UDC 004.415.2
DOI https://doi.org/10.32838/2663-5941/2021.1-1/21

Movchan K.O.
Ukrainian Scientific and Research Institute of Special Equipment
and Forensic Expertise of the Security Service of Ukraine

Oleshchenko L.M.
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"

CLUSTERING OF INTERNET USERS SEARCH QUERIES USING GRAPH THEORY

One of the principles of Internet marketing is to focus on the customer in relation to his needs, characterized by search queries on the Internet. Building the semantic core of the site solves the problem of distributing user searches on the pages of the web resource. When working with the semantic core, we need to determine which page best matches a particular search query or group of user queries. By analyzing the characteristics of user clusters, we can better understand potential customers and provide users with more relevant web documents. Clustering is used to identify and classify subgroups of Internet users based on their behavior. Each clustering method uses different criteria to group data objects. This paper discusses the features of using k-means, EM-algorithm and Kohonen neural networks to cluster search queries. The method of clustering user searches using graph theory is considered in more detail. Nodes represent elements to be clustered, and the weights at the edges connecting the two nodes indicate the distance (dissimilarity) between objects. After applying the algorithm, the associated components indicate which objects the clusters belong to objects whose nodes are connected by edges are in the same cluster. It is shown that the proposed approach is more efficient in comparison with the basic clustering algorithm. For this research we used a sample of 5,000 search queries and the number of clusters k = 40. The change in results between runs comes from random initialization in the first step of the k-means algorithm. Studies show better convergence results by presenting data (web documents) for clustering in the form of graphs instead of vectors. The disadvantage of this solution is the use of more memory than in the basic method, and the problem of choosing the initial number of clusters is not solved.

Key words: *clustering algorithms, graph theory, k-means, Euclidean distance, graph vertices, graph measure of distance, convergence.*

Problem statement. The problem of clustering search queries of web users is to divide the sets of search queries into clusters so that the queries in one cluster were more similar to each other than in other clusters [1]. Existing software solutions used for this tasks use semantic clustering algorithms to recognize thematically related web pages and have a number of disadvantages. Most of these solutions rely on text analysis of the content of web documents, which leads to certain limitations, such as long processing time, the need for representative textual content or blurring of natural languages. The main problem when using these algorithms is the choice of the number of clusters, which is usually chosen at random. The purpose of this research is to analyze and improve existing methods of clustering user search queries.

Related research. Analysis of existing work has shown that there are many methods of clustering web users. However, applying these methods directly to simple datasets is not effective enough, as web servers typically contain thousands or even millions of pages, and Internet users can access web pages for a variety of purposes [2–4].

Analysis of the main methods of clustering search queries. Existing clustering methods that can be used to solve this problem are divided into three main categories: graph clustering algorithms, statistical clustering algorithms and hierarchical clustering algorithms. Two forms of graph clustering can be performed on data in the form of graphs. Vertex clustering tries to combine the nodes of the graph into groups of tightly connected regions based on either edge weights or edge distances. The second form of graph clustering treats graphs as objects to be clustered and clusters these objects based on similarity. The second approach is often found in the context of structured or XML data. Data for clustering can be represented as a graph, where each element is represented as a node, and the distance between the two elements is simulated by a certain weight at the boundary connecting the nodes. In graph clustering, elements inside a cluster are connected to each other,

but have no connection with elements outside that cluster. Some important approaches to graph-based clusters are adjacency-based clusters. The essence of such algorithms is that the sample of objects is represented as a graph $G=(V,E)$. The main feature of clustering graphs is the lack of distance between two arbitrary points in space, because there is no space itself, no norm, and it is impossible to determine the distance. Instead, there are edge metadata. If there is a "weight" of an edge, then it can be interpreted as a distance and then determine the distances for each pair of vertices. Many of the clustering algorithms in Euclidean space are also suitable for graphs, since these algorithms only need to know the distance between observations and not between arbitrary "points in space". Graphs have many of their unique properties that can also be used, such as connectivity components, local edge clusters and information flow loops. The vertices of the graph correspond to the objects of the sample, and the edges correspond to the pairwise distances between the objects $\rho_{ij}=(x_i,x_j)$. The advantage of graph-based clustering algorithms is clarity, relative ease of implementation, and the possibility of various improvements based on geometric considerations.

