O. M. Tkachenko, Cand. Sc. (Eng.), Assist. Prof.; O. F. Grijo Tukalo; O. V. Dzys; S. M. Lakhovets

METHOD OF CLUSTERIZATION, BASED ON SERIAL LAUNCHING OF K-MEANS WITH IMPROVED SELECTION OF THE CANDIDATE FOR NEW POSITION OF INSERTION

The paper suggests improved method of K-means clusterization, which, unlike the conventional method, allows to obtain the solution, close to global minimum of distortion by means of serial launching of k-means for $1, 2, \ldots, k$ *.centroides. Decrease of distortion is achieved at the expense of improvement of the procedure of vectors-candidates definition on the selection of insertion position of new centroide without considerable slowing down of operation time. .*

Key words: *code books, clusterization, k-means, centroides, kd-trees.*

Introduction

Clusterization it is the division of certain set of objects into nonintersecting subsets (clusters) so that, each cluster contains similar objects, and objects of various clusters differ from each other. Clusterization is often used for statistical analysis of data [1], vector quantization, pattern recognition [3], etc. In field of speech compression clusterization algorithms are used for creation of coding books – special tables, containing most representatives data sets. The task of data clusterization is an important element of data processing problem, for its solution there exists many approaches and algorithms: from intuite and heuristic to strictly mathematic. The task of clusterization can be formulated in the following way: the set of n vectors each of which has dimensionality d, must be divided into subjects in accordance with the preset optimization criterion. As a rule, minimization of distortion is this criterion. There exists various ways of distortion evaluation, but in greater part of applied realization the sum of root-mean-square Euclidian distances between the vector and the centre of the clusters (centroide), to which it belongs [2, 4].

Method of k-means clusterization is widely spread and well studied as compared with other clusterization methods. It minimizes the above-mentioned distortion, distributing the data between regions, which are not intersected and are identified by their centers. Wide application of k-means method is due to its main advantages: simplicity, flexibility, rapid convergence. However, practical application of the method is limited by its drawbacks: the results of clusterization, applying the method of k-means greatly depend on the selection of initial configuration of centroides (initialization), algorithm operation is considerably slowed down while clusterization of large volumes of data; algorithm can be converged to local minimum of efficiency function.

To avoid these drawbacks some modifications of k-means method were suggested. The improved initialization procedure is introduced in k-means++ method, that enables to improve the results of clusterization at the expense of special choice of initial configuration of centroides [5]. To accelerate the process of distances computation from points to centroides in [6] it is suggested to exclude from consideration static centroides, i.e., centroides, remained at their positions during current iteration. In [7], in order to reduce the computational complexity of k-means method, special data structure - kd-trees, is used, that enabled to reduce considerably computational complexity of the method. In order to prevent local convergence in [8] iterative algorithm is suggested, allowing to approach to global optimum by means of step-wise serial launching of k-means.

Problem set-up

The given paper combines the advantages of the considered approaches in order to improve the method of k-means clusterization, in which the solution, close to global minimum, is obtained by

means of serial launching of k-means for 1, 2, ..., k centroides and decrease of distortion is achieved at the expense of improved procedure of vectors –candidates determination on selection of insert position of new centroide.

Classic algorithm of k-means

Clusterization by k-means method distributes the input sets of vectors by k clusters $\mathbf{S}_i (i = 1, 2, ..., k)$, centroide $\mathbf{c}_i$ is connected with each of them. Let us denote the set of input vectors $\mathbf{S} = \{\mathbf{x}\}, |\mathbf{S}| = n$ Let D(x,c) be the distance between vector $\mathbf{x}$ and centroide $\mathbf{c}$ .In the given paper non-weighted Euclidian distance between vector $\mathbf{x} = (x_1, x_2, ..., x_d)$ and centroide $\mathbf{c} = (c_1, c_2, ..., c_d)$:

$$D^2(\mathbf{x}, \mathbf{c}) = \sum_{i=1}^{d} (x_i - c_i)^2.$$

We will denote the set of centroides, obtained on iteration t, $\mathbf{SC}_t = \{\mathbf{c}_i\}$. Clusterization algorithm of k-means in its conventional variant is described in the following way:

1. We set $t = 0$ and set initial location of centroides $\mathbf{SC}_0$.

2. For the given set of centroides $\mathbf{SC}_t$, we perform operations, described in paragraphs 2.1 and 2.2 and obtain improved set of centroides $\mathbf{SC}_{t+1}$

2.1 We find such division of $\mathbf{S}$, that distributes $\mathbf{S}$ by k clusters $\mathbf{S}_i (i = 1, 2, ..., k)$ and satisfy the condition

$$\mathbf{S}_i = \{\mathbf{x} \mid D(\mathbf{x}, \mathbf{c}_i) \le D(\mathbf{x}, \mathbf{c}_j) \ \forall j \ne i\}.$$

2.2 We calculate centroide $\mathbf{c}_i$ for each cluster $\mathbf{S}_i (i = 1, 2, ..., k)$, to obtain new set of centroides $\mathbf{SC}_{t+1}$:

$$c_{ij} = \frac{1}{m_i} \cdot \left(\sum_{l=1}^{m_i} x_{lj}\right), j = 1, 2, ..., d, \qquad (1)$$

where m_i is the number of vectors belonging to cluster $\mathbf{S}_i$.

