Tamara SAVCHUK[1], Anton SAKALIUK[1], Waldemar WÓJCIK[2], Aron BURLIBAY[3]

Vinnitsa National Technical University (1), Lublin University of Technology (2), Kazakh National Technical University after K. I. Satpaev (3)

Transformation of "user-object" matrix for the collaborative filtering

Abstract. *The paper is devoted to application of collaborative filtering that is one of the method of automatic data filtering in the Internet. The main disadvantage of the approach is the necessity of performing a large number of operations. The authors have presented a mean of overcoming this problem by reduction of the dimension of the input matrix. Experimental results show that it had led not only to reduction of computational time, but also increased the accuracy of recommendations obtained.*

Streszczenie. *Artykuł poświęcony jest filtrowaniu kolaboracyjnemu, które jest jedną z metod automatycznej filtracji danych w sieci Internet. Główną wadą wspomnianego podejścia jest konieczność wykonywania bardzo dużej liczby operacji. Autorzy przedstawili rozwiązanie tego problemu polegający na redukcji wymiarowości przetwarzanej macierzy. Rezultaty badań pokazują, że oprócz zmniejszenia czasu obliczeń, uzyskano poprawę dokładności uzyskiwanych rekomendacji.* (***Przekształcenie macierzy „user-object" w filtrowaniu kolaboracjnym***)

Keywords: Data mining, sparse matrices, recommender system, collaborative filtering.
Słowa kluczowe: Eksploracja danych, macierze rzadkie, systemy rekomendujące, filtrowanie kolaboracyjne.

doi:10.12915/pe.2014.01.13

Introduction

The rapid growth of information in modern Internet pushes to development of new tools of providing, searching and systematization of information. Usage of classic tools of search and interaction with the information no longer meets the growing demands of its users neither in terms of convenience nor confidence. To a greater extent this applies to commercial information and information about the preferences of users.

The method of content-based filtering with content analysis is commonly applied to solve a problem. But this method has particular disadvantages [1]:

1. Objects must be in an accessible to machine processing form.

2. Technology with content filtering doesn`t have an installed method for generating random preferences.

In order to solve these challenges, new methods for structuring and filtering data are considered in this article.

Collaborative filtering is a method that enables the automatic filtering of data on request of a user using the collected information about the preferences of other users (collaborating with each other) for this data [2].

Recommendation system based on collaborative filtering - a convenient alternative to classical search algorithms, as well as using factors that can not be obtained from the technical analysis. Therefore, the implementation of such mechanisms in websites increases the speed of finding relevant information, increases its completeness and accuracy.

Powerful tools of recommendation systems, widely used in various industries, help to identify items that are offered by experts to users based on their overall popularity, demographic characteristics and behaviour analysis, which determine the relevance of improving algorithms for filtering defined systems.

Expert filters cannot always process the amount of modern information. Collaborative technology can harmonize this imbalance: whatever rate has generation of content, it counteracts the equally productive mass selection [3].

Collaborative filtering allows to establish parity between user and producer awareness about products' quality, blocking tendencies of selection worsening, evolving under conditions of information asymmetry.

Analysis of various known facilities of collaborative filtering, allows us to assert the relevance of the task of developing an information system with the following requirements [3]:

- easiness of configuration for application on different types of objects;
- scalability;
- extensibility (ability to introduce additional factors in order to improve the quality of analysis);
- versatility (the ability of the system to issue precise recommendations not depending on the type of evaluated object).

  The scalability and extensibility requirements can be achieved by transformation the original matrix, which includes smoothing and singular value decomposition to reduce the size of data to be performed.

Main problems of collaborative filtering

Among main problems that accompany the process of collaborative filtering, we should consider the following:

1. Problems at the stage of "cold start" [4], which include:

- problem of a new user. New users are out of touch with current ones and thus cannot receive recommendations. The system should teach user preferences on the basis of its estimates to generate precise recommendations. Several methods were proposed are to solve this problem. Most of them use hybrid recommendation approach that combines thematic and collaborative algorithms. These techniques are used in strategies based on the popularity of objects, their entropy, personalization of users and combinations of these techniques;
- problem of a new object. New items are regularly added to the recommendation system. Collaborative system when formulating recommendations are guided only by users' needs. Therefore, the recommendation system can not recommend the object until it has received enough ratings.

