Представлена методика аналізу правила зупинки для алгоритму попередньої кластеризації даних, використовуючи зв'язний ациклічний граф. Правило зупинки дозволяє зупинитись на деякому кроці, вважаючи що подальша кластеризація не призведе до знаходження нових кластерів. Аналіз полягав в застосуванні алгоритму попередньої кластеризації та правила зупинки до серій тестових даних із нормальним законом розподілу, які належали до однієї або багатьох груп

Ключові слова: алгоритм попередньої кластеризації, правило зупинки, зв'язний ациклічний граф

Представлена методика анализа правила остановки для алгоритма предварительной кластеризации данных, используя связный ациклический граф. Правило остановки позволяет остановиться на некотором этапе, считая, что дальнейшая кластеризация не приведет к нахождению новых кластеров. Анализ состоял в применении алгоритма предварительной кластеризации и правила остановки к сериям тестовых данных с нормальным законом распределения, принадлежащих к одной или многим группам

Ключевые слова: алгоритм предварительной кластеризации, правило остановки, связный ациклический граф

UDK 004.9
DOI: 10.15587/1729-4061.2015.51090

DEVELOPMENT OF A STOPPING RULE OF CLUSTERING PERFORMANCE BY USING THE CONNECTED ACYCLIC GRAPH

V. Mosorov
Doctor of Technical Science*
E-mail: volodymyr.mosorov@p.lodz.pl
T. Panskyi
Postgraduate student*
E-mail: panskyy@gmail.com
S. Biedron
Postgraduate student*
E-mail: SBiedron@wpia.uni.lodz.pl
*Institute of Applied Computer Science
Lodz University of Technology
Stefanowskiego str., 18/22, Lodz, Poland, 90-924

1. Introduction

Clustering is a process of dividing data objects (or observations) into subsets (groups of objects). Each subset is a cluster when objects in a single subset are alike and differ from objects from other subsets. A cluster is the unity of congenerous similar objects which can be considered as separate independent unit with certain properties [1]. Different methods of clustering can determine the different number of clusters in the same set of input data. The split of the group of objects into clusters is not performed by humans, but with the use of computer-aided clustering algorithms, though most clustering algorithms need some partial human intervention for inputting initial parameters for the efficient and adequate division. Therefore, clustering is a useful and effective process, since it can cause the detection of previously unknown groups of objects in given input data [2].

Preclustering is performed when the user wants to know whether to perform the clustering process in general, whether there is no strict structure in input data and clustering is not necessary. Preclustering algorithm includes decision rule and stopping rule. When data clustering does not lead to the clusters identification the preclustering algorithm indicates about it. However, when the input data contains clusters, the preclustering algorithm is able to calculate the optimum number of it, and stop using stopping rule without making the clustering redundancy.

Nowadays preclustering algorithm is quite relevant and perspective. Classical clustering algorithms perform clustering post factum, i. e. input data is always at least once undergoes the clustering stage. However, is the clustering required when the input data have no structure (noise, random data set)? The first application of preclustering algorithm (in the automatic control field) is the answer to the question whether the input data may include clusters. Another application is the property borrowed and integrated from the classical clustering algorithms, i. e. calculating the optimal number of clusters, in the case when the decision rule showed the possibility of its existence.

In this article a stopping rule for a preclustering algorithm has been developed, which is described in detail below at section 5.

2. Analysis of published data and problem statement

Clustering analysis is widely used in many applications, such as business analytics, pattern recognition, web search, biology and safety. Nowadays, data clustering is developing rapidly which causes the appearance of hundreds of clustering algorithms just published and new ones still appearing. The fields of science and technology prospective for the development and investigating new solutions in clustering analysis are data analysis, statistics, machine learning, spatial database technologies, search engines, marketing, production systems for defect identification, software and hardware systems for detection of failures, faults etc [3].