3. We calculate total distortion $E^2 = \sum_{\mathbf{x} \in S} D^2(\mathbf{x}, \mathbf{c})$ for $\mathbf{SC}_{t+1}$. If it differs from the distortion, obtained at previous iteration by minor value, we stop the process. In other case we assume $t \leftarrow t + 1$ and return to step 2.

Algorithm converges during finite number of iterations. Clusterization error and number of iterations depend initial selection of centroides, that is why, common practice is launching of k-means several times with different initial candidates to centroides [9].

Global algorithm of k-means

At it was mentioned above the algorithm of k-means clusterization has certain drawbacks. In order to overcome these drawbacks the improved variant of k-means clusterization, called by the authors greedy global k-means algorithm is suggested in [10]. The assumption, that global optimum can be reached by means of k-means launching, when $(k-1)$ centroide is in optimal positions, obtained for the solution of clusterization problem for $(k-1)$ centroide, and k-th centroide can be placed in corresponding position, which must be determined is the base of this variant. Optimal clusterization for $k = 1$ is easy to obtain, after calculation of the coordinates as arithmetic mean of corresponding coordinates of all vectors of $\mathbf{S}$ set. Thus, for realization of the given approach, aimed at obtained of k centroides, the serial launching of k-means for 1, 2, ..., k centroides is required.

Global algorithm of k-means provides, that the search of corresponding position of i-th centroide, which is unknown, where as the positions of previous $(i-1)$ centroides are known, requires the

launching of k-means for each vector $\mathbf{x}_i \in \mathbf{S}$ from the set of input vectors, that is considerd as the candidate for the position of new centroide insert. Finally the variant, that provides minimal total distortion for all centroides is chosen. It means, that for practical applications, where the quantity of vectors is several tens of hundreds of thousands, and the quantity of clusters is several thousands, time of algorithm operation is too long.

The complexity of the algorithm can be considerably reduced, if launching of k-means algorithm is performed not for each input vector $\mathbf{x}_i \in \mathbf{S}$, but for certain set of vectors $\mathbf{X} = \{\mathbf{x}_0\} \subset \mathbf{S}$, which will be used as the candidates for position of new centroide insert. The choice of $\{\mathbf{x}_0\}$ set can be performed, for instance, according to the scheme, applied in the algorithm k-means++. It is clear, that the power of candidates to centroides set $\{\mathbf{x}_0\}$ will influence total distortion and operation time of the algorithm. Determination of such amount of candidates, that would provide minimal distortion without considerable slowing-down of operation algorithm, will be studied in the given paper.

In our case, the obtaining of k centroides is carried out by means of serial launching of k-means for $1, 2, ..., k$ centroides, that is why, having chosen the candidate for insertion of the next centroide, operation of recalculation of new position of the centroide may be carried out (choice of the position of i^{th} centroide with recalculation of new position of centroides) or may not be carried out (choice of the position of i^{th} centroide without recalculation on centroides position). This is due to the fact, that as a result of recalculation coordinates of centroides change considerably only at the beginning, when the quantity of certain centroides is not great, further, with the increase of the number of determined centroides, redistribution of the points between centroides does not take place, i.e. recalculation is not of great importance the coordinates of centroides do not change greatly.

In [10] it is suggested for the search of suitable position of i-$^{\text{th}}$ centroide to be limited by the choice of the vector, that provides minimal distortion while adding it as initial position of new centroide, instead of launching k-means for each vector. This allows to reach considerable reduction of the time of algorithm operation. The results obtained are close to global optimum.

Total distortion for k centroides is calculated by the formula:

$$E_k^{\,2} = \sum_{i=1}^{k} e_i^2 = \sum_{i=1}^{k}\sum_{j=1}^{N_i} D^2(\mathbf{x}_j, \mathbf{c}_i) = \sum_{i=1}^{k}\sum_{j=1}^{N_i}\sum_{m=1}^{d} (x_{jm} - c_{im})^2 . \qquad (2)$$

Computation of total distortion according to the formula (2) requires the execution of such number of operations: $N_{op} = 2 \cdot d \cdot N_i \cdot k$, where N_i – is the number of points, belonging to centroide $\mathbf{c}_i$.