2. Ratings sparsity [2]. In any recommendation system, the number of evaluations that need to be predicted exceed the number of issued ratings. The system should predict ratings based on the minimal number of objects and users. The solution of sparsity of ratings can go through the use of user profile when searching for similar measure. Collaborative filtering is based only on correlations between users. Thus, recommendations for an active user base on the ratings provided by similar users. In the case of high

sparse matrices (> 90% zero elements), it is very difficult to find a correlation between users.

3. Scalability. In collaborative recommendation systems, users with similar preferences are determined based on memory-based or model-based approaches. In memory-oriented approach based on active users than all users, therefore computational complexity increases with increasing number of users. In model-oriented approach existing users are grouped basing on their similarity [5]. In formulating recommendations, similar groups are grouped together and only then are compared inside the cluster. This approach doesn't have a tremendous influence on solving the problem of scalability. Traditional approaches do not perform previous offline computation, and therefore the number of calculations increases directly with the number of users and objects. The algorithm cannot be used on large datasets, unless it uses dimensionality reduction by reducing the quality of recommendations [6].

Problem definition

Formally, the problem of making recommendations can be represented as following:

Let's say, that C is the set of users, S - set of proposed objects. Capacity of the set of proposed objects S and set of users C can reach hundreds of thousands or even millions of units. Utility function u_{ij} describes the utility of the object s_j for the user c_i, where i - user number, j - serial number of the object [4]:

$$u_{ij} : c_i \times s_j \to R, \tag{1}$$

where R - the number of ordered objects ($R \in N$).

For each user $c_i \in C$ an object $s_j \in S$ is selected for which the maximum value of utility for the user:

$$\forall c_i \in C, s_j = \arg\max_{s_j \in S} u_{ij}. \tag{2}$$

Systems use different approaches for calculating the measure of similarity between users in collaborative recommendation. In most of them, the similarity between two users is determined on how they rate objects.

The method uses user's filtering shows sufficient accuracy for practical applications. However, the drawback of all the algorithms of this method is its dependence on resource (memory requirements) and complexity (number of calculations required to obtain recommendations).

If we store ratings for vectors of all transactions in memory (for quick access), i.e. the matrix of n rows by m columns, then the average system (~ 1 million transactions and about 10 thousand objects) will need to be stored in memory ~ 10 billion reals (8 bytes). Of course it is possible to provide access to the data from external data carriers, but it strongly slows operations process and, therefore, increases the hardware requirements that ensure a sufficient level of speed of access to information and tempo of operations [7].

Each comparison of this transaction with one of the n other takes $O(m)$ operations, it is necessary to perform

$$n \times O(m)$$

operations to determine the measure of similarity of the transaction with others, where n – the number of transactions, m – the number of objects in each transaction ($m, n \in N$).

Calculating the rating for each of the m products will require implementation of $O(n)$ operations averaging transactions, i.e.

$$m \times O(m)$$

operations for all products.

Total conduction will be in following:

$$O(m \times n)$$

operations for ratings of all products for this transaction to get recommendations.

The above considerations suggest that the traditional filtering data about users can be applied only to relatively non cardinal databases.

Unlike filtering algorithm data about users, filtering algorithm data about products measure of similarity of the analyzed product to all other products that can be calculated in deferred mode on a schedule because vectors of rankings of all products are available by the time of recommendation development.

Thus, after dividing the process of making recommendations on deferred stage (computing the measure of similarity products to each other) and the stage in real time (computing product ratings), we find that the complexity of the algorithm filtering data on products at the stage of recommendations is

$$O(m^2),$$

in contrast to the complexity of filtering data transaction does not depend on the number of transactions.

So, if the number of transactions is much higher than the number of products, the filtering algorithm of data about products is more efficient in terms of time forming recommendations than filtering algorithm of data about transactions due to the possibility of deferred data pre-processing.

However, data filtering neither of users, nor of products can be effective in high-load systems. The use of mathematical statistics for the task of making recommendations in complex system is the most applicable. But there is a problem with the amount of data because statistical methods work well when there is a large amount of a priori information and in recommendation system a priori information is limited. These statistical techniques cannot guarantee a successful outcome.

Among heuristic methods for searching recommendations genetic algorithms can be used. Generally, data mining is not the main application area of genetic algorithms. They need to be seen more as a powerful tool for solving combinatorial and optimization problems. However genetic algorithms can be adapted to solve this task as well [8].