Cluster analysis is a statistic method which assumes that input data structure is unknown or partially known. The aim of analysis in this case is the detection of some "category" structure, hierarchic inner structure which would refer to observations, so, often the problem is formulated as the task of choosing "natural groups" from the total number of all input objects. But, in spite of this, one of the most important factors of performing cluster analysis is the stopping rule of clusterization. The distinctive feature of clustering is finding a structure in the investigated data, but its disadvantage is the introduction of an additional redundant structure into these data. Clustering allows finding structures even in the data which do not have it a priori (overclustering) [4]. For example, if input data do not follow strict distribution laws, are chaotic, or make up a single general group of all objects, the application of clustering algorithms in these cases would cause the undesirable artificial data splitting. Overclustering leads to the appearance of artifacts, that is, erratic results of cluster finding. The clustering process involves finding a balance between underclustering and overclustering. The stopping rule helps to perform clustering up to the certain step for the optimal determination of the number of clusters and gives the chance to avoid data overclustering. Another advantage of the use of the stopping rule in the preclustering algorithm is the considerable decrease in the algorithm calculation complexity.

Presented problem of setting the optimal, appropriate input parameters of clusters number, that is setting the initial number of clusters K for further clustering is well described in the literature, where the new as well as the classical techniques and methods are explained [5, 6]. However, some of them needs a prior knowledge about the possible number of clusters and most requires statistical sequential substitution the number of clusters for comparison and determining the desired parameters for one or another used technique [7, 8]. In this article presented and used a stopping and the decision rule will automatically find the optimal number of clusters without the previous information about the number of possible clusters.

3. Purpose and objectives of the study

The key purpose of this paper is analysis of a stopping rule for the data preclustering algorithm

In accordance with the set goal the following research objectives are identified:

– creating a stopping rule and modifying a preclustering algorithm in its accordance, using the connected acyclic graph for result visualization;

– testing the preclustering algorithm for the selected cases of input data.

4. Review of main methods and techniques of clusterization

Today, in literature sources many different data clustering algorithms are described. It caused the difficulty of the precise classification and categorization of clustering methods, because some of them can belong to different categories at the same time. The intersection of the method categories is the reason of the fact that the method possesses the features of different categories and it cannot be referred to certain category. Despite this fact, the general view of the arranged representation of the classification of clustering methods is quite important. In general, the main basic clustering methods can be classified by such categories [9–11]:

1. Partitioning methods. Taking into account the set of n objects, partitioning methods split data into k groups, where each group is a cluster and k<n. Each group after its splitting should contain at least one object. Each object belongs to only one group. This requirement can become weaker, for example, in fuzzy methods of partition. Most partitioning methods are based on the distance between objects. These clustering methods work well when seeking for spherical clusters in small or medium databases.

2. Hierarchical methods. The algorithm creates the hierarchical decomposition of data objects. Hierarchical methods can be classified either as agglomerative or divisive according to the way of forming the hierarchical decomposition. Agglomerative, or so-called bottom-up approach begins its activity from assigning every object to a separate group, sequentially merging objects or groups of objects which are similar to each other into bigger clusters, until all objects belong to one general group, or at the condition of stopping the clustering process. Divisive or so-called top-down approach begins with merging all objects into one general cluster and then sequentially splits the group of the objects into smaller ones, until every object belongs to separate cluster. The hierarchical methods are based on the distance between objects, but the parameters of density of object aggregations can also be used here. The hierarchical methods act by sequential split or merge of objects at each iteration of the algorithm. These methods are those of direct action and their drawback is the impossibility of returning to the result of previous iteration (for example, for correcting erratic decisions).

3. Density-based methods. Such methods can detect only spherical clusters, and problems may arise while detecting arbitrary form clusters. Other clustering methods were developed on the basis of the notion of density. The cluster is growing till density (a number of objects) in some area of investigation exceeds some threshold value. This group of methods (based on the density of objects) precisely detects clusters with dense object aggregation, but these methods are not effective at the analysis of fuzzy groups of objects (for example, at the uniform distribution of objects).

4. Grid-based clustering methods. Grid methods are based on the quantization of space of the objects into the given number of sells which make up grid structure. The peculiarity of the algorithms belonging to this group is the transfer from the analysis of separate sample objects to the analysis of the objects of the grid structure. Being calculated easily, they offer a possibility of detecting the clusters of a complex shape. One of the drawbacks of grid algorithms is the strong dependence of the quality of detected clusters on cell dimensions.

5. *Model-based methods.* This type of algorithms use certain models for clusters and tries to optimize the fit between the data and mathematical models. In model-based clustering, it is assumed that the data are generated by a mixture of probability distributions in which each component represents a different cluster. Main drawbacks are finding the initial distribution parameters, and setting the appropriate

model which is user dependence. Another disadvantage is slow processing time on large data sets.