The expression for the distortion of e_i centroide $\mathbf{c}_i = (c_1, c_2, ..., c_d)$ can be reduced to minimum:

$$\begin{aligned} e_i^{\,2} &= \sum_{j=1}^{N_i}\sum_{m=1}^{d} (x_{jm} - c_{im})^2 = \sum_{j=1}^{N_i}\sum_{m=1}^{d} (x_{jm} - c_{im})^2 = \sum_{j=1}^{N_i}\sum_{m=1}^{d} (x_{jm}^{\,2} - 2 \cdot x_{jm} \cdot c_{im} + c_{im}^{\,2}) = \\ &= \sum_{m=1}^{d} (\sum_{j=1}^{N_i} x_{jm}^{\,2} - 2 \cdot c_{im} \cdot \sum_{j=1}^{N_i} x_{jm} + \sum_{j=1}^{N_i} c_{im}^{\,2}) = \\ &= \sum_{m=1}^{d} (\sum_{j=1}^{N_i} x_{jm}^{\,2} - \frac{2}{N_i} \cdot (\sum_{j=1}^{N_i} x_{jm})^2 + \frac{N_i}{N_i^{\,2}} \cdot (\sum_{j=1}^{N_i} x_{jm})^2) = \sum_{m=1}^{d} (\sum_{j=1}^{N_i} x_{jm}^{\,2} - \frac{1}{N_i} \cdot (\sum_{j=1}^{N_i} x_{jm})^2), \end{aligned} \qquad (3)$$

Instead of storing coordinates of centroide $\mathbf{c}_i$ computation of the distortion according to the formula (3) requires storage of the quantity and the sum of points coordinates, belonging to it by each dimensionality d (i.e. $\sum_{j=1}^{N_i} x_{jm}^{\,2}$). It is clear, that having $\sum_{j=1}^{N_i} x_{jm}^{\,2}$, the coordinates of centroide can always be obtained.

In general case for the selection of the best candidate for $(k+1)$-th centroide. It is necessary to evaluate total distortion $E_{k+1}{}^2$ for each candidate for for centroides and select the variant, providing minimum value. But since $E_k{}^2$ is the same for all candidates, then it is sufficient to evaluate the difference Δ_{k+1}^2, obtained as result of introduction of the given candidate as $(k+1)$-th centroide:

$$\begin{aligned}
&\Delta_{k+1}^2 = E_k^2 - E_{k+1}^2 = \sum_{i=1}^{k}(e_i)^2 - \sum_{i=1}^{k+1}(e_i')^2 = \\
&= \sum_{i=1}^{k}\left[\sum_{m=1}^{d}(\sum_{j=1}^{N_i} x_{jm}{}^2 - \frac{1}{N_i}\cdot(\sum_{j=1}^{N_i} x_{jm})^2) - \sum_{m=1}^{d}(\sum_{j=1}^{N_i-M_i} x_{jm}{}^2 - \frac{1}{N_i - M_i}\cdot(\sum_{j=1}^{N_i} x_{jm})^2)\right] - \\
&- \sum_{m=1}^{d}\left(\sum_{i=1}^{k}(\sum_{j=1}^{M_i} x_{jm}{}^2 - \frac{1}{\sum_{i=1}^{k} M_i}\cdot\sum_{i=1}^{k}\sum_{j=1}^{M_i}(x_{jm})^2\right) = \\
&= \sum_{m=1}^{d}\left[\begin{array}{l}\sum_{i=1}^{k}\left(\sum_{j=1}^{N_i} x_{jm}{}^2 - (\sum_{j=1}^{N_i} x_{jm}{}^2 - \sum_{j=1}^{M_i} x_{jm}{}^2) + \sum_{j=1}^{M_i} x_{jm}{}^2\right) - \\ -\sum_{i=1}^{k}\frac{1}{N_i}\cdot(\sum_{j=1}^{N_i} x_{jm})^2 + \sum_{i=1}^{k}\frac{1}{N_i - M_i}\cdot(\sum_{j=1}^{N_i-M_i} x_{jm})^2 + \frac{1}{\sum_{i=1}^{k} M_i}\cdot\sum_{i=1}^{k}(\sum_{j=1}^{M_i} x_{jm})^2\end{array}\right] = \\
&= \sum_{m=1}^{d}\left\{\sum_{i=1}^{k}\left[\frac{1}{N_i - M_i}\cdot\left(\sum_{j=1}^{N_i} x_{jm} - \sum_{j=1}^{M_i} x_{jm}\right) - \frac{1}{N_i}\cdot(\sum_{j=1}^{M_i} x_{jm})^2\right] + \frac{1}{N_{k+1}}\cdot\sum_{i=1}^{k}(\sum_{j=1}^{M_i} x_{jm})^2\right\},
\end{aligned} \tag{4}$$

where M_i – is the number of points $\mathbf{c}_i$, that will belong to centroide $\mathbf{c}_{k+1}$.