However, genetic algorithms have several disadvantages. Criteria of chromosome selection and the whole procedure are heuristic and do not guarantee finding the optimal solution. Also, the evolution may "loop" on any unproductive sector. This is particularly noticeable in solving high dimension problems with complex internal connections.

Another way to solve the task of making recommendations may be neural networks that can be trained on existing data set. In this case the initial information is the amount of object's ratings, and as the target field is set of recommended objects.

One of the methods of data mining is the search for association rules. Detecting rules of association is the process of identifying frequent item sets of user-selected objects. The form "if A and B then D with probability x" has generated based on these sets of rules. On the basis of the existing rules tabulated ratings are recommended to the objects that are found on the right side of the rules if the objects from the left side are already in the set. Therefore, an approach based on association rules is similar to filtering

of objects especially in the option of "frequency pair entry." The difference is that in the algorithm of association rules, the rules are formed in deferred mode on schedule and search for recommendations is conducted in real time on the basis of received rules. In filtering by products the weights are calculated, which describes the frequency of entry for each pair of vector object in real time in forming recommendations is much more time consuming. On the other hand, the weakness of association rules algorithm lies in that for not every set of objects in the analyzed transaction there is a corresponding rule with sufficient support and reliability. For example, there are rules for type "$A, B \rightarrow D$" for each pair of objects A and B to the existence of partial sets (a subset of 3 elements in the set of n elements, each subset of 3 elements can form 3 rules depending on whether item will be on the right side of this rule). For n = 1000, the number of partial sets in this case should be 498 501 000, that at limitation on the frequency set in 10 transactions leads to the necessity of keeping at least 5 billion transactions, which is a crucial requirement. This problem is solved by finding rules that contain only one item on the left side, i.e. type "if A, then B», for each object *A* in an existing transaction. However, this simplification leads to lower accuracy of produced recommendations [8].

Clustering algorithms have a better scalability than conventional collaborative filtering algorithms, because they make predictions in much smaller clusters, and not for the entire customer base. Online classification of user cluster is almost as resource consuming, as the search of similar customers through collaborative filtering. Using clustering on high-capacity datasets is unfeasible, most programs use different forms of greedy methods for constructing clusters. For high-capacity datasets, especially with high dimensionality there is a necessity of sampling or dimension reduction.

Smoothing of the "user-object" matrix

Finding correlated users in sparse matrices is a very resource consuming task. Therefore "User-Object" matrix should be smoothed which means removing noise from a matrix, allowing important patterns to stand out. Smoothing is performed using the radial basis function (Radial Basis Function Networks, RBFN).

Let us have:

1. Group of M users $\{u_i \mid i = 1,2,...,M\}$.

2. Group of N different objects $\{s_j \mid j = 1,2,...,N\}$.

3. Rating table r_{ij} - a matrix $[M \times N]$ that contains ratings of the i-th user's on j-th object. Unrated objects represented as zero values.

Neural network based on radial basis function is a family of artificial neural networks. It has three layers: input layer, hidden layer and output layer. The input layer contains *M* neurons to which the input is user's vector-ranking. This layer is fully connected to all neurons in the hidden layer. Each neuron in the hidden layer has a function of activation. Hidden layer is fully connected to the output layer. The output layer contains M neurons, user's rating vector is smoothed to it. The output layer performs a simple function of the summation of the users.

Technology of radial basis function is based on picking function F [7]:

$$F(X_i) = \sum_{k=1}^{K} w_k \phi(\| X_i - C_k \|), \quad (3)$$

where w_k - the vector of weight from the hidden layer to the output layer; X_i - given set of points; C_k - the center of the set of points; ϕ - function of activation.

We can use three different activation functions (4-6):

1) Gaussian function:

$$\phi(r) = \exp(-\frac{r^2}{2\sigma^2}) \text{ for } \sigma > 0. \quad (4)$$

2) Multicriterial function:

$$\phi(r) = r^{\beta}, \quad (5)$$

where β - a positive odd number.

3) Smoothing Thin-plate function:

$$\phi(r) = r^k \log(r), \quad (6)$$

where $k > 0$, $r = \| X_i - C_k \|$; σ, β, k - positive parameters.

Smoothing algorithm converts sparse matrix of user's ratings r_{ij}. For its work also need to determine $\phi_j^{\max}, \phi_j^{\min}$ - maximum and minimum of the activation function for object j and r_{ij} - rating of object j from user i.