Review of clustering methods and their brief characteristics is presented at the Table 1.

Table 1

Review of clustering methods and their brief characteristics

Method	General characteristics
Partitioning	– detect spherical clusters – are based on the distance between objects – can use mean value or medoid for presenting a cluster center – are effective for small or medium data sets
Hierarchical	– are based on data decomposition – cannot correct erratic merges or splits of objects
Density-based	– detect arbitrary form clusters – cluster are groups of objects with high density separated by areas with low density – can filter outliers
Grid-based	– use the grid resolution of data structure – are independent of a number of objects, but depend on grid dimensions
Model-based	– each component is described be the density function – cluster objects match the distribution

The preclustering algorithm [12] as opposed to other existing algorithms does not require that input parameters or threshold values should be set for the correct determination of the number of clusters. Therefore, this preclustering algorithm has been chosen from the totality of the clustering algorithms as priority for the primary analysis of input data being investigated.

5. Description of the investigated algorithm

The published preclustering algorithm and its main part – the decision rule – determines the existence of one or two clusters in the input data array. This algorithm uses several assumptions:

a) input array is partitioned into two clusters K_1 and K_2 accordingly,

b) input array is a single general cluster including all investigated objects.

The preclustering algorithm is based on the Euclidean distance between objects in two-dimensional space. Preclustering is the procedure of checking the possibility of input data clustering. For effective and adequate performance of this algorithm forced c-means clustering of the whole input data array is carried out. Forced clustering allows the split of the input array into two clusters and the comparison of their parameters with the help of the decision rule. After the forced split of the input array into two clusters, the average distances $d_1(K_1)$ and $d_2(K_2)$ of each cluster K_1 and K_2 accordingly are calculated separately, as well as the mean distance of the general cluster $d_{12}(K_1 \cup K_2)$ created by merging two clusters.

Calculated mean distances $d_1(K_1)$ and $d_2(K_2)$ of separate clusters K_1 and K_2 are compared with the mean distance of general cluster $d_{12}(K_1 \cup K_2)$ with the help of the decision rule.

$$\begin{cases} \text{if} \quad d_{12}(K_1 \cup K_2) > d_1(K_1) + d_2(K_2), \\ \text{otherwise.} \end{cases} \tag{1}$$

If the first decision rules inequality is true – analyzed data array consists of two separate clusters, otherwise – analyzed data array makes up one cluster. Despite the advantages of this preclustering algorithm, it has several drawbacks. The problem of this algorithm is hidden in the decision rule. Its performance is based on the assumption that the input data array is either one general cluster or two separate clusters, but the algorithm cannot detect more than two clusters in one data array (for example, three of them). But in practice the number of clusters in one data array can be infinitely large, so the preclustering algorithm for the correct detection of any quantity of clusters should be verified. For algorithm verification a modified decision rule is proposed, as well as its interpretation in the form of connected acyclic graph.

6. Verification of algorithm parameters

The verification of the parameters of preclustering algorithm consists in the change of the decision rule and its testing for different types of input data samples. Due to the impossibility of testing all possible cases some simplifications and generalizations were introduced. Chosen cases are input data with normal distribution law which were aggregated into one or more clusters. For the adequate finding of the number of clusters the decision rule was modified as follows:

Step 1. The application of forced k-means clustering for the initial input data array. Forced clustering splits all input array into two clusters. With the help of the decision rule (1) it is checked if this data array contains one general cluster or two separate clusters. If a general array is a single cluster, the preclustering algorithm stops at this step, but if according to the decision rule two clusters exist, the algorithm continue to split each separate cluster.

Step 2. For each separate cluster the forced split into two clusters is performed with the help of a k-means algorithm and the calculations and comparison of mean distances are carried out using rule (1).

Cluster splitting stops when at a proper step the mean distance of the general cluster (parent) is less than the sum of mean distances of split clusters (sons), i. e.

$$d_1(K_{i,j...n}) + d_2(K_{i,j...n}) > d_{12}(K_{i,j...n-1}).$$

Numbering of clusters can be like $K_{i,j...n}$, where $i,j...n = 0,1$ is cluster indexation according to the splitting step. At one step the cluster can take on either value 0 or 1, because the forced clustering splits a cluster into two separate groups. For example, when a primary cluster K_0 splits, two clusters $K_{0,0}$ and $K_{0,1}$ can be formed.

