_1(600 \times 180) = I_0(1), \\ &F_1(600 \times 180 + 1),...,F_1(600 \times 360) = I_0(2), \\ &\dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \\ &F_1(600 \times 720 + 1),...,F_1(600 \times 900) = I_0(5), \end{aligned} \tag{11}$$

where $F_1(n)$ is the first feature value of the n-th object, $I_0(5)$ is intensity of the 5-th segment in the original image.

The next step of the experiment is inversion of the original image in Fig. 5 to get data of the second feature. The inverted image is given in Fig. 7.

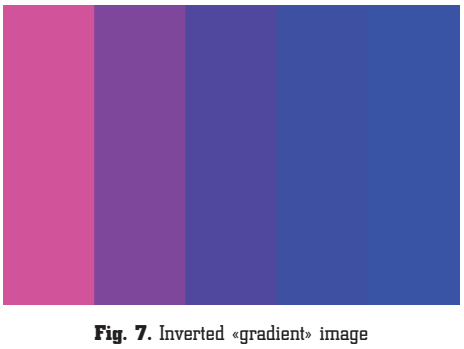


Fig. 7. Inverted «gradient» image

The inverted image contains 540,000 pixels too. It means that 540,000 objects have 540,000 values of the second

ISSN 2664-9969

feature. They are taken as corresponding pixel intensity values of the inverted image ($I_i(1)$–$I_i(5)$).

Applying the RDP algorithm to the function of the cumulative $G_{F_2}(s)$ in Fig. 8, the threshold values for multilevel segmentation of the function and the image giving this function are obtained.

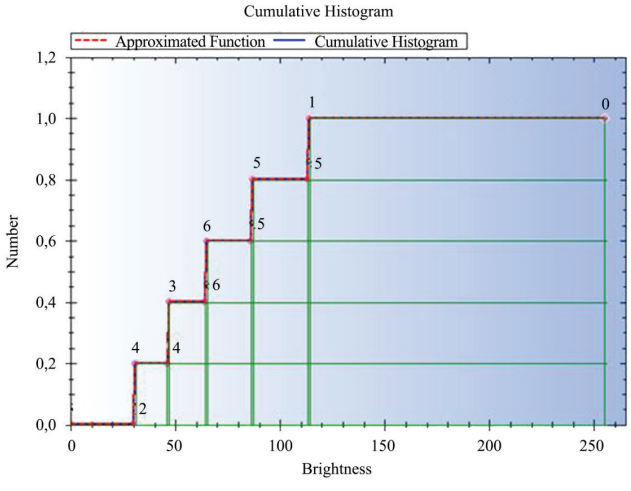


Fig. 8. Cumulative histogram of the inverted image and its approximation by the RDP algorithm

Coordinates of intensity intervals are the following: 30, 47, 65, 87, 114, 255 and corresponding values of intervals are as follows: 16, 18, 22, 26, k(114–255). They are inverted in mirror for the function from Fig. 6.

Multilevel segmentation gives five intervals of pixel intensity which correspond to five clusters grouping values of the second feature characterizing objects:

$$\begin{aligned}
&F_2(1),\ldots,F_2(600\times180)=I_i(1),\\
&F_2(600\times180+1),\ldots,F_2(600\times360)=I_i(2),\\
&\ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\\
&F_2(600\times720+1),\ldots,F_2(600\times900)=I_i(5),
\end{aligned}\qquad(12)$$

where $F_2(n)$ is the second feature value of the n-th object, $I_i(5)$ is intensity of the 5-th segment in the inverted image.

As results of performed operations, data imaginary is classified, marked by different colors of quadrates in Fig. 9. In this rectangle axes have conditional scales but sizes of quadrates are equal between themselves.

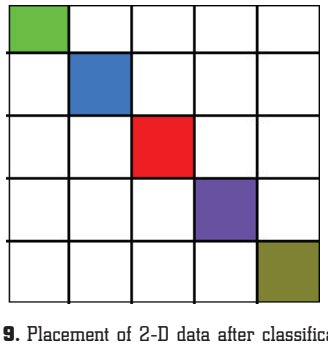


Fig. 9. Placement of 2-D data after classification (original and inverted images)

If in the previous example the second feature of data coincides with the first feature given by Fig. 7, the classification shown in Fig. 10 is obtained.