It should be noted, that in many cases (especially with the increase of the amount of determined centroides) the number of points $\mathbf{c}_i$, which will belong to the centroide $\mathbf{c}_{k+1}$ is not great, i.e. $M_i << N_i$.

Thus, since the sum of points coordinates $\sum_{j=1}^{N_i} x_{jm}{}^2$, belonging to centroide $\mathbf{c}_i$, remains and $\sum_{j=1}^{M_i} x_{jm}{}^2$ can be obtained in the process of determination of points belonging to centroide $\mathbf{c}_{k+1}$, the obtained benefit, according to (4) can be calculated during rather short period of time. In this case the expression of operations amount will have the form: $N_{op} = (M_i + 5)\cdot d \cdot k$.

Method of k-means with computation of distances to active centroides

First of all it should be noted, that the most labour-consuming part of computations is to find centroide closest to the given vector, since this requires computations of distances from each vector to each centroide.

After performing of each next iteration t still fewer centroides, change their position (active centroides $\mathbf{SC}_t^{(a)}$), and greater part of them remain at their positions (passive centroides $\mathbf{SC}_t^{(p)}$).
Hence, we store for each point the distances to all centroides on iteration t, it is sufficient to calculate the distance only to active centroides $\mathbf{SC}_t^{(a)}$ on iteration $t+1$. In this case, gain in time will be greater, if the relative share r_t of active centroides among all centroides on iteration t is less:

$$r_t = \frac{\left|\mathbf{SC}_t^{(a)}\right|}{\left|\mathbf{SC}_t\right|}, \tag{5}$$

where $\left|\mathbf{SC}_t^{(a)}\right|$ and $\left|\mathbf{SC}_t\right|$ – are powers of sets $\mathbf{SC}_t^{(a)}$ and $\mathbf{SC}_t$ correspondingly.

Fig 1 shows, how the share of active centroides r changes with the growth of total number of centroides $|\mathbf{SC}|$, obtained as a result of clusterization of 75000 vectors of $d = 5$ dimensionality. The data along both axes are given in logarithmic scale. For smoothing the curve, presented in Fig, the data were averaged while 1000 iteration.

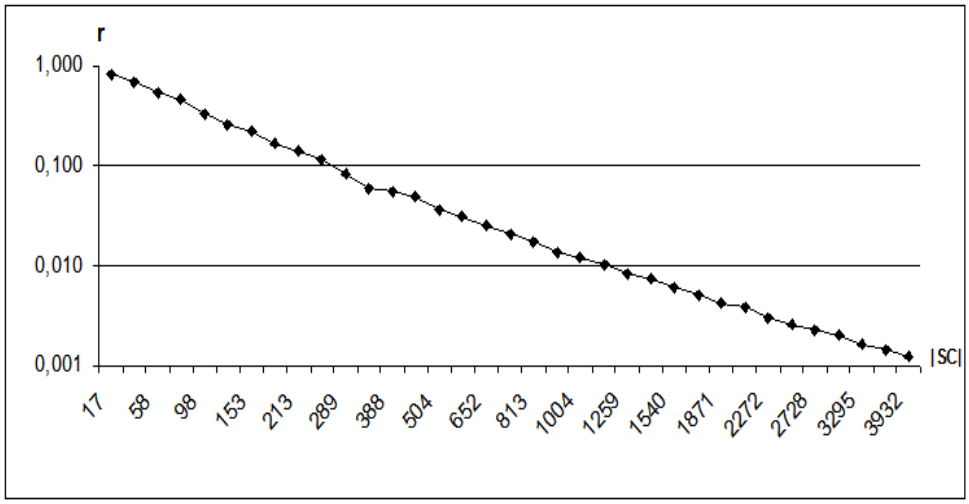

Fig. 1. Dependence of the share of active centroides in total number of centroides

As it can be seen, as $|\mathbf{SC}|$ grows from 2 to 40000, r gradually decreases from 0,9 to 0,0012. Average share of active centroides r_{av} is approximately 0,015. Thus, volume of distances computations must be reduced 50 – 100 times as compared with global algorithm of k-means. It should be noted, that the centres of clusters will not differ from those, obtained while application of global clusterization algorithm.

The above-mentioned gain in fast acting is obtained at the expense of considerable increase of memory expenditures, since for each vector it is necessary to store the distance to all centroides. However these expenditures can be considerably reduced, if we will store for each vector the distance not to all, but to m nearest centroides.

Method of k-means with computation of the distances to active centroides is realized in the following way:

1. We define the set of points $\{\mathbf{x}_0\} \subset \mathbf{S}$, which will be used for determination of initial position of new centroide insert.

