Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Computer Science

Master`s Thesis

CONSTRUCTION AND ANALYSIS OF THE GRAPH OF SOCIAL NETWORK PROFESSIONAL COMMUNITIES

Adel Shavalieva

Supervisor: **Ing. Karel Frajták, Ph.D.**

Study Program: Open Informatics
Field of Study: Software Engineering
May 24, 2019

MASTER'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Shavalieva Adel** Personal ID number: **484745**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Computer Science and Engineering**

Study program: **Open Informatics**

Specialisation: **Software Engineering**

II. Master's thesis details

Master's thesis title in English:

Construction and analysis of the graph of social network professional communities

Master's thesis title in Czech:

Construction and analysis of the graph of social network professional communities

Guidelines:

Design and analysis of the graph of social network "VKontakte".
This includes collecting the necessary data according to KFU students profiles and creating a weighted of the graph.
Formulation and testing of the hypothesis by analyzing the resulting graph.

Bibliography / sources:

[1] Newman M. Modularity and community structure in networks, PNAS Vol. 103, N 23, pp 8577-8582, 2006
[2] Гусарова Н.Ф. Интеллектуальные системы в управлении социальными процессами. – СПб: Университет ИТМО, 2015. 90 с.
[3] Matthew D. "Social Network Analysis", University of Massachusetts Amherst, 2014
[4] Newman M. "The physics of networks." Physics Today, 2008
[5] Porter, Mason A., Jukka-Pekka Onnela, and Peter J. Mucha. "Communities in Networks". Notices of the AMS, 56(9), 2009

Name and workplace of master's thesis supervisor:

Ing. Karel Frajták, Ph.D., Software Testing Intelligent Lab, FEE

Name and workplace of second master's thesis supervisor or consultant:

Date of master's thesis assignment: **05.04.2019** Deadline for master's thesis submission: **24.05.2019**

Assignment valid until: **19.02.2021**

Ing. Karel Frajták, Ph.D. Supervisor's signature	Head of department's signature	prof. Ing. Pavel Ripka, CSc. Dean's signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce her thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

.	
Date of assignment receipt	Student's signature

Acknowledgements

I would like to express my sincere gratitude to my supervisors, Ing. Karel Frajták (CVUT) , Ing. Zulfira Enikeeva (KFU) and Galim Vakhitov (KFU) for guidance and immense to help on my thesis. I want to especially thank my family, friends and colleagues who prop up me morale throughout my thesis work. Also I want to appreciate Turilova Ekaterina, Enikeev Arslan, Miroslav Bureš and all Czech team for the opportunity to participate in the Double Degree program (CVUT, KFU)

Declaration

I hereby declare that I have completed this thesis independently and that I have listed all the literature and publications used.

I have no objection to usage of this work in compliance with the act §60 Zákon č. 121/2000Sb. (copyright law), and with the rights connected with the copyright act including the changes in the act.

Prague, May, 2019 _____

Abstract

SHAVALIEVA, Adel: Construction and analysis of the graph of social network professional communities. [Master's Thesis] - Czech Technical University in Prague. Faculty of Electrical Engineering, Department of Computer Science. Supervisor: Ing. Karel Frajták, Ph.D.

Social Network Analysis is a study of social networks that considers social relationships in terms of network theory. It is widely used in a number of applications and disciplines, e.g. selection of user segments for targeted advertising campaigns or collaborative filtering. Network analysis includes data collection and accumulation, network and sample modeling, analysis of user characteristics and behavior.

The object of the thesis is to develop a model for analyzing the social network graph. The problem of the lack of a ready-made method for identifying the interests of users of a social network based on the communities they are in is considered. To solve this problem, a social network model was constructed in the form of a weighted graph, and data clustering based on their specific features was implemented on the basis of social network materials.

Keywords: social network, community detection, clustering, graphs, graph visualization.

Contents

List of figures

List of tables

Introduction

Social networks as groups of people connected and communicating with each other have always existed. In sociology, this phenomenon stood out and began to be studied long before the advent of the Internet. However, it was the specialized Internet resources that made it possible overnight to collect a huge database of social interactions valuable for study.

Social data analysis is rapidly gaining popularity around the world thanks to the emergence of online social networking services in the 1990s (Facebook, Instagram, Twitter and others). The phenomenon of personal data socialization is connected with this: facts of biography, photo, video, audio, information about interests and belonging to communities have become publicly available. Thus, social networks are a unique source of data about the personal life and interests of real people. This opens up unprecedented opportunities for solving research problems.

A social network, like any network, can be mathematically modeled by a graph in which vertices represent network objects, and edges are interconnections. In contrast to the classical methods of analysis, which investigate the individual properties of objects, the main goal of the analysis of social networks is the study of interactions between social objects. For this, a number of quantitative and qualitative concepts are used, such as the degree of clusterization, connectivity, and others.

1.1 Thesis content

The analysis of social networks is an actively developing area of Western sociology. The interest of researchers in this area is due to the fact that it provides a new set of explanatory models and analytical tools that are outside the framework of conventional quantitative methods.