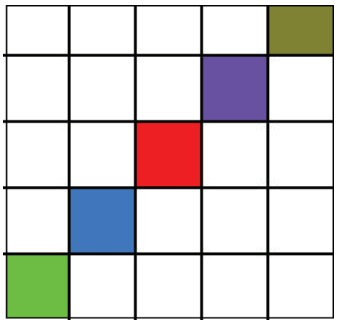


Fig. 10. Placement of 2-D data after classification (features are only from original images)

The next example is for data with first feature from Fig. 7 for original image and the second feature being as pixel intensity of modified original image with three vertical columns in Fig. 11.

Placement of classified data is given in Fig. 12.

Fig. 11. Modified original image

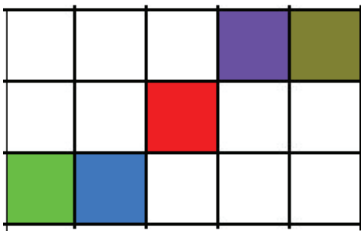


Fig. 12. Placement of 2-D data after classification (original and modified images)

Quadrates representing classified data are clusters having close features (the same within one cluster for these three examples). They correspond to the clusters planned to be obtained by hierarchical algorithms but not obtained for a lack of time. To terminate the clustering process, the hierarchical algorithm is being applied to objects as hypercubes. Resulting trees are shown in Fig. 13.

The algorithmic complexity of the proposed method is $O(N)$. So, all clustering processes for these three big examples were very short comparatively to impossible clustering of all 540,000 objects.

3.3. Overlapping features. To illustrate the approach in 2-D space for more complicated cases, the data as white pixels in two rectangles reviewed. Their sizes are not principle. Rectangles are isolated with crossing coordinates of borders (Fig. 14, *a*) and crossing by their area (Fig. 14, *b*). In this example two coordinates of white pixels are accepted as the object features. Multilevel segmentation by a piecewise linear function gives steps of the grid and corresponding cells.

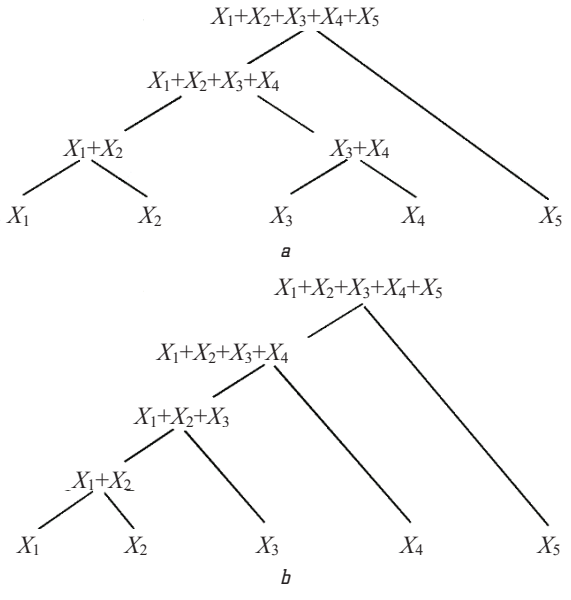


Fig. 13. Examples of hierarchical trees:
a – for hypercubes in Fig. 9 and in Fig. 10; *b* – for hypercubes in Fig. 12

Fig. 14. Images of objects (white pixels): *a* – two rectangles are isolated; *b* – two rectangles are overlapped

This example changes the cumulative histogram (not of the image but of numbers of white pixels in columns and rows) by ordinary histogram (also by numbers) and the last one by a pixel intensity function in the W columns and H rows of the image matrix:

$$\bar{I}(i)=\frac{1}{W}\sum_{j=1}^{W} I(i,j), \quad i=1,2,...,H,$$

$$\bar{I}(j)=\frac{1}{H}\sum_{i=1}^{H} I(i,j), \quad j=1,2,...,W, \qquad (13)$$

where $I(i, j)$ is pixel intensity in i-row and j-column $(1\le i\le H, 1\le j\le W)$.