The main algorithms are the algorithm for the selection of connected components, the algorithm for constructing a minimal skeletal tree and the algorithm for layer-by-layer clustering. The disadvantages of this algorithm include limited applicability and poor control of the number of clusters. Statistical clustering algorithms include *k*-means and EM-algorithm. The basic idea of *k*-means is that at each iteration the center of mass is recalculated for each cluster obtained in the previous step, then the vectors are divided into clusters again according to which of the new centers was closer to the selected metric.

This algorithm does not guarantee the achievement of the global minimum of the total square deviation, but only one of the local minima, the result depends on the choice of source centers of clusters, their optimal choice is unknown, we must know the number of clusters in advance.

EM-algorithm is used to find estimates of the maximum plausibility of the parameters of probabilistic models, in the case where the model depends on some hidden variables. Each iteration of the algorithm consists of two steps. In the E-step (expectation) the expected value of the likelihood function is calculated, and the hidden variables are considered as observable. At the M-step (maximization) the estimate of the maximum likelihood is calculated, thus increasing the expected value of the likelihood calculated at the E-step. This value is then used for the E-step on the next iteration. The algorithm is executed to convergence. An auxiliary vector of hidden variables G, which has two properties, is artificially introduced. On the one hand, it can be calculated if the values of the parameter vector Θ are known. On the other hand, finding the maximum likelihood is greatly simplified if the values of the hidden variables are known. Consider the disadvantages of the algorithm. The main algorithm is unstable according to the initial data (ie those that initialize the parameter vector in the first iteration), we find a local extremum, the value of which may be much lower than the global maximum. Depending on the choice of the initial approximation, the algorithm may converge to different points, and the rate of convergence can also vary greatly. Hierarchical clustering

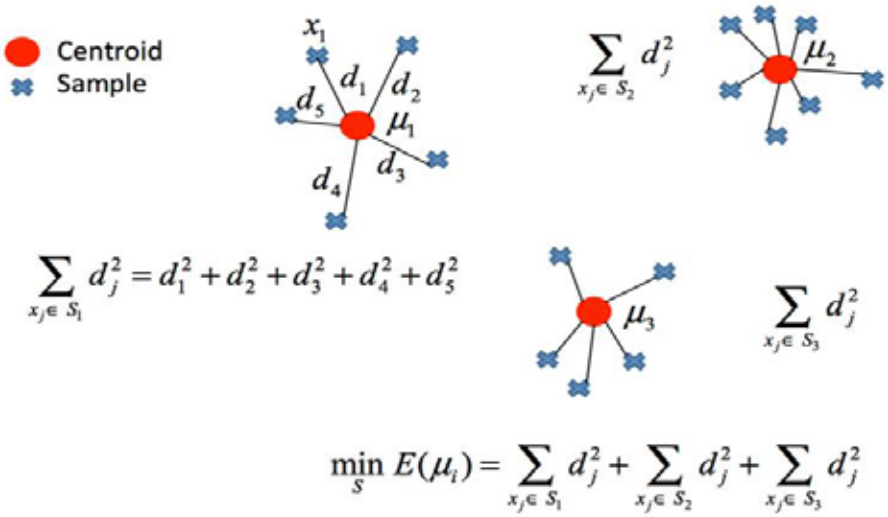

Fig. 1. Demonstration of the *k*-means algorithm [5]

is a set of data ordering algorithms aimed at creating a hierarchy (tree) of nested clusters. In hierarchical algorithms, the adjacency matrix of vertices $n \times n$ is used as an input, and the adjacency matrix contains the distance value, not a simple Boolean value. Hierarchical clustering algorithms assume that the set of objects being analyzed is characterized by a certain degree of connectivity.