For the detailed analysis let us consider cluster determining by the example shown in Fig. 1, *a, b*.

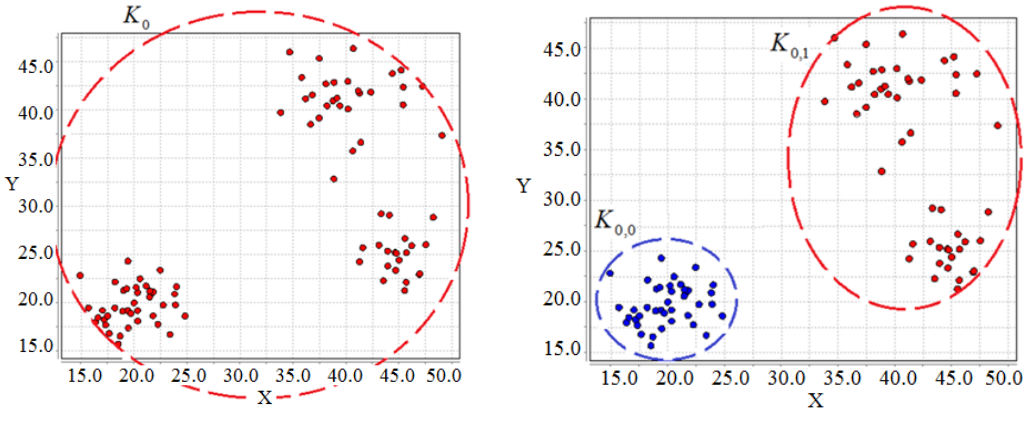
Step 1. Using forced k-mean clustering we split input data array into two clusters having $K = 2$ (the number of clusters) as an input parameter for k-means. Forced clustering splits the whole array into two clusters $K_{0,0}$ and $K_{0,1}$. With the help of the decision rule (1) we check whether one general cluster $(K_0 = K_{0,0} \cup K_{0,1})$ is present in

this data array, or there are two separate clusters $K_{0,0}$ and $K_{0,1}$ there.

Checking split clusters and general joint cluster with the decision rule (1), we can draw a conclusion that more than one cluster exists in this array. For determining the particular number of clusters the force split of separate clusters is continued.

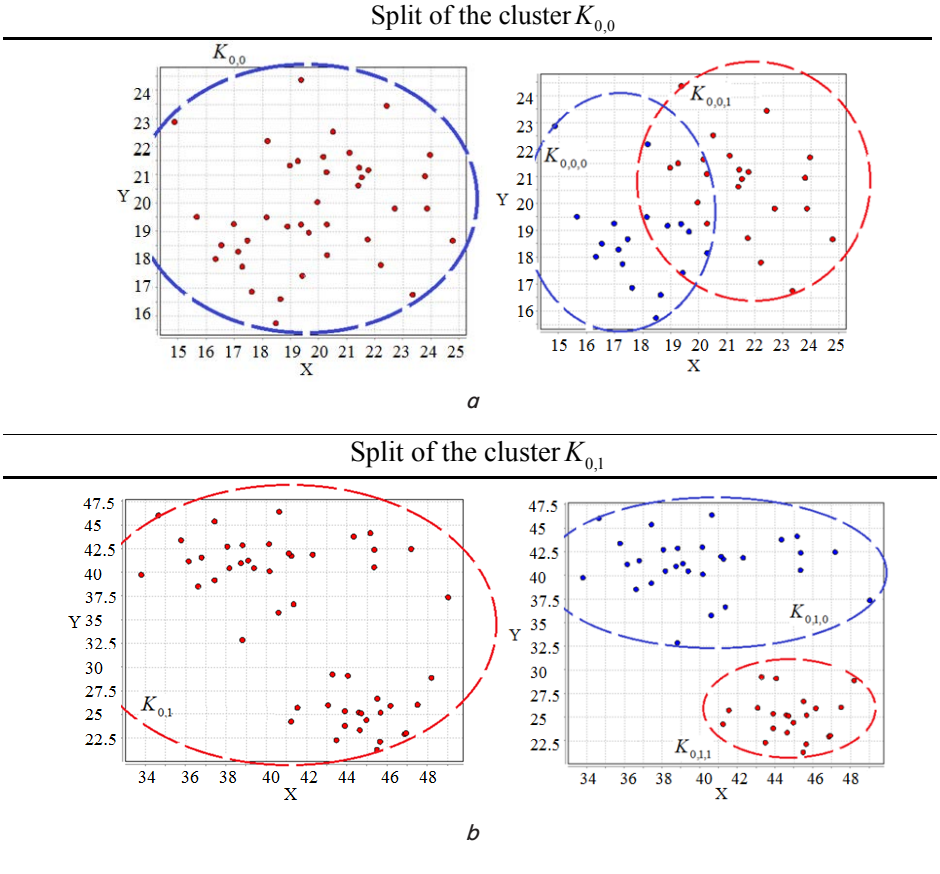
Step 2. For each separate cluster, that is, for $K_{0,0}$ and $K_{0,1}$ the forced split into two clusters is performed with the help of the k-means algorithm and the calculation of their mean distances. At the split of the cluster $K_{0,0}$ into two clusters, the cluster $K_{0,0}$ is considered to be a unified general cluster (a parent), and clusters $K_{0,0,0}$ and $K_{0,0,1}$ are resulting separate clusters (sons). Similarly to this, the split of the cluster $K_{0,1}$ into two separate clusters $K_{0,1,0}$ and $K_{0,1,1}$ is performed. The forced split of the clusters is shown in Fig. 2.