To do such an exchange it is possible only if objects are as white pixels and background is black. The goal of this exchange is to demonstrate robustness of the algorithm to different models of data presentation: cumulative histograms, histograms and mean intensity functions.

Mean intensity functions for rectangles are calculated. To determine coordinates of vertical and horizontal bands to build the grid, these functions are approximated by piecewise linear funcrtions, which are given in Fig. 15.

So axis *OX* will be divided in five intervals by point of 87, 233, 269, 495. The axis *OY* will be divided also in five intervals by point of 61, 164, 196, 287.

By coordinate values of the function intervals the space for hypercubes (9 cells) is being built in Fig. 16 (in our case the grid with uneven steps). Objects are imaginary placed in corresponding hypercubes (here cells). Fully independent parts of input data are marked with different colors, belonging to the white rectangles and those having common coordinates with other rectangle. Steps of the grid reflect them by axes *OX* and *OY*.

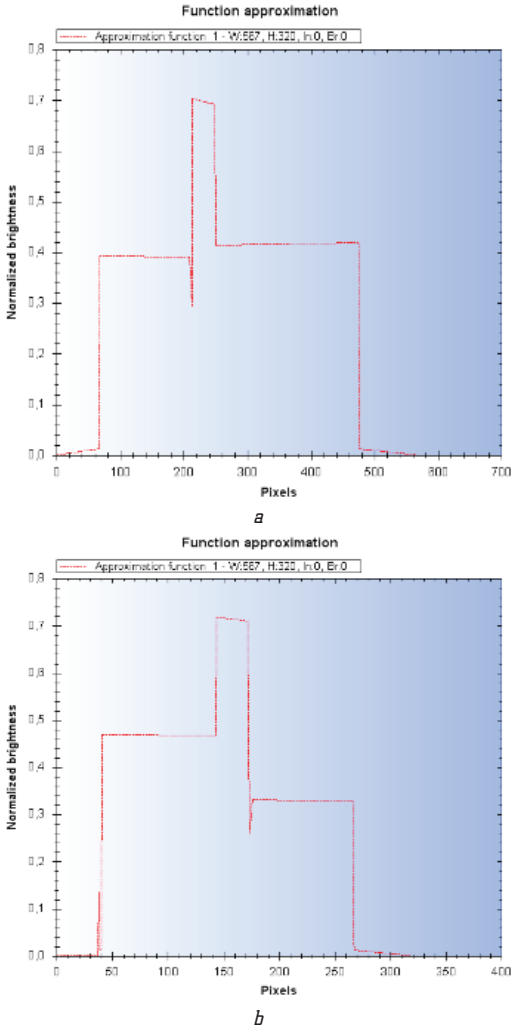


Fig. 15. Mean intensity functions for images of rectangles:
a – in rows; *b* – in columns

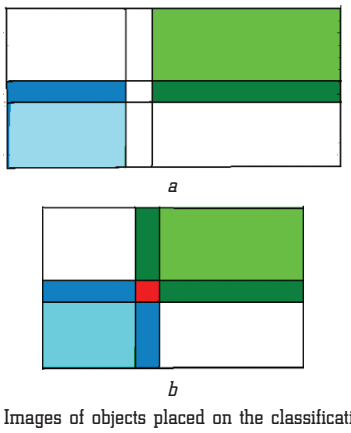


Fig. 16. Images of objects placed on the classification grid:
a – for isolated rectangles; *b* – for overlapped rectangles

Consider one more experiment with data being of more complicated structure (Fig. 17, *a*). Missing the intermediates calculations given in previous examples, the grid of obtained hypercubes is presented (Fig. 17, *b*). The independent hypercubes are marked by lighter colors and hypercubes having common features are marked by darker colors.

ISSN 2664-9969

a *b*

Fig. 17. Figures of pixels for clustering: *a* – initial image of white pixels; *b* – hypercubes of pixels on the classification grid

From resulting images of experiments it is clear that further clustering algorithm is needed to perform clustering to the root. But now in Fig. 16 the space of the input data is much less than at the beginning. For the case of Fig. 16, *a*, there are 4 objects, for the second image in Fig. 16, *b* there are 7 objects. For the grid in Fig. 17, *b* there are 18 objects.