According to the methods of adjusting the input weights of the adders and the solution of problems, there are many types of Kohonen networks. The most well-known of these are vector signal quantization networks, closely related to the simplest basic algorithm of cluster analysis (the method of dynamic nuclei or *k*-means), self-organized Kohonen maps and vector quantization networks.

The Kohonen layer consists of n parallel linear elements that have the same number of inputs m and receive at their inputs the same vector of input signals $x = (x_1, ...x_m)$. At the output of the j-th linear element we obtain the signal $y_j = w_{j0} + \sum_{i=1}^{m} w_{ji} x_i$, where w_{ij} is the weight of the i-th input j of the neuron, w_{i0} is the threshold coefficient. After passing the layer of linear elements, the signals are sent for processing, among the output signals y_j is looking for the maximum, its number $j_{max} = \arg\max\{y_i\}$. At the output, the signal with the number j_{max} is equal to one, the rest to zero. If the maximum is reached simultaneously for several j_{max}. Then either receive all the corresponding signals equal to one, or only the first in the list.

An overview of some existing commercial software products. Datawiz.io provides online services for data analytics in retail and restaurant business. The main task of the company is to automate the analysis processes: checks, sales and loyalty program data. Datawiz.io uses clustering as a method of grouping customers by data about their behavior (purchases, banking transactions, credit histories). The *k*-means algorithm is used to cluster an array of data (checks, data on loyalty programs). It is well scalable and optimized for the Hadoop platform. Also, the Affinity Propagation algorithm is used as an alternative. It has a number of significant disadvantages, it is slow and poorly scalable. But in some cases we can use it for clustering at short intervals. The Data-Centric Alliance (DCA) creates digital marketing technologies and develops products based on Big Data and Programmatic (fig. 2). It has one of the largest arrays of anonymous user data.

DCA helps marketers and analysts learn about people's behavior on the Internet and in real life. It has an R&D department to develop new tools that allow to learn more about Internet users and advertise more effectively. For its application, the company performs clustering of graphs created using the *k*-medoids algorithm developed by Python for clustering web domains. Using the DMP database (data on visits by users of different domains), a graph is constructed where domains act as nodes, and as edges a selective estimate of the extent of user visit events for domains [6].

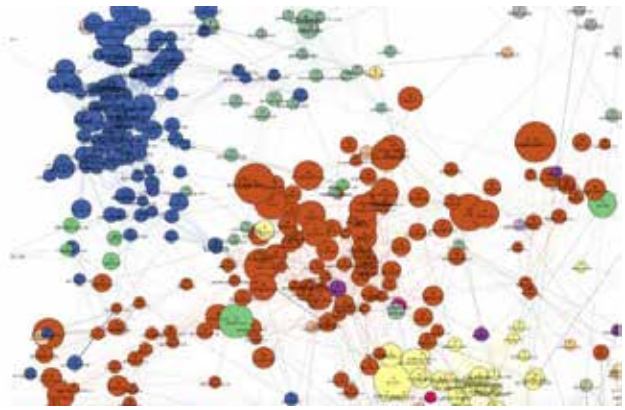

Fig. 2. Clustering of news sites [6]

Modification of the selected method of clustering user search queries. Consider the clustering method, when the data (web documents) to be clustered will be represented by graphs instead of vectors. Nodes represent elements to be clustered, and the weights at the edges connecting the two nodes indicate the distance (dissimilarity) between objects, and represent nodes. After applying the algorithm, the associated components indicate which objects the clusters belong to objects whose nodes are connected by edges are in the same cluster. For *k*-means, data elements are displayed as a set of *n*-numeric values (vectors in the R^n space). Then the Euclidean distance in this space and the centroid of the set of vectors are used to calculate the average value of the data in the cluster. As input we have a set of *n* data elements and the parameter *k*, which determines the number of clusters to create, the output data are the centroids of the clusters and the cluster (an integer within [1, k]) for each data element to which it belongs. First, we assign each data element to a random cluster (1 to k), using the initial assignment, to determine the centroids of each cluster. Given the new centroids, we assign each data element so that it is in the cluster of its nearest centroid. Then we recalculate the centroids. Extending the classical *k*-means clustering algorithm by using graphs will store information that is often discarded in simpler models. For example, by representing web documents with

graphs instead of vectors, we can store information such as the order in which the terms appear or where the terms appear in the document. This can improve the quality of clustering.