The goal of the master's thesis is to implement data clustering on the basis of social network materials taking into account the specificity of data.

Before starting work, a number of tasks were set.

1. To discover the theoretical aspects of social network analysis.
2. To form a social network model in the form of a weighted graph.
3. To cluster data using various graph transformation methods.
4. To make a comparative analysis of clustering results.

This thesis consists of the introduction, theoretical, algorithmic, practical chapters and conclusions. At the end of the work are the applications in which you can find additional images and the source code for building, analyzing and visualizing graphs.

Chapter 1

Theoretical aspects of social network analysis

1.1 General information about Theory of Social Network Analysis

The specifics of the analysis of social networks

Social networks are an excellent tool for research because users themselves publish information about themselves, their views, interests, preferences and much more. The analysis of social networks is a study of social networks that consider social relations in terms of the theory of networks. These terms include the concept of a node (displaying an individual participant within a network) and communication (mapping relationships between network objects, such as friendship, kinship, common interests, etc.). These networks are often described in the form of social network schemes, where the dots denote nodes, and the lines represent connections.

Studying social networks covers three main topics:

- analysis of network structure,
- network formation,
- processes on networks (for example, the spread of rumors, trends, etc.). [1]

The study of social networks as a scientific direction originated at the junction of a number of scientific disciplines - sociology, discrete mathematics, Computer Science (algorithms on graphs and networks). Recently, statistical physics has been added to this list, where objects such as social networks have also been identified, as well as economics, from which approaches of independent agents, game theory, etc. have come.

The specificity of social networks as complex network entities is that, on the one hand, they are not regular, but at the same time they cannot be regarded as purely random - this is their complexity.

Absolutely random networks are studied by statistical physics, absolutely regular (regular lattices) are studied by mathematics. Social networks have an intermediate position. Therefore, the construction of the image of large networks, provided that the necessary characteristics of the network are clearly shown on it, is a nontrivial task. [2]

However, when working with social data, factors such as instability in the quality of user content (spam and false accounts), problems with ensuring the privacy of users' personal data during storage and processing, as well as frequent updates of the user model and functionality should be taken into account. All this requires continuous improvement of algorithms for solving various analytical and business problems.

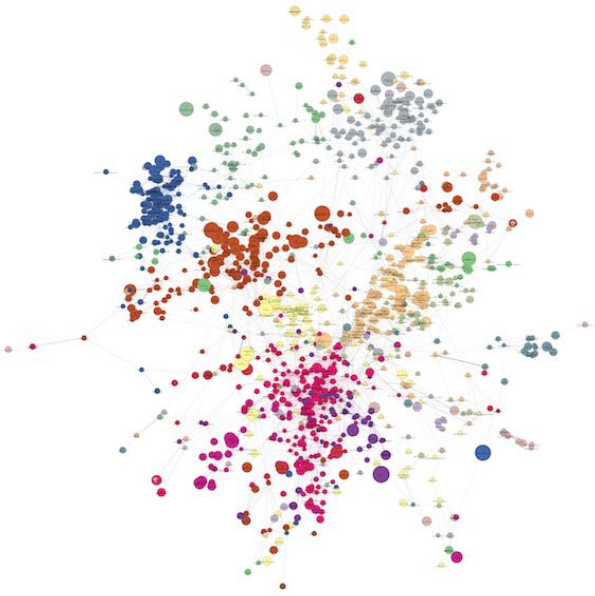

Fig. 1.1: Social network graph

Directions social networking research

To date, there are 4 approaches to analyzing social networks: resource, dynamic, regulatory and structural.

The resource approach analyzes the participants' ability to use individual and network resources to achieve their goals. Under the individual resources are understood: knowledge, gender, age, money. Network resources can be information, influence, status.

The structural approach considers the geometric shape of the network (graph). The network objects studied are represented as graph vertices, and the edges are the interactions between them. The weight of the edge is the intensity of such interactions. The analysis of the graph takes into account the weight of the edges, the location of the vertices, centrality, the diameter of the graph and much more.

The regulatory approach is aimed at studying the behavior of network objects and the processes of their interaction. The social roles of objects and their relationships with each other are analyzed. For example, relationships can be friendly, kindred, or working.

The dynamic approach is one of the latest trends in the analysis of the social network. This approach tracks changes of network structure over time. Special attention is paid to the causes of the appearance and disappearance of edges. It also considers the issues of changing the network structure under the influence of external factors, the existence of stable social network configurations and others.

Data collection

For the collection of data was taken social network “Vkontakte”.
“Vkontakte” is one of the most popular social networks in Russia, in which more than 410 million accounts are registered.

Fig. 1.2: Statistics of the social network VKontakte

In this social network, users actively join communities, share trends and their opinions, track news and much more. This is one of the main reasons for choosing it as a subject for research, the second is its popularity among Russian youth, and the third is a convenient and accessible API for data collection.

VKontakte API is an interface that allows you to receive information from the vk.com database using http requests to a special server. With it, you can easily go to the pages of users and get the necessary information for analysis.