The hierarchical tree is being built for objects representing hypercubes in Fig. 16, *a*, and it is shown in Fig. 18.

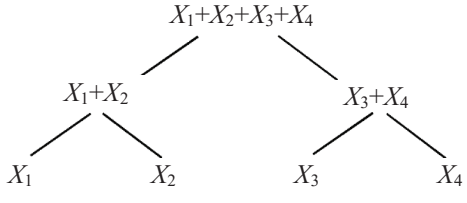


Fig. 18. Hierarchical tree of hypercubes

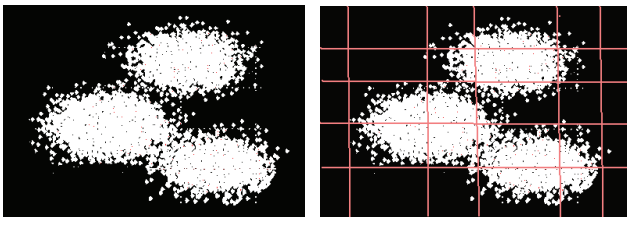
Two examples of 3-D objects and distribution them in hypercubes are illustrated by simple images. For rectangles in Fig. 19, *a* having 5 colors and 5 intervals in every coordinate there are 5×5×5=125 hypercubes and for rectangles in Fig. 19, *b* having 10 colors and 2 intervals in every coordinate there are 10×2×2=40 hypercubes. These hypercubes are taken as leaves for clustering by below considered algorithms.

a *b*

Fig. 19. Pixels of image as 3-D objects: *a* – grouping in 125 hypercubes; *b* – grouping in 40 hypercubes

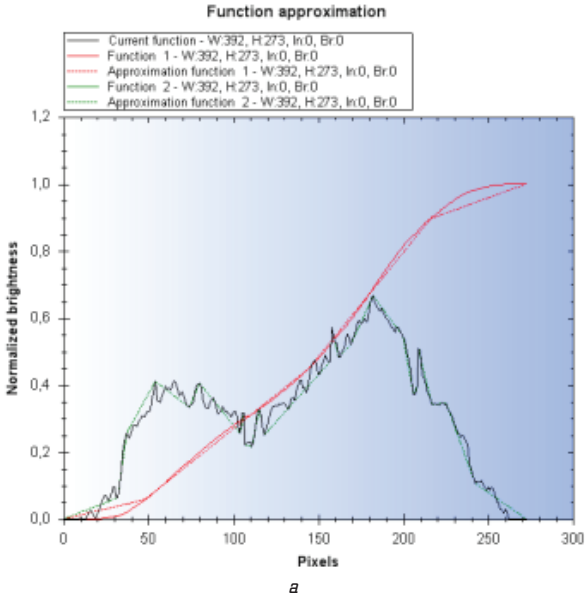
The decomposition process for not discrete example given in Fig. 20 is now illustrated.

For white pixels as objects to be classified, four histograms of white pixels in rows and columns: ordinary and cumulative (Fig. 21) are calculated. It can be seen that ordinary histograms have oscillations connected with distribution of objects in the space. Cumulative histograms have no such peculiarities which influence on the result of approximation.

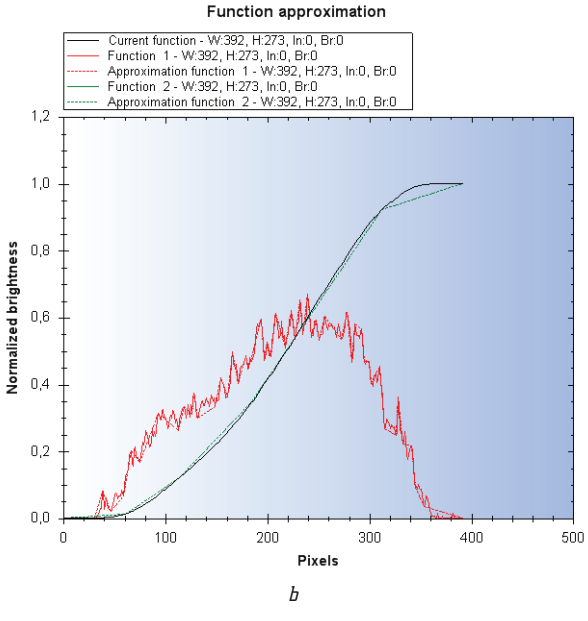


a *b*

Fig. 20. White pixel as objects: *a* – in original image; *b* – divided by multilevel segmentation of histogram



a



b

Fig. 21. Ordinary and cumulative histograms of white pixels: *a* – in rows; *b* – in columns