Checking the split of the cluster $K_{0,0}$ with the decision rule (1) we can draw a conclusion that resulting clusters $K_{0,0,0}$ and $K_{0,0,1}$ are not separate, that is, the split of the cluster $K_{0,0}$ will not cause the appearance of new clusters. The stopping rule does not allow splitting clusters $K_{0,0,0}$ and $K_{0,0,1}$, because the condition of existence of two clusters is not satisfied.



a *b*

Fig. 1. Example of splitting the two input data array into two clusters (Step 1):
a — initial data set, *b* — data set is divided into clusters



a

b

Fig. 2. Example of the split of separate clusters $K_{0,0}$ and $K_{0,1}$ (Step 2):
a — split of a cluster $K_{0,0}$ into two clusters, *b* — split of a cluster $K_{0,1}$ into two clusters

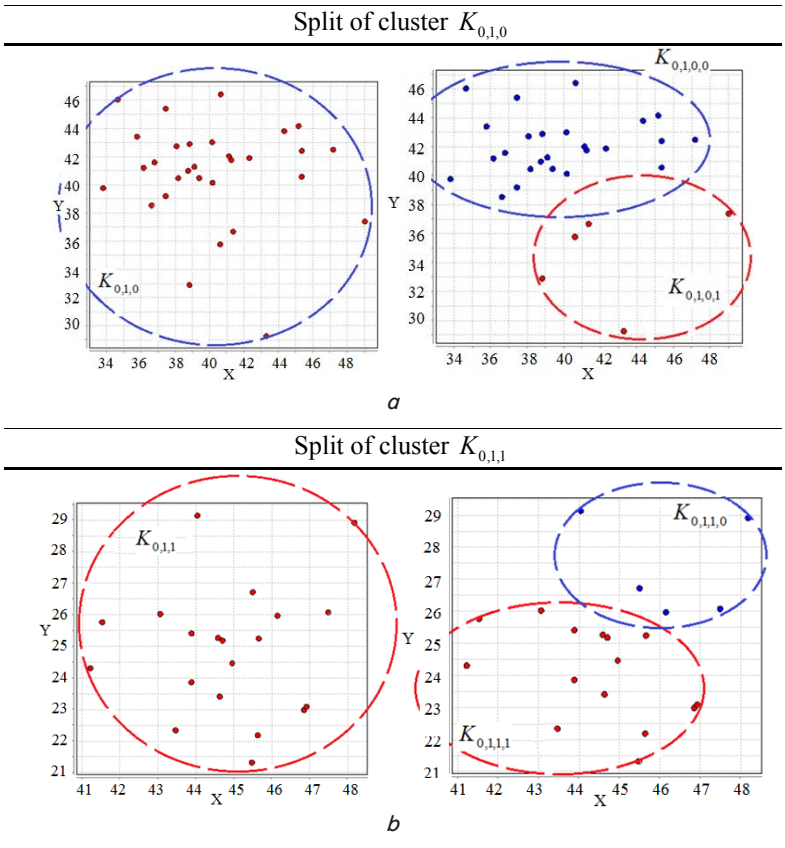


Fig. 3. Example of the split of separate clusters $K_{0,1,0}$ and $K_{0,1,1}$ (Step 3): *a* — split of a cluster $K_{0,1,0}$ into two clusters, *b* — split of a cluster $K_{0,1,1}$ into two clusters

Splitting the cluster $K_{0,1}$ and checking it with the decision rule (1) we can draw the conclusion that two separate clusters $K_{0,1,0}$ and $K_{0,1,1}$ exist. For checking the possibility of the existence of smaller clusters each separate cluster $K_{0,1,0}$ and $K_{0,1,1}$ is forced to split into two clusters.

Step 3. Each separate cluster $K_{0,1,0}$ and $K_{0,1,1}$ is forced to split into two clusters. The k-mean split of each cluster is shown in Fig. 3.