1. $range = (\max\_rating - \min\_rating) + 1$.

2. Find the number of clusters k so that $\frac{range}{k} \leq 3$.

3. For each j:

3.1. Separate users into k clusters.

3.2. Calculate centers

$$C_k = \frac{\sum_{p=1}^{k_1} r_{ip}}{k_1}, \quad (7)$$

where k_1 - the number of users who belong to this cluster.

3.3. Calculate the matrix of Euclidean distances

$$g_{ip} = \left\| r_{ip} - C_k \right\| \text{ where } 1 \leq p \leq k1. \quad (8)$$

3.4. Calculate the activation function $\phi_{ip}(g)$.

3.5. Calculate weights using function of pseudorandom weights:

$$\omega_i = \frac{(\phi_I^{\max} - \phi_I(r_{iI}))/(\phi_I^{\max} - \phi_I^{\min})}{\sum_{x=1}^{n} (\phi_{I,x}^{\max} - \phi_{I,x}(r_{xI}))/(\phi_x^{\max} - \phi_x^{\min})}, \quad (9)$$

$1 \leq i \leq M$.

3.6. Calculate $r_{ij}^{'} = F(r_{ij})$ with formula (3).

Reducing the dimension of "user-object" matrix

The main disadvantage of collaborative filtering algorithms is the need to perform a large number of operations to calculate the measure of similarity of products or transactions and for averaging product's ratings in predicting unknown rating. To reduce the complexity of averaging operations, system will not use averaging data on all transactions and on all products, but only on K most similar. The general trend is to increase the accuracy during the initial increasing of the number of K, and then, after

reaching the maximum, accuracy stabilizes or smoothly worsens. Dilution of precision at further increasing K is explained by the fact, that an increasing number of "dissimilar" transactions or products are taken into consideration. Then consideration of only K closest transactions or products instead of all available and existing ones not only accelerates the process of calculating the unknown rating, but also increases the accuracy of prediction [9].

To reduce the complexity of calculating the measure of similarity of vectors of products or transactions, the approach of reducing the dimension of the matrix products-transactions is used, based on the decomposition of this matrix by singular value. Decomposition in the singular value (Singular Value Decomposition, SVD) is a matrix $A \in Mat(n,m)$ representation with rank

$$r = ran(A) \le \min\{n,m\} \tag{10}$$

as

$$A = USV^T, \tag{11}$$

where the matrixes $U \in Mat(n,r)$ and $V \in Mat(m,r)$ consist of orthonormal columns [5], which are eigenvectors with nonzero eigenvalues of matrices $\mathbf{AA}^{\mathrm{T}}$ and $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ respectively, and

$$S = \begin{Bmatrix} \lambda_1 & 0 & ... & 0 \\ 0 & \lambda_2 & ... & 0 \\ ... & ... & ... & ... \\ 0 & 0 & ... & \lambda_r \end{Bmatrix} \in Mat(r,r) \tag{12}$$

is diagonal matrix with positive diagonal elements, sorted in descending order.

Diagonal elements $\lambda_1, \lambda_2, ..., \lambda_r$ of the matrix **S** is the eigenvalue corresponding to the nonzero eigenvectors $\mathbf{AA}^{\mathrm{T}}$ and $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ (columns of **U** and **V**). The columns of the matrix **U** is therefore orthonormal basis column space of the matrix **A**, and the columns of the matrix **V** – orthonormal basis space rows of the matrix **A**. An important property of SVD-decomposition is the fact, that if for $d < r$ to transform matrix **S** into a matrix consisting only of d largest diagonal elements and leave only the first d columns in the matrix **U** and **V**, i.e. transform them into

$$U_d \in Mat(n,d) \text{ and } V_d \in Mat(m,d), \tag{13}$$

then the matrix

$$A_d = U_d S_d V_d^T, \tag{14}$$

will be the best approximation of a matrix **A** of all matrices of rank d [10].

$$S = \begin{Bmatrix} \lambda_1 & 0 & ... & 0 \\ 0 & \lambda_2 & ... & 0 \\ ... & ... & ... & ... \\ 0 & 0 & ... & \lambda_d \end{Bmatrix} \in Mat(d,d). \tag{15}$$