2. Having assigned $k = 1$, we calculate the coordinates of the first centroide as average of all vectors coordinates:

$$c_{kj} = \frac{1}{n} \cdot \sum_{i=1}^{n} x_{ij}, j = 1, 2, ..., d.$$

3. We perform $k \leftarrow k + 1$ and find initial position of insertion of $\mathbf{c}_k$ centroide by means of vector $\mathbf{x} \in \{\mathbf{x}_0\}$ selection, that provides minimal distortion $E^2 = \sum_{\mathbf{x} \in S} D^2(\mathbf{x}, \mathbf{c})$.

4. We start k-means algorithm for k centroides, performing steps 4.1 – 4.3:

4.1 We divide the set $\mathbf{SC}$ into subsets $\mathbf{SC}^{(a)}$ and $\mathbf{SC}^{(p)}$, consisting of active and passive

centroides, correspondingly .

4.2 For each vector $\mathbf{x} \in S$ we define $\mathbf{W} = \{\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_m\}$, containing set from m closest to $\mathbf{x}$ centroides. For each $\mathbf{c}_i \in \mathbf{SC}^{(a)}$ we calculate the distance $r_i = D(\mathbf{x}, \mathbf{c}_i)$. If $\mathbf{c}_i \in \mathbf{W}$, we correct the corresponding value of distance r_i. If $\mathbf{c}_i \notin \mathbf{W}$, we check the execution of condition $r_i < r_{\max}$, where $r_{\max} = \max_j D(\mathbf{x}, \mathbf{c}_j)$ $j = 1, 2, ..., m$. If condition is executed, we add $\mathbf{c}_i$ to the set $\mathbf{W}$.

4.3 Using (1), we calculate new position of centroides. If the condition of convergence is not executed, we return to 4.1.

5. We check if the preset number of centroides $k_{\max}$ has been received. If $k < k_{\max}$, we return to 3.

The suggested method requires additional memory for storing the indices of the nearest centroides and distances to them for each vector m. Besides, to accelerate the verification of $\mathbf{c}_i \in \mathbf{W}$ the search of suitable element can be organized by means of hashing, that also requires additional memory.

As it is seen from Fig. 1, the value m must be greater for small k and smaller for larger values of k. However, small number of $m = 4$ allows to avoid increase of distortion and may recommended for practical application.

Application of kd-trees for determination of candidates for the selection of insert position of new centroide

One more method of decreasing of computational complexity is the usage of multidimensional binary trees of search (kd-trees) [11, 12]. kd-trees are built on the basis of input vectors, which closet centroides assign in the process of tree descent. Decrease of the amount of computations in this case can be reached due to the fact, that many input vectors can belong to the node of the tree. Thus, computations can be performed not for the given points, but for points, which belong to certain node of the tree simultaneously. Since kd-tree is determined for input vectors but not no necessity to up-date this data structure, i.e., its construction is performed once. It allows to store in the structure of the tree the information which is calculated in the process of tree construction and further can be useful: number of vectors, associated with the node of the tree and sum of vectors coordinates in the node.

In the given research kd-trees are used for determination of a candidate for initial position of the centroide. For each node of the tree the list of candidates for centroides is supported, the number of which in the process of descent along the tree will be reduced at the expense of exclusion of those which can not be closer for one of the points of the given node. If the node, which was reached is a terminal one (number of vectors in which does not exceed the preset value), then computations of distances to all the candidates for each of the points that are associated with the given node of the tree are carried out, and each point is matched up with certain centroide. It this case, the performance of the search is defined by the number of candidates in the list for the given node. In case, when only one candidate is associated with the node, it may be considered as the closest centroide for all the points of the node. As it is seen, in both cases considerable reduction of distances computations is observed. Hence, application of kd-trees allows to obtain further acceleration of centroide search operation, closest to the given vector, that is the main component of computations, performed at each iteration of k-means.

Experimental results

The research were carried out on the set of vectors of LSF-parameters [13], obtained from speech data base TIMIT, in the amount of n=50000 for dimensionalities d=5 and d=10. All the computations were carried out by Intel Core 22.0 GHz with the memory 2 GB.

The given research studied the influence of the number of vector-candidates in centroides on the quality of clusterization and fast acting of the algorithm. Two variants of the search of the suitable

position of i^{th} centroide were considered: on the base of vecto-candidate choice, that provides minimal distortion with recalculation of new position of centroides (RC), and without recalculation of centroides (WRC). In both cases (RC, WRC), the efficiency was evaluated by total distortion (Error) and time of algorithm operation (Time).

Table 1, and Fig 2 (1000 centroides) and Fig 3 (4000 centroides show the dependence of distortion and operation time on the number of candidates to centroides for both variants of i-th centroide choice (RC and WRC).