To get coordinates of multilevel segmentation these functions were approximated (Fig. 22). The difference between two approximation functions can be seen. For ordinary histogram approximation function has 13 intervals the end points of them could be taken as coordinates of hypercube. For cumulative histogram approximation function there are have four intervals to build a hypercube.

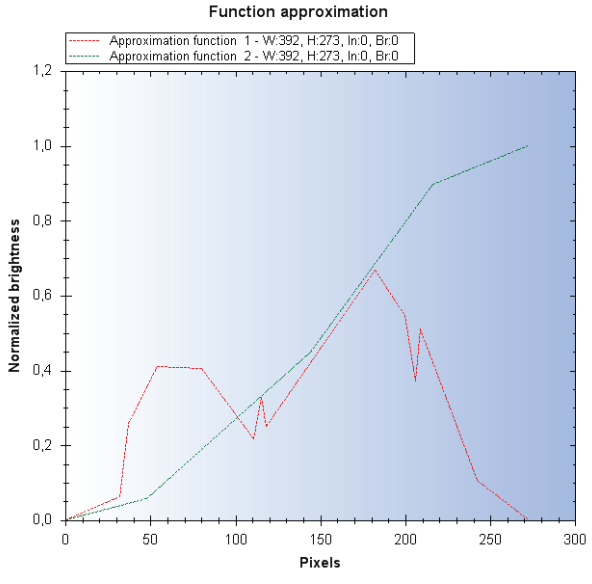


Fig. 22. Approximated by the RDP algorithm ordinary and cumulative histogram functions in rows

According to coordinates of approximated functions a space of the image is being divided into rectangles (for 3-D into cubes and hypercubes). Approximation of cumulative histograms for pixels in row and columns gives 5×6=30 rectangles in Fig. 20, *b* which are being accepted as input data for the next algorithms.

3.4. Three stages clustering algorithm of hypercubes. After decomposition of the data space into a number of hypercubes of objects to data classification continues by a consequence of clustering algorithms. They are applied to data objects in hypercubes and to objects obtained from hypercubes by clustering.

3.4.1. Algorithm of clustering in cascades.

– S0. Leaves for clustering are being prepared: for two cascades they are copied from hypercubes, for many cascades they are copied from a list of resulting clusters (S3).

– S1. Objects of every hypercube $H(H_1, H_2, ..., H_p)$ are clustered by hierarchical agglomerative algorithm. Sets of the clusters $K_1(k_1, k_2, k_3, ...)$, $K_2(k_s, k_{s+1}, k_{s+2}, ...)$, ..., $K_p(k_r, k_{r+1}, k_{r+2}, ...)$ are obtained and $k_1, k_2, ..., k_i$ are clusters of objects from hypercubes $H_1, H_2, ..., H_p$. A number of resulting clusters depends on the coefficient k_r (1, 0.95, ..., 0.01) indicating a reduction ratio of the object's number in a hypercube. So, a resulting number can be from one to full number of objects in the hypercube. The reduction coefficient control time consumption and accuracy of the algorithm.

– S2. Resulting clusters are being accepted as input objects for a new set of leaves:

$$K = K_1(k_1,k_2,k_3,...) \cup \\ \cup K_2(k_s,k_{s+1},k_{s+2},...) \cup ... \cup K_p(k_r,k_{r+1},k_{r+2},...). \quad (14)$$

– S3. Control for next steps is transferred to steps S0 and S1 to continue merging and to build the binary hierarchical tree.

The procedure of so-called multilevel cascading clustering is illustrated by Fig. 23.