The main stages of dimensionality reduction using SVD-decomposition of the matrix **A** transactions, products are as follows [10]. At first constructed decomposition $A = USV^{-1}$ (11), then for a fixed chosen $d << ran(A)$ get the best d-rank approximation of the matrix **A** in the form

$$A \approx A_d = U_d S_d V_d^T. \tag{16}$$

When filtering by products each j-th column Y_j of the matrix A, which corresponds to ratings of j-th product is approximated by j-th column of the matrix A_d, which is a projection of the vector Y_j on space, formed by d orthonormal columns of matrix $\mathbf{U}_d$ coefficients of expansions $C_j = (S_d V_d^T)_j$, corresponding to the j-th d-dimensional column vector of the matrix. So instead of n-dimensional vector of j-th product Y_j, d-dimensional vector C_j is considered, which is a vector of coefficients of Y_j projection decomposition in the basis U_d. Using the described approach, to determine the measure of similarity vectors products and Y_u and Y_k, the measure of similarity of their d-dimensional approximations is calculated.

Unlike the traditional approach of calculating the similarity of all products, in the proposed approach the number of operations for calculating the measure of similarity between vectors of products is $O(d)$ unlike $O(n)$, which speeds up the computation when $d << n$.

After performing smoothing and transformation of the "User-Object" matrix, it's possible to perform the process of collaborative filtering algorithm. It involves the following main steps:

1. If no rated objects (cold start problem) then send T popular objects from each cluster recommendation agent.
2. Otherwise, determine the positive and negative neighbouring clusters using Pearson correlation function [8]:

$$sim_{i,j} = \frac{\sum_{t \in r_{i\Delta r_j}} (r_{it}^{'} - \overline{r_i^{'}})(r_{jt}^{'} - \overline{r_j^{'}})}{\sqrt{\sum_{t \in r_{i\Delta r_j}} (r_{it}^{'} - \overline{r_i^{'}})^2} \sqrt{\sum_{t \in r_{i\Delta r_j}} (r_{jt}^{'} - \overline{r_j^{'}})^2}}. \tag{17}$$

3. Identify the nearest positive and negative neighbours with correlated clusters using the cosine similarity function:

$$csim_{i,j} = \frac{\sum_{k=1}^{N} r_{ik} \cdot r_{jk}}{\sqrt{\sum_{k=1}^{N} r_{ik}^2 \cdot \sum_{k=1}^{N} r_{jk}^2}}. \tag{18}$$

4. Prediction of ratings for nonrated objects using prediction function. Then choose a subset of K most similar users based on their similarity to the active user. Weighted average deviation from the neighbour [9]:

$$P_{ui} = \overline{r_i} + \frac{\sum_{m=1}^{c} (r_{mi}^{'} - \overline{r_m^{'}}) \cdot csim_{u,m}}{\sum_{m=1}^{c} csim_{u,m}}. \tag{19}$$

5. Performing step 4 for a subset of least similar users.
6. Let X be the set of recommended objects predicted on the basis of positive nearest neighbours and Y, as a set of recommended objects predicted on the basis of negative nearest neighbours.
7. Calculate $Z = X - Y$. The set Z is sent to the recommendation agent.
8. Recommend a set of objects Z to an active user.

Experimental results

For experiment was used publicly available dataset MovieLens which consists of 100,000 ratings by 943 users on 1682 movies. The results of the research (see fig. 1), rank of the approximation matrix d affects the accuracy of the resulting prediction. This number should be small

enough to significantly affect the acceleration performance computing and minimize retraining on the one hand, and large enough to hold important objective relationship between the transaction and the products contained in the original data [5].

The accuracy of prediction varies according to the rule: an increase in the number of prediction accuracy of d increases rapidly and reaches its maximum (about d = 6 in average), and accuracy deteriorates. The reason for the deterioration of accuracy of prediction with increasing rank approximating of the matrix due to retraining (unnecessary complication) model, which does not lead to the discovery of objective relationships between products and transactions, and to the training data.

Therefore, the use of only a limited number of the most similar products and transactions, and transactions-of the matrix approximation matrix products significantly lowers rank not only simplifies the calculations, but also increases the accuracy of prediction by reduction of factors retraining of model [11]. Minor changes in the original “user-object” matrix do not impact on the calculated rank, so the rank recalculation to be performed is not often (per each request), but periodically, depend on filling of the matrix with new data.