Table 1

Total distortion for different amount of candidates: *n*=50000

Quantity of centroides, *k*	Dimensionality	Choice of *i*-th centroide	Quantity of vectors candidate in centroides, |C|					
			1000	2000	4000	8000	16000	32000
1000	5	WRC	$2{,}642 \cdot 10^8$	$2{,}632 \cdot 10^8$	$2{,}623 \cdot 10^8$	$2{,}614 \cdot 10^8$	$2{,}610 \cdot 10^8$	
		RC	$2{,}628 \cdot 10^8$	$2{,}619 \cdot 10^8$	$2{,}611 \cdot 10^8$	$2{,}604 \cdot 10^8$	$2{,}598 \cdot 10^8$	
	10	WRC	$1{,}928 \cdot 10^9$	$1{,}919 \cdot 10^9$	$1{,}914 \cdot 10^9$	$1{,}910 \cdot 10^9$	$1{,}909 \cdot 10^9$	
		RC	$1{,}925 \cdot 10^9$	$1{,}915 \cdot 10^9$	$1{,}909 \cdot 10^9$	$1{,}905 \cdot 10^9$	$1{,}902 \cdot 10^9$	
4000	5	WRC			$1{,}279 \cdot 10^8$	$1{,}252 \cdot 10^8$	$1{,}238 \cdot 10^8$	$1{,}232 \cdot 10^8$
		RC			$1{,}262 \cdot 10^8$	$1{,}235 \cdot 10^8$	$1{,}220 \cdot 10^8$	$1{,}213 \cdot 10^8$
	10	WRC			$1{,}205 \cdot 10^9$	$1{,}185 \cdot 10^9$	$1{,}173 \cdot 10^9$	$1{,}164 \cdot 10^9$
		RC			$1{,}188 \cdot 10^9$	$1{,}167 \cdot 10^9$	$1{,}155 \cdot 10^9$	$1{,}146 \cdot 10^9$

It is seen from Table 1, that increase of the number of candidates for the choice of position of initial placement of new centroide allows to reduce distortion. Exponential character of the dependence of time on the number of candidates, as it is shown in Figs 2 and 3 takes place with the increase of the number of candidates more rapid growth of time is observed fro RC. Besides, while applying more rough estimation of centroides position by WRC, minor increase of distortion (approximately 1%) is observed, where as time is reduced by 5 – 20% (depending on the number of candidates) with the increase of dimensionality character of dependence remains, absolute values of distortion increase by the order. The conclusion can be drown, that the increase of the number of candidates allows to obtain only minor reduction of total distortion due to 2 times increase of the time.

That is why, greater efficiency can be obtained, using RC, with the number of candidates $|C| = 4 \cdot k$ (i.e. 4000 candidates for generation of 1000 centroides and 16000 candidates for obtaining 4000 centroides), since it allows to obtain less distortion without considerable slowing down of algorithm operation, as compared with WRC, where approximately the same value of distortion can be obtained at $|C| = 8 \cdot k$.