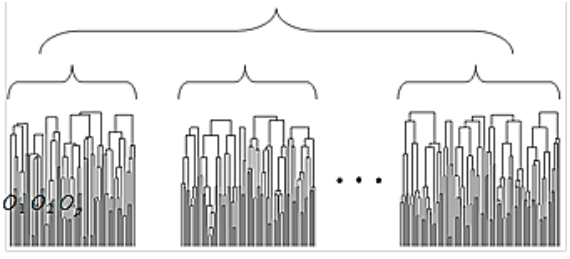


Fig. 23. Decomposition of leaves and clustering trees

Decomposition of basic leaves (data objects) could be arbitrary (random) or controlled by some rules.

Hypercubes obtained from segmentation technique are between themselves in the following relations: some of them are neighbors and have common borders as lines, planes etc., other hypercubes are touching between themselves by one point, and the last group of hypercubes has no relations, i. e. they are isolated by empty space. In such consequence works the proposed clustering algorithm.

The hierarchical clustering algorithm (1)–(6) is applied to hypercubes obtained by decomposition technique. Hypercubes are not trivial objects with a set of features. That is why a clustering procedure is being modified. Every hypercube is characterized by a set of features some of which are also sets: N_i is a number of objects; $x_i(x_{i_1}, x_{i_2}, x_{i_3}, x_{i_4})$ are coordinates of a center; $b_i(b_{i_1}, b_{i_2}, b_{i_3}, b_{i_4})$ are borders of a hypercube; s_i is a square. Hypercubes have two types of borders: internal ones connecting the current hypercube with a neighbor and external ones being borders to free space (Fig. 24).

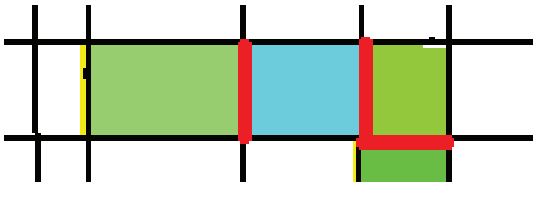


Fig. 24. Two types of hypercube borders

3.4.2. Algorithm for clustering of hypercubes. The algorithm for clustering of hypercubes consists of a consequence of main steps:

– S0. Formation of a control list P of pointers to adjoining hypercubes.

– S1. Selection the two best candidates for merging. There are some strategies of clustering in this step:

a) to select minimal distance between centroids of hypercubes:

$$F^* = \min(F_{kj}), \quad k, j \in P, \quad (15)$$

where F presents distance between centroids:

$$F = d_{ij}(w_1 |a_i - a_j| + w_2 |b_i - b_j| + w_3 |c_i - c_j| + ...); \quad (16)$$

b) to select maximal number of objects in the resulting cluster:

$$F^* = \max(F_{kj}), \quad k, j \in P, \quad (17)$$

where

$$F_{kj} = n_k + n_j; \quad (18)$$

ISSN 2664-9969

c) to minimize displacements of a center of gravity of a new hypercube comparatively to every participant of merging:

$$F^* = \min(F_{ij}), \quad i, j \in P, \tag{19}$$

where

$$F_{ij} = \frac{\min\{|C_i - C_n|, |C_j - C_n|\}}{d_{ij}}, \tag{20}$$

where C_i, C_j are centers of gravity of candidates to be merged; C_n is a center of the new cluster:

$$C_n = C_n\left(\frac{k^*a_i + r^*a_j}{k+r}, \frac{k^*b_i + r^*b_j}{k+r}, \ldots\right). \tag{21}$$

– S2. To build a new hypercubes from both selected, deleting of data connected with merged hypercubes and determination of features for new hypercube.

Three criteria for searching of candidates are used to control a type of the built clusters. First criterion is traditional to consider only distances and minimize it. Second criterion is planned to form maximal clusters on low level of the tree. Third criterion is planned to join small parts to greater ones at the beginning of clustering process.

Very important characteristic of the algorithm is a so called control panel of the convolution process. The panel contains pointers connected with internal borders of hypercubes. So, they point on hypercubes which are candidates to be merged (Fig. 25). Merging process is being continued until the list of the pointers will become empty.