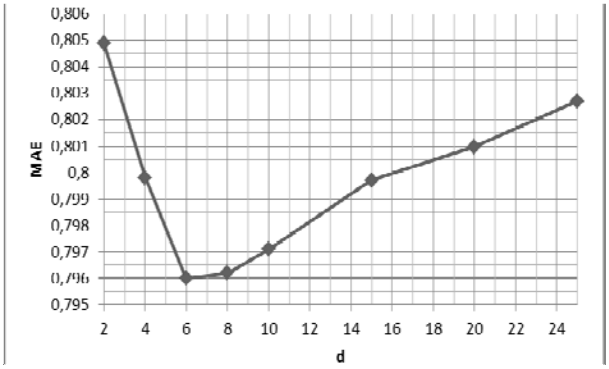

Fig.1. Dependence of the value of rank *d* approximation on the accuracy of prediction

Determine accuracy making recommendations by using the mean absolute error (MAE) rating prediction [12]. Comparable results of collaborative filtering using dimensionality reduction and without it are given in table 1.

Table 1. Mean absolute error for collaborative filtering system

	CF without reducing the dimension	CF after dimension reduction
MAE	0,8398	0,7965

Reducing of dimension not only reduced the time for processing information on products and transactions, but also increased the accuracy of recommendations by 5.43%.

Conclusions

The main limitations for using of known Collaborative filtering methods are the problems of scalability and sparsity of ratings that do not allow making recommendations. These problems can be solved by reducing the dimension of the input matrix.

Analysis of the results of the research showed that by reducing the dimension of the input of the "User-Object" matrix increases the speed of computing user’s data and their preference’s data, and also increases the accuracy of making recommendations by sifting unimportant information (the mean absolute error in the formulation of recommendations decreases by 5.43%). Thus, collaborative filtering with the previous dimension reduction of input data enhances the quality of recommendations to users.

REFERENCES

[1] Savchuk T.O., Sakaliuk A.V., Use of cluster analysis for the improvement of an algorithm for collaborative filtering, *Visnuk Khmelnytskogo natsionalnogo universytetu*, nr 1 (2011), 186-192

[2] Xiaoyuan Su., *A Survey of Collaborative Filtering Techniques*, Hindawi Publishing Corporation USA (2009)

[3] Segaran T., *Programmed collective mind,* Simvol-Plus (2008)

[4] Marlin B., *Collaborative Filtering a Machine Learning Perspective*, National Library of Canada (2004)

[5] Mobasher B., Recommender systems, *Kunstliche Intelligenz, Special Issue on Web Mining*, vol. 3 (2007), 41–43

[6] Billsus D., Pazzani M.J., Learning collaborative information filters. *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, (2007), 46-53

[7] Deshpande M., Karypis G., Item-based top-n recommendation algorithms. *ACM Trans. Inf. Syst.*, 22(1), (2004), 143–177

[8] Rudenko O.G., Bodianskyi E.V., Artificial neuron networks, *Kharkiv* (2005)

[9] Kogan J., Nicholas C., Teboulle M., Clustering Large and High Dimensional Data electronic resource. *Access mode: http://www/csee/umbc/edu/nicolas/clustering/tutorial.pdf*

[10] Savchuk T.O., Sakaliuk A.V., Decreasing the matrix dimentions «User-Object» during the collaborative filtering, *Proceedings of the international conference «Internet – Education – Science 2012»*, Vinnytsia (2012), 45

[11] Ning Ye, Shuo Zhang, Xia Huang, Jian Zhu, Collaborative Filtering Recommendation Algorithm Based on Item Clustering and Global Similarity, *Fifth International Conference Business Intelligence and Financial Engineering (BIFE),* (2012), 69-72

[12] Weiliang Kong, Qingtang Liu, Zhongkai Yang, Shuyun Han, Collaborative Filtering Algorithm Incorporated with Cluster-based Expert Selection, *National Engineering Research Center for E-learning, Central China Normal University*.

***Autors**: Ph.D. Tamara Savchuk, professor of computer science department, Vinnitsa National Technical University, Ukraina, tel. +380664124037, E-mail: savchtam@rambler.ru; M.Sc. Anton Sakaliuk, Vinnitsa National Technical University, Ukraina, tel. 00380965554604, E-mail: santei90@gmail.com; Prof. Waldemar Wójcik, Lublin University of Technology, Poland, tel. +48815384309, E-mail: waldemar.wojcik@pollub.pl; M.Sc. Aron Burlibay, Kazakh National Technical University after K. I. Satpaev, Ałmaty, Kazakhstan*