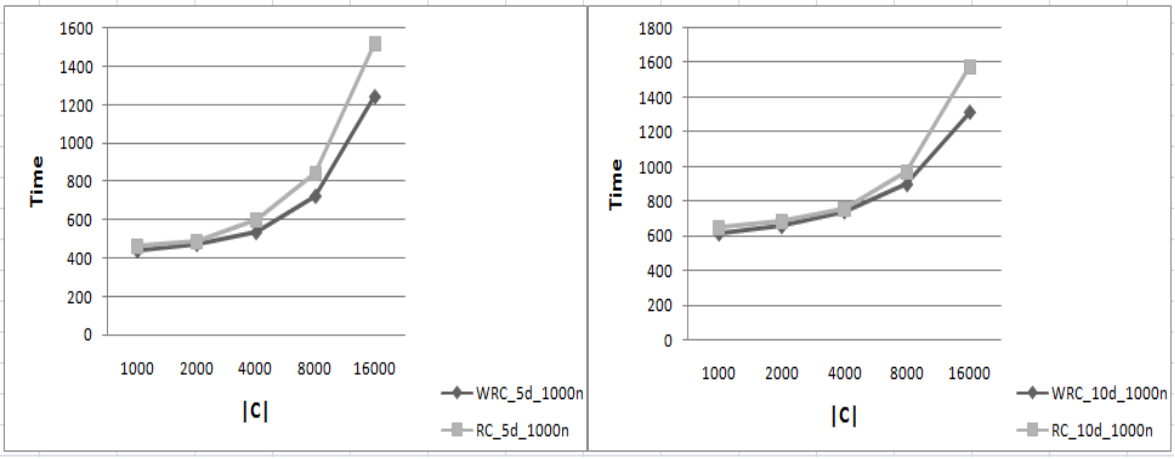

Fig. 2. Dependence of operation time on the quantity of input vectors

a) $d = 5, n = 1000$ b) $d = 10, n = 1000$

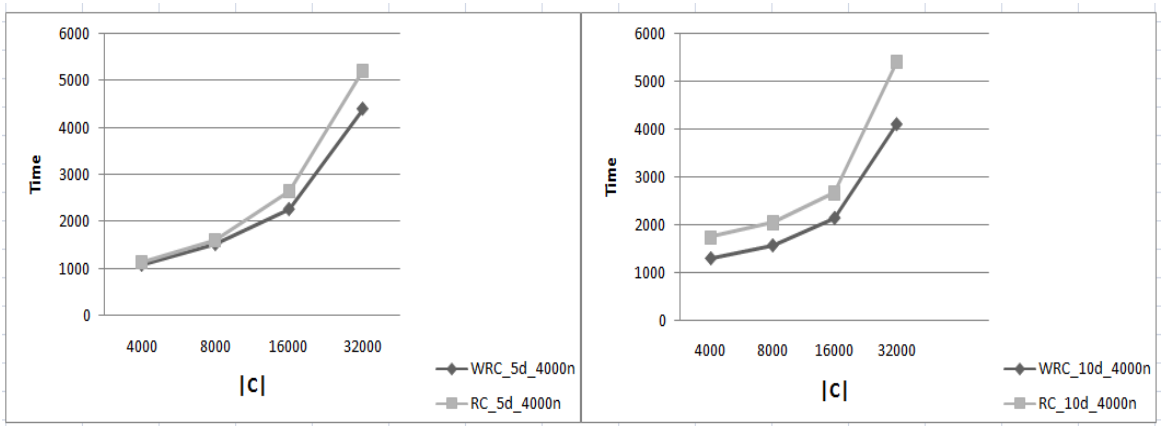

Fig. 3. Dependence of operation time on the quantity of input vectors

a) $d = 5, n = 4000$ b) $d = 10, n = 4000$

Algorithm of clusterization (RC at $|C| = 16000$), suggested in the given research, was compared with two modifications of k-means algorithm (see Fig 4): classic algorithm of k-means (further k-means), realized in MATLAB without limitation of the number of iterations and realized on the base of kd-trees with maximum number of iterations 500 (further hybrid), suggested in [12]. Characteristic feature of hybrid algorithm is that for approximation to global minimum it combines classic algorithm of k-means and local search (exchange between existing centroides and candidates to centroides on condition, that such exchange leads to decrease of average distortion).

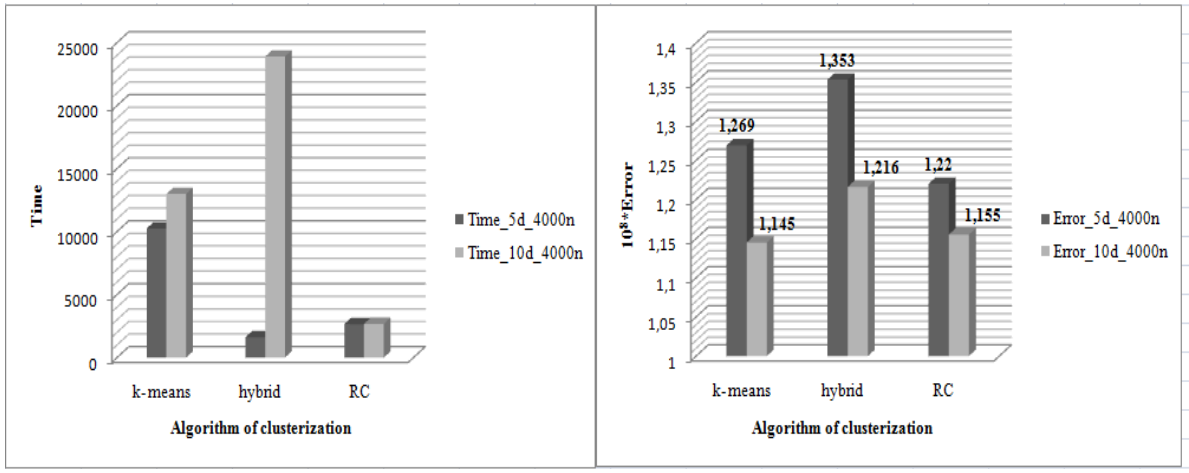

Fig. 4. Comparison of algorithms by total distortion (on the left) and operation time (on the right): $n = 4000$

Results, presented on the left in Fig 4, show that for both dimensionalities the largest value of total distortion is observed while applying clusterization algorithm, based on kd-trees, the smallest distortion allows to obtain RC for $d = 5$ and k-means (MATLAB) for $d = 10$.