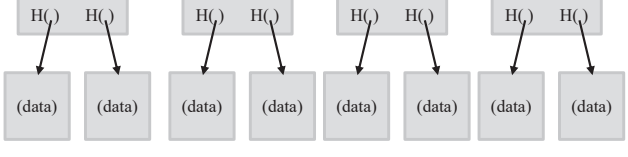


Fig. 25. Pointers on hypercubes

At this point the first stage of the clustering algorithm is being terminated. The second stage begins to work with a control list filled by new pointers. They correspond to the angle points common to two hypercubes (Fig. 26, *a*). The second stage could be excluded if isolated hypercubes in result are preferable.

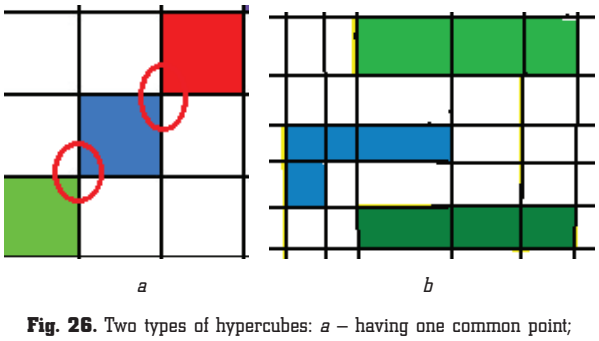


Fig. 26. Two types of hypercubes: *a* – having one common point; *b* – without common elements

And the last stage of the algorithm works then two control panels will be empty and objects for clustering are isolated that is all obtained clusters will be as closed areas (Fig. 26, *b*).

Fig. 27 shows the clustering tree for hypercubes in Fig. 28. It gives distinct two clusters shown by the dendogram. Gray lines correspond to the first stage of the algorithm, blue lines correspond to the hypercubes having one common point and red lines correspond to the third stage of the algorithm. Ordinary algorithm does not allow to get such a result because distances within the same cluster could be greater than distances between different clusters.

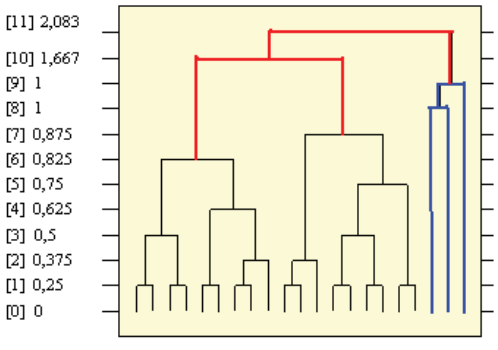


Fig. 27. Hierarchical trees of hypercubes

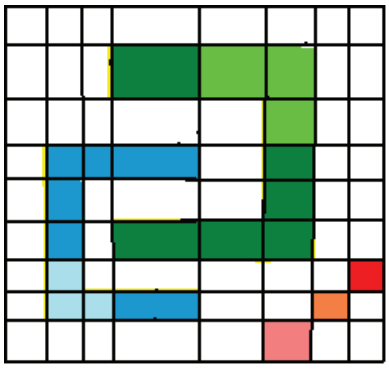


Fig. 28. Example of hypercubes on the classification grid

Above described experiments show that reducing the input data for the clustering algorithm (a number of hypercubes are radically smaller than a number of input objects) a complexity of the classification algorithm is being reduced drastically. The proposed type of segmentation of data space by every coordinate allows to classify objects placed in spaces of complicated architecture.

4. Conclusions

The algorithm of data preliminary classification by decomposition of its space into hypercubes is developed. Coordinates of hypercubes are obtained when the ordinary and cumulative histogram functions are approximated by piecewise linear functions. Then hypercubes are clustered by three stages clustering algorithm: hypercubes having common borders, having common point and isolated areas. Algorithm allows to classify data having complex structure and large size.

A novelty of the proposed classification algorithm lies in a type of data segmentation based on linear piecewise approximation of the cumulative histogram. The cumulative histogram is calculated for objects as white pixels

for every coordinate. Data hypercubes are considered of two types: adjacent and independent. This proposed and realized novelty reduces the algorithm complexity allowing it application to very big data.