Assume that the graph g is the maximum general subgraph (*mcs*) of the graphs G_1 and G_2, ie: $g = mcs(G_1, G_2)$, if $g \subseteq G_1, g \subseteq G_2$ and there is no other subgraph $g' = (g' \subseteq G_1, g' \subseteq G_2)$, such that $|g'| > |g|$. The graph g is the minimum general supergraph (*MCS*) of the graphs G_1 and G_2, ie $g = MCS(G_1, G_2)$, if $G_1 \subseteq g, G_2 \subseteq g$ and there is no other supergraph $g' = (G_1 \subseteq g', G_2 \subseteq g')$, such that $|g'| < |g|$. A common subgraph is the part of both graphs that does not change when we delete or insert nodes and edges. To modify the graph G_1 in G_2, follow these steps:

1. Remove nodes and edges from G_1 that are not displayed in $mcs(G_1, G_2)$.
2. Make any substitutions for nodes or edges.
3. Add nodes and edges from G_2 that are not displayed in $mcs(G_1, G_2)$.

The size of the maximum general subgraph is related to the similarity of the two graphs. Determine the following measure of distances based on *mcs*:

$$d_{MCS}(G_1, G_2) = 1 - \frac{|mcs(G_1, G_2)|}{\max(|G_1|, |G_2|)}, \quad (1)$$

where $max(x, y)$ is the usual maximum of two numbers x and y and | … | indicates the size of the graph (usually considered the number of nodes in the graph). This measure of distance has four important properties:

– restrictions on the formation of numbers in the range [0,1];

– the distance is 0 only when the two graphs are the same;

– the distance between two graphs is symmetric;

– the distance obeys the inequality of the triangle, which provides the measurement of distance in an intuitive way.

The advantage of this approach over the graph editing distance method is that it does not require the determination of any cost factors or other parameters. The notion of the median of a set of graphs acts as a representative of the set. The median of a set of graphs S is a graph $g \in S$ such that g has the smallest average distance to all elements in S:

$$g = \arg\min_{\forall_S \in S} \left(\frac{1}{|S|} \sum_{i=1}^{|S|} d(s, G_i) \right). \quad (2)$$

Since $g \in S$, it is simple to calculate the average distance to all graphs for each graph in S. The median of the set of graphs always exists, and may or may not be the average value. Now that we have y as a measure of distance for graphs (1) and a method for determining the representative of a set of graphs (2), we can apply the method to data sets whose elements are graphs, not vectors. Thus, it is necessary replace the distance measurement used in step 3 with a graph distance measurement and replacing the centroid calculated in step 2 with the median of the set of graphs.

In the modified *k*-means algorithm using graph theory, the input data is a set of *n* data elements (represented by graphs) and the parameter *k*, which determines the number of clusters. The initial data are the centroids of the clusters (represented by the middle graphs) and the cluster (an integer in [1, k]), for each data element it belongs. First, we assign each data element to a random cluster (1 to k). Using the initial assignment, determine the median of the set of graphs of each cluster. Given the new medians, assign each data element in the cluster to its nearest median using a graph of distance. Recalculate the medians.