On the right in Fig 4 the efficiency of the algorithms is evaluated by the time of operation. For $d = 5$ the largest fast-acting is achieved by the algorithm, using kd-trees, however, with the increase of dimensionality for hybrid rapid growth of operation time is observed, that is caused by exponencial character of time on dimensionality dependence. It should be noted, that the efficiency of the algorithm, suggested in the research, practically does not depend on dimensionality.

Conclusions

The improved method of k-means clusterization, suggested in the research, allows to obtain solution close to global minimum of distortion. Decrease of total distortion is up to 11% as compared with hybrid algorithm and up to 5% using k-means (MATLAB) algorithm. Concerning operation rate the suggested method gives the results 4-8 times better, than k-means (MATLAB) algorithm for $d = 5$ and $d = 10$ dimensionalities 9 times better than hybrid algorithm for $d = 10$, in case, when $d = 5,$ it is slightly yields to hybrid regarding fact acting (1,6 times). We should note, that the efficiency of the suggested method practically does not depend on dimensionality. Reduction of clusterization error is achieved at the expense of improvement of selection procedure of vector-candidates for the insert of new centroide (RC) and determination of optimum quantity of candidates, which is $| C |= 4 \cdot k$ and leads to slowing down of algorithm of operation 1,5 – 2 times.

REFERENCES

1. Fayyad U. M. Advances in Knowledge Discovery and Data Mining / U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy // AAAI/MIT Press. – 1996. – 611 p.

2. Gersho A. Vector Quantization and Signal Compression. / A. Gersho, R. M. Gray // Boston: Kluwer Academic. – 1992. – 760 p.

3. Duda R. O. Pattern Classification and Scene Analysis / R. O. Duda, P. E. Hart // New York: John Wiley & Sons. – 1973. – 512 p.

4. Jain A. K. Algorithms for Clustering Data / A. K. Jain, R. C. Dubes // Englewood Cliffs, N.J.: Prentice Hall. – 1988. – 334 p.

5. Arthur D. k-means++: The advantages of careful seeding / D. Arthur, S. Vassilvitskii // ACM-SIAM Symposium on Discrete Algorithms (SODA 2007) Astor Crowne Plaza – New Orleans, Louisiana. – 2007. – P. 1027 – 1035.

6. Lai Jim Z. C. Fast k-means clustering algorithm using cluster center displacement / Jim Z. C. Lai, Tsung-Jen Huang, Yi-Ching Liaw // Pattern Recognition. – 2009. – No 11, vol. 42. – P. 2551 – 2556.

7. Kanungo T. An Efficient k-means clustering algorithm: analysis and implementation / T. Kanungo, D. M. Mount, N. S. Netanyahu, C. Piatko, R. Silverman and A.Y. Wu // IEEE Transactions On Pattern Analysis And Machine Intelligence. – 2002. – No. 7, vol. 24. – P. 881 – 892.

8. Likas A. The global k-means clustering algorithm / Aristidis Likas, Nikos Vlassis, Jacob J. Verbeek // Pattern Recognition. – 2002. – No 2, vol. 36. – P. 451 – 461.

9. Refining initial points for KMeans clustering : (Conference on Machine Learning) [Електронний ресурс] / P. S. Bradley, U. M. Fayyad // Proceedings of Fifteenth Intl. – 1998. – P. 91 – 99. / Режим доступу: ftp://ftp.research.microsoft.com/pub/tr/tr-98-36.pdf.

10. Hussein N. A Fast Greedy K-Means Algorithm / Master's Thesis Nr:9668098 N. Hussein. – University of Amsterdam Faculty of Mathematics, Computer Sciences, Physics and Astronomy Euclides Building Plantage muidergracht 24. – 2002. – p. 62.

11. Moore Andrew William Efficient memory based learning for robot control / Moore Andrew William. – PhD thesis Nr: UCAM-CL-TR-209. – 1990. – p. 248.

12. Kanungo T. A Local Search Approximation Algorithm for k-Means Clustering / T. Kanungo, D. M. Mount, N. S. Netanyahu, C. Piatko, R. Silverman, A. Y. Wu // Computational Geometry: Theory and Applications. – 2004. – No 2. – P. 89 – 112.

13. Ткаченко О. М. Ефективне векторне квантування LSF-параметрів при ущільненні мовних сигналів / О. М. Ткаченко, О. Д. Феферман, С. В. Хрущак // Інформаційні технології та комп'ютерна інженерія. – 2007. – № 1. – С. 124 – 129

Tkachenko Oleksandr – Cand. Sc. (Eng.), Assistant Professor, Department of Computer Engineering, e-mail: ant@vstu.vinnica.ua.

Grijo Tukalo Oksana – Student, e-mail: xxmargoxx@gmail.com.

Dzys Olexiy – Student, e-mail: alexdz47@gmail.com.

Lakhovets Sergiy – Student, e-mail: selema.tsx@gmail.com
Vinnytsia National Technical University.