Checking split separate clusters with the decision rule (1) we can draw the conclusion that clusters $K_{0,1,0}$ and $K_{0,1,1}$ do not contain smaller clusters, that is, the split of these clusters will not cause the appearance of new clusters. This fact stops the preclustering algorithm, since the whole input data array has been checked.

To show how the decision rule and the stopping rule work the connected acyclic graph is used which can be shown as in Fig. 4.

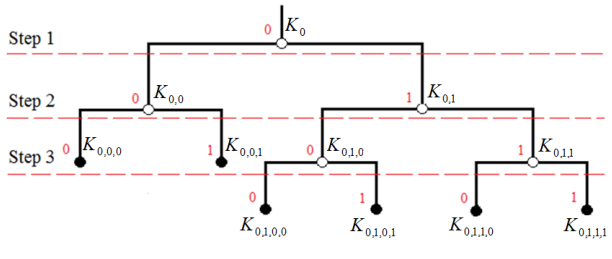


Fig. 4. Decision and stopping tree of the preclustering algorithm

The division of the cluster begins from the main general cluster sequentially splitting every next cluster into two clusters and checking clusters with the decision rule (1) at every step of the division. As it can be seen in Fig. 4, every cluster (a parent) can be split into only two parts, that is, it can have only two sons, and their indexation will be 0 or 1. The indexation begins from general cluster and grows sequentially (top-down) to the desired particular cluster. On these conditions the index of a previous cluster shows which cluster have been split (which cluster is a parent). The results of performing the decision rule (1) are marked as follows:

"○" – This group of objects is a separate cluster. Forced division is performed for checking the possibility of the existence of smaller clusters in given separate cluster.

"●" –Given group of objects is not a separate cluster; it is only a part of a bigger cluster. At this step the forced division stops.