For the problem of automatic determination of the optimal number of clusters, it is necessary to know the cluster validation index, which is an indicator of the quality of clustering. The Dunn index is one such indicator, but it is sensitive to noise. We first calculate the index C, which is defined as:

$$C = \frac{S - S_{min}}{S_{max} - S_{min}}, \quad (3)$$

where S is the sum of all distances of pairs of elements in one cluster. Determine that l is the number of pairs used to calculate S. S_{min} and S_{max} are the sum of l of the smallest and largest distances, respectively. The smaller the value of C, the better the clustering. Another indicator is the Davies–Bouldin index, defined as:

$$DB = \frac{1}{M} \sum_{i=1}^{M} \max_{j=1,..,M;\, j \neq i} (d_{ij}), \quad (4)$$

where M is the number of clusters and

$$d_{ij} = \frac{r_i + r_j}{d(c_i, c_j)}. \quad (5)$$

In equation (5), r_i is the average distance of all data elements of the cluster i to their center, $d(c_i, c_j)$ is the distance between the centers of clusters i and j. The measure d_{ij}, similar to the Dunn index, in terms of compactness (numerator) and separation (denominator) of cluster pairs.

Experiments were performed with k values ranging from 2 to 6 for F-series and J-series datasets using both global *k*-means and random initialization. Random initialization is performed by randomly assigning each data element to the cluster. It is not possible

to reuse the same random initialization for different values of *k*, so each experiment has a separate random initialization. The best number of clusters is determined from the random index and mutual information, which indicate efficiency compared to the primary values. From the obtained results, these two indices are the same for *k* experiments that used the global *k*-means. There were no agreements for experiments using random initialization, and therefore it is not possible to definitively determine the best number of clusters in these cases.

The Dunn and *C* index does not seem very useful in terms of finding the correct value of *k*. Only in the case of the global *k*-means from the F series did the Dunn index coincide with other indices, and even then the Dunn index had the same values for all *k*. Index C performed slightly better, coinciding with other *k*-means indices for J-series; although the optimal value was at k = 2 for k-means from the series F.

Evaluating the effectiveness of the clustering algorithm. To assess the quality of the algorithm for clustering user search queries, it is necessary to investigate to determine the key parameters that are important for the operation of the algorithm. After a detailed study of the question, we can enter indicators of the number of vertices of the graph, the degree of convergence of nodes and the speed of the algorithm. These characteristics fully assess the improvements in the framework of continuous data transmission in decentralized networks, with a variable nature.

A sample of 5000 search queries was selected and the number of clusters k = 40 was recorded. The change of results between runs occurs from random initialization in the first step of the *k*-means algorithm. For these experiments, we used distance measurement and standard representation. Mutual information is a general degree of agreement provided by truth, with the advantage of clusters that have a high purity (ie are homogeneous with respect to the classes of merged objects). Higher values mean better performance of the clustering algorithm.

The use of Euclidean distance in the *k*-means algorithm gives worse results due to the invariance property of the vector length. For example, for the number of clusters 150, the degree of convergence of the clustering of search queries using a graph measure of distance gives a better result of 0.2218, while the convergence when using Euclidean distance is equal to 0.046. Because of this, documents with similar keyword frequency ratios but different in overall frequency have large distances between clusters, even if they are considered similar. Even with only 5 nodes, the modified method is superior to both *k*-means using Euclidean distance and a random baseline; as the number of nodes per graph increases, the performance approaches the efficiency of other *k*-means methods until it exceeds even the best *k*-means method, at 75 or more nodes. However, the increase in productivity is not always strictly proportional to the increase in the size of the graph. For example, an improvement of 60 to 75 is greater than an improvement of 75 to 90, even with the addition of 15 new nodes in each case. This may be because the additional nodes that are added when we increase the size of the graph, although they are common, may not always provide information that is useful for distinguishing between phrases, but may actually interfere with the input of third-party data.

Conclusions. This article analyzes the existing methods for the task of clustering user searches. Their main advantages and disadvantages are revealed. The obtained information was systematized for use in further work to determine the requirements for software development, which will create an application for clustering user searches. A modification of the *k*-means method is that in comparison with the basic method it is necessary to replace the measurement of Euclidean distance, graph measurement, as well as to replace the centroid with the median of the set of graphs. Further refinement may be to find the best node selection methods to be used in each graph, rather than relying heavily on the frequency of the term.