References

1. Yip, A. M., Ding, C., Chan, T. F. (2006). Dynamic cluster formation using level set methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 28 (6),* 877–889. doi: http://doi.org/10.1109/tpami.2006.117
2. Viattchenin, D. (2009). D*evelopments in fuzzy clustering. The collection of papers.* Minsk: Vever, 216.
3. Sandeep, V. (2010). *Effective level sets and shape detection: an application to natural images.* Gulbarga: Electronics and Communications Engineering, 132.
4. Grady, L., Schwartz, E. L. (2006). Isoperimetric graph partitioning for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 28 (3),* 469–475. doi: http://doi.org/10.1109/tpami.2006.57
5. Pavan, M., Pelillo, M. (2007). Dominant Sets and Pairwise Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 29 (1),* 167–172. doi: http://doi.org/10.1109/tpami.2007.250608
6. Foggia, P., Percannella, G., Vento, M. (2014). Graph matching and learning in pattern recognition in the last 10 years. *International Journal of Pattern Recognition and Artificial Intelligence, 28 (1),* 1450001. doi: http://doi.org/10.1142/s0218001414500013
7. Dong, X., Shen, J., Shao, L., Van Gool, L. (2016). Sub-Markov Random Walk for Image Segmentation. *IEEE Transactions on Image Processing, 25 (2),* 516–527. doi: http://doi.org/10.1109/tip.2015.2505184
8. Buluç, A., Meyerhenke, H., Safro, I., Sanders, P., Schulz, C. (2016). Recent Advances in Graph Partitioning. *Lecture Notes in Computer Science,* 117–158. doi: http://doi.org/10.1007/978-3-319-49487-6_4
9. Wang, N., Gao, X., Tao, D., Yang, H., Li, X. (2018). Facial feature point detection: A comprehensive survey. *Neurocomputing, 275 (31),* 50–65. doi: http://doi.org/10.1016/j.neucom.2017.05.013
10. Ding, C., He, X. (2005). Cluster aggregate inequality and multilevel hierarchical clustering. *Proc. 9th European Conf. Principles of Data Mining and Knowledge Discovery*, 71–83. doi: http://doi.org/10.1007/11564126_12
11. Melnyk, R., Tushnytskyy, R. (2009). Algorithm accuracy and complexity optimization by inequality merging for data clustering. *Proc. of the 10-th Intern. Conf. The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM),* 453–455.
12. Melnyk, R., Aleekseev, O. (2006). Clustering of pattern keys base on decomposition of their set. *Editing and processing of information, 24 (100),* 110–114.
13. Melnyk, R., Tushnytskyy, R. (2008). Pattern keys clustering using large-scale dataset cascading decomposition. *Computer Science and Information Technology, 604,* 249–254.
14. Shenkoya, T., Kim, E. (2019). A case study of the daedeok innopolis innovation cluster and its implications for Nigeria. *World Technopolis Review, 8 (2),* 104–119. doi: http://dx.doi.org/10.7165/wtr19a1218.21
15. Ramer, U. (1972). An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing, 1 (3),* 244–256. doi: http://doi.org/10.1016/s0146-664x(72)80017-0
16. Douglas, D., Peucker, T. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization, 10 (2),* 112–122.

Melnyk Roman, *Doctor of Technical Sciences, Professor, Department of Software, Lviv Polytechnic National University, Lviv, Ukraine, e-mail: ramelnyk@polynet.lviv.ua, ORCID: https://orcid.org/0000-0002-4329-6740*

Tushnytskyy Ruslan, *PhD, Associate Professor, Department of Software, Lviv Polytechnic National University, Lviv, Ukraine, e-mail: ruslan.b.tushnytskyy@lpnu.ua, ORCID: http://orcid.org/0000-0002-8522-0293*

Kvit Roman, *PhD, Associate Professor, Department of Higher Mathematics, Lviv Polytechnic National University, Lviv, Ukraine, e-mail: roman.i.kvit@lpnu.ua, ORCID: http://orcid.org/0000-0002-2232-8678*

Salo Tetyana, *PhD, Associate Professor, Department of Higher Mathematics, Lviv Polytechnic National University, Lviv, Ukraine, e-mail: tetyan.salo@gmail.com, ORCID: http://orcid.org/0000-0001-9469-7459*