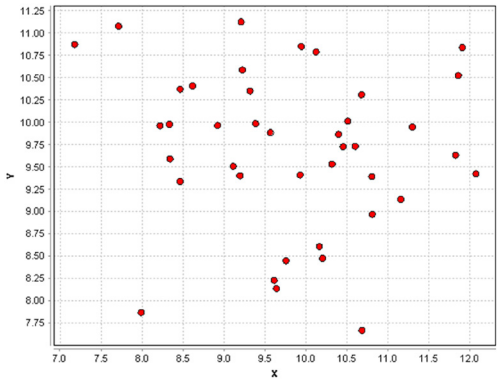
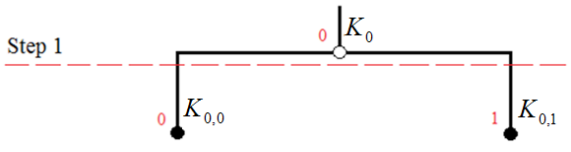
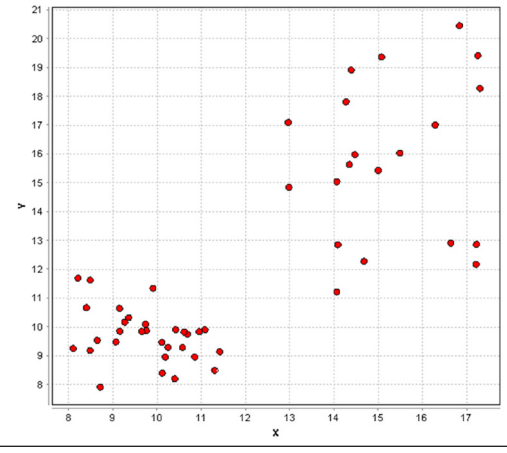
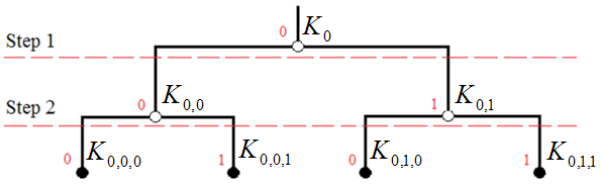
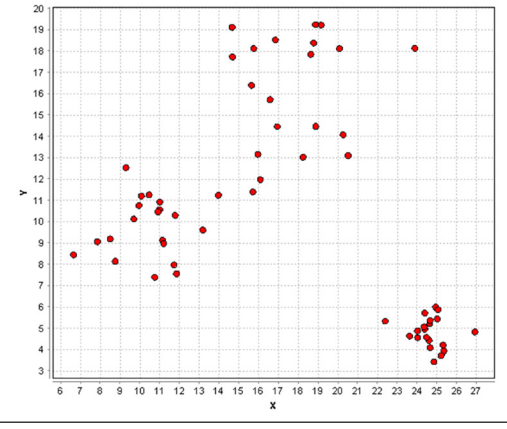
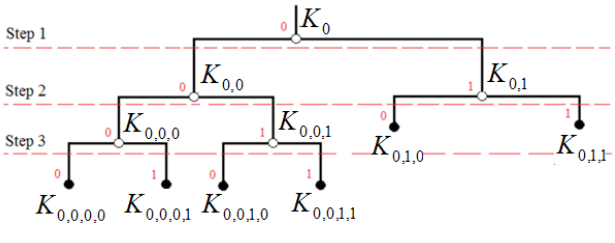
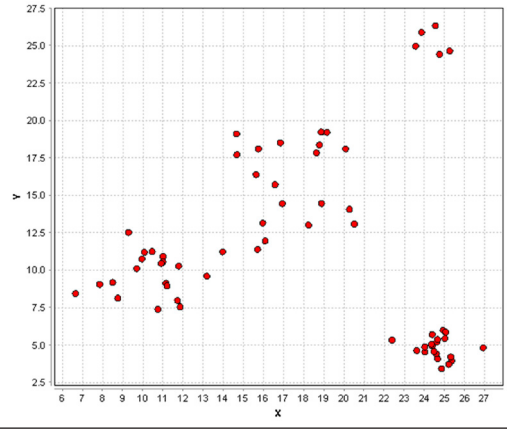
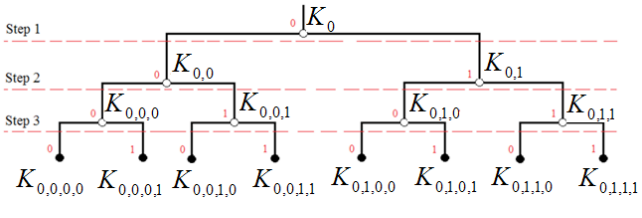
At Fig. 4 it can be seen that at the second step of splitting two clusters $K_{0,0}$ and $K_{0,1}$ appeared, but at the third step of splitting clusters $K_{0,1,0}$ and $K_{0,1,1}$ descended from the cluster $K_{0,1}$. The preclustering algorithm together with the stopping rule detected three clusters in the given input array. The calculation of the number of clusters is performed at the last step under the condition that the given group of objects is a cluster.

7. The verification results of the preclustering algorithm

Chosen samples of input data were analyzed and tested by the preclustering algorithm together with the stopping rule and then represented in the form of trees in Table 2.