Table 1

Search query clustering metrics using a graph of distance

The number of vertices of the graph	The degree of convergence	The number of vertices of the graph	The degree of convergence
150	0.2218	60	0.1865
120	0.2142	45	0.1758
90	0.2074	30	0.1617
75	0.2045	15	0.1540

References:

1. How to distribute the semantic core of the website. URL: https://serpstat.com/blog/how-to-distribute-the-semantic-core-of-the-website/.
2. Ling Chen, Sourav S. Bhowmick, Wolfgang Nejdl. Web user clustering based on evolutionary web sessions. *Data & Knowledge Engineering*. Vol. 68. Issue 10. October 2009. P. 867–885.
3. Miao Wan, Arne Jönsson, Cong Wang, Lixiang Li & Yixian Yang. Web user clustering and Web prefetching using Random Indexing with weight functions. *Knowledge and Information Systems*. Vol. 33 2012. P. 89–115.
4. Poornalatha G., Prakash S. Raghavendra. Web User Session Clustering Using Modified K-Means Algorithm. International Conference on Advances in Computing and Communications ACC 2011. P. 243–252.
5. K-means algorithm applied to image classification and processing. URL: https://www.unioviedo.es/compnum/labs/new/kmeans.html.
6. Rybachuk K. Community detection. URL: https://www.slideshare.net/kirillfishkurtosis/community-detection-51552715.
7. Ferrer M., Valveny E., Serratosa F., Bardají I., Bunke H. Graph-Based k-Means Clustering: A Comparison of the Set Median versus the Generalized Median Graph. *CAIP 2009: Computer Analysis of Images and Patterns*. P. 342–350.

Мовчан К.О., Олещенко Л.М. КЛАСТЕРИЗАЦІЯ ПОШУКОВИХ ЗАПИТІВ КОРИСТУВАЧІВ МЕРЕЖІ ІНТЕРНЕТ ІЗ ВИКОРИСТАННЯМ ТЕОРІЇ ГРАФІВ

Одним із принципів Інтернет-маркетингу є орієнтація на клієнта щодо його потреб, що характеризуються пошуковими запитами в мережі Інтернет. Побудова семантичного ядра сайту вирішує задачу розподілу пошукових запитів користувачів на сторінках веб-ресурсу. Семантичне ядро сайту допомагає визначити, яка сторінка найточніше відповідає конкретному пошуковому запиту або групі запитів користувачів. Аналізуючи характеристики кластерів користувачів, можна краще зрозуміти потенційних клієнтів і видавати користувачам більш релевантні веб-документи. Кластеризація використовується для ідентифікації та класифікації підгруп користувачів мережі Інтернет на основі їхньої поведінки. Кожен метод кластеризації використовує різні критерії для групування об'єктів даних. У статті розглянуто особливості використання k-means, EM-алгоритму та нейронних мереж Кохонена для кластеризації пошукових запитів. Більш детально розглянуто дослідження методу кластеризації пошукових запитів користувачів із використанням теорії графів. Вузли представляють елементи, що підлягають кластеризації, а ваги по краях, які з'єднують два вузли, вказують на відстань (несхожість) між об'єктами. Після застосування алгоритму пов'язані компоненти вказують, до яких об'єктів кластери належать об'єктам, вузли яких з'єднані ребрами, що знаходяться в одному кластері. Показано, що запропонований підхід є більш ефективним порівняно з базовим алгоритмом кластеризації. Для нашого дослідження використано вибірку із 5 000 пошукових запитів і кількість кластерів k = 40. Зміна результатів між прогонами відбувається від випадкової ініціалізації на першому кроці алгоритму k-means. Дослідження показують кращі результати збіжності завдяки представленню даних (веб-документів) для кластеризації у вигляді графів замість векторів. Недоліком цього рішення є використання більшого обсягу пам'яті, ніж у базовому методі, також не є вирішеною проблема вибору початкової кількості кластерів.

Ключові слова: *алгоритми кластеризації, теорія графів, k-means, евклідова відстань, вершини графа, графова міра відстані, збіжність.*