Table 2

Samples of input data analyzed and tested by preclustering algorithm and stopping rule

Nr..	Input array of two-dimentional data	Representing the algorithm performance in the form of a tree
1		In this array there is one cluster K_0 
2		In this data array there are two clusters $K_{0,0}$ and $K_{0,1}$ 
3		In this data array there are three clusters $K_{0,0,0}$, $K_{0,0,1}$, $K_{0,1}$ 
4		In this data array there are four clusters $K_{0,0,0}$, $K_{0,0,1}$, $K_{0,1,0}$, $K_{0,1,1}$ 

Forced k-means clustering can be replaced by some other clustering algorithm based on the distance between objects (c-means, x-means, k-medoid).

8. Conclusions

In this article the analysis of the applying the stopping rule in the preclustering algorithm with the usage of connected acyclic graph for results visualization has been performed. Testing the performance of the stopping rule on different samples of input data confirms the expediency of its usage in the preclustering algorithm. The modified decision rule and stopping rule allow us to find any number of clusters in the input data, and at the same time the number of clusters that had been found is the optimal number and does not require checking using different validity measures. Their advantage is also the considerable simplification of algorithm calculations. In spite of the advantages of the preclustering algorithm applied together with the stopping rule, there are drawbacks [13] which impose some limitations of the correct determination of the number of clusters. One of the drawbacks is the dependence of the results on calculated mean distances, that is, on the results of k-means clustering. If clusters are located close to one another and contain isolated objects (single objects located far from other objects of a cluster), the stopping rule can work incorrectly (for example, k-means algorithm will continue splitting the data array when it really should be stopped).

Since in practice clusters could be of arbitrary size and shape, next step will be the development of the preclustering algorithm and its further modification for detecting clusters based on the objects density.

References

1. Bailey, K. Numerical Taxonomy and Cluster Analysis [Text] / K. Bailey // Typologies and Taxonomies, 1994. – 34 p. doi: 10.4135/9781412986397.n3
2. Jain, A. K. Data Clustering: A Review [Text] / A. K. Jain, M. N. Murthy, P. J. Flynn. – ACM Computing Reviews, 1999. – 69 p.
3. Aggarwal, C. C. Data Clustering: Algorithms and Applications. 1st Edition [Text] / C. C. Aggarwal. – Chapman & Hall, 2013. – 652 p.
4. Illumina. Diagnosing and Preventing Flow Cell Overclustering on the MiSeq® System [Text] / Illumina. – 2015. – 10 p.
5. Hofmann, M. RapidMiner: Data Mining Use Cases and Business Analytics Applications [Text] / M. Hofman, R. Klinkenberg. – Chapman & Hall/CRC, 2013. – 431 p.
6. Kovács, F. Cluster Validity Measurement Techniques [Text] / F. Kovács, C. Legány, A. Babos // Proceeding AIKED'06 Proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases, 2006
7. Rendón, E. Internal versus External cluster validation indexes [Text] / E. Rendón, I. Abundez, A. Arizmendi, E. M. Quiroz // International journal of computers and communications. – 2011. – Vol. 5, Issue 1. – P. 25–34.
8. Liu, Y. Understanding of Internal Clustering Validation Measures [Text] / Y. Liu, Z. Li, H. Xiong, X. Gao, J. Wu // IEEE International Conference on Data Mining, 2010. – P. 911–916. doi: 10.1109/icdm.2010.35
9. Rokach, L. Clustering Methods [Text] / L. Rokach, L. O. Maimon // Data Mining and Knowledge Discovery Handbook, 2005. – P. 321–352. doi: 10.1007/0-387-25465-x_15
10. Jain, A. K. Algorithms for clustering data [Text] / A. K. Jain, R.C. Dubes. – Prentice Hall, 1988. – 320 p.
11. Gan, G. Data Clustering: Theory, Algorithms and Applications [Text] G. Gan, C. Ma, J. Wu. – ASA-SIAM Series on Statistics and Applied Probability, 2007. – 466 p.
12. Mosorov, V. Image Texture Defect Detection Method Using Fuzzy C-Means Clustering for Visual Inspection Systems [Text] / V. Mosorov, L. Tomczak // Arabian Journal for Science and Engineering. – 2014. – Vol. 39, Issue 4. – P. 3013–3022. doi: 10.1007/s13369-013-0920-7
13. Qian, W. Analyzing popular clustering algorithms from different viewpoints [Text] / W. Qian, A. Zhou // Journal of Software. – 2002. – Vol. 13, Issue 18. – P. 1383–1394.