To use the VK API, you need to register on the social network and register your application. There are three types of applications:

- Standalone application is an API_ID for a mobile or desktop client, an external site where the work with the API will be carried out on Javascript. The basic idea is that requests to the API should be made from the user's device. In the application interface with this type, the SDK settings and the connection of certificates for push notifications are available.
- Website - register API_ID for an external site and work with API from the server.
- IFrame / Flash application - applications that uploaded directly to the VKontakte (Flash) server or embedded in a frame from an external site.

Network“VKontakte”, like every social network, has privacy settings and blacklists. For some users, access is denied, and some are able to see partial or complete information. VK API uses the same principle. If a user has allowed only for friends seeing his community list, then and through the API only his friends can see communities. Therefore, when collecting data, you must consider the privacy settings.

1.2 Main issues and approaches of social network analysis

While analyzing social networks one has to face many different problems. Some of them require a large amount of time to be solved, others - a large amount of resources. Consider the main problems and approaches to their solution.

Data collecting issue.

Data is not stored in the public domain for free download. Each social network has a different degree of difficulty in obtaining data from it. There are two main ways to do it:

- obtaining data via API (open or with the need to register your service)
- crawling of html-pages with their subsequent parsing and isolating the necessary data

The problem of limiting fast data collection.

In case of the API:

The social network identifies the application using it (hereinafter referred to as the Client) by the unique key issued to it, which the latter subsequently is specified in requests to the API. The client in most cases is limited to a certain number of requests per unit of time. If the limit is exceeded, the requests fail and the Client does not receive the requested data.

In case of crawling:

Social networks basically, like any large service, have a limit on the number of requests to the server from users per unit of time.

This problem is especially acute in case of:

- the need to obtain a sufficiently large amount of data
- a high speed of changing of data relevancy

Validity

When working with social network data, only the public part of them is often available. Exploring a social network, you can build many options for links between network objects, but not always in studies there is a theoretical justification for the choice of a particular structure and its connection to data. [3] The result is a large discrepancy between the selected model and the data explored.

When analyzing data, it is important to understand the technical and social aspects. For example, the lack of connections between objects does not mean that they do not interact with each other in real life. Although in most cases, such an analysis approach is justified.

Intensity of relations

Another problem when analyzing networks is the intensity of interactions between objects. This task still has no explicit solution. For example, a person on Facebook has links to his friends, but he doesn’t communicate with some of them at all. To determine the strength of interaction between friends, you can take as the criterion of connectivity the number of sent messages or the amount of information exchanged. The intensity of interactions can be taken into account using weighted networks.

Such a network can be built in several ways. In this example, you can apply unit weighting, which increases the edge weight for each message between objects. Another way is to apply the fading effect, that is, to give more weight to the messages sent last than the older ones. Unfortunately, most SNA

methods are not designed for weighted networks. Since they were intended for networks that have approximately equal power of interaction between objects. Applying methods not intended for weighted networks can lead to incorrect results.

A careful analysis of the determination of the threshold of the boundary values is also necessary. For example, take into account only objects that have many connections with other objects, or take into account interaction intensities that are larger than the threshold value. Such an approach may not always be correct, since an accurate analysis of the choice of boundary values is required. Some networks may be sensitive to discarding small data, because they can make up the majority of sample data due to the concept of the power of weak ties of M.Granovetter.

Data stability

In the process of researching the social network, the question arises about the stability of the selected data. Data can be stable, for example, when we consider the connection between close friends or relatives. They can also be dynamic, so the structure of the network will constantly change over time. It should be noted that in most cases the stability of the data does not have a strong influence on the accuracy of the data when analyzing a social network. If the data in the social network is stable, then you can aggregate the data as snapshots or make a longitudinal study. For unstable data, you can also take a snapshot, but after a while this data will not give an accurate picture of the network.

The problem of processing a large amount of data.

From the very beginning of working with social network data one has to face the problem of a large amount of data. First, data processing is not possible on a computer with an insufficient amount of RAM due to the fact that the data allocated memory is directly proportional to their volume. Secondly, the visualization takes a large amount of time, and with too much data, this time will approach the time spent on the entire study. To solve the first you need to use a machine with a large amount of RAM. As for the issue of visualization, there are several solutions. For example, you can cluster the graph and the number of the cluster to which this node belongs, take as a node. Another solution is to sample the graph according to certain criteria.

Ethical issues

Data taken for research may be anonymous or confidential. For this reason, an independent ethics committee (IEC) may reject a research proposal. Network data is taken from a public source and there is no need in user consent. Although the data is public, it is better to resort to privacy and hide the user's personal data. For example, anonymize the nodes. However, this may not be enough, since if there are nodes with a high degree of connectivity, you can guess which object is behind them. Therefore, you should try to take into account the possible consequences of the analysis, because there

is a chance that the result of the study will be useful not only to the researcher himself, but also to the attackers.

There are many problems that need to be solved in the analysis. Validating the data, choosing the correct normalization of data and research method, it is possible to get good results in the analysis of social networks.

1.3 Existing social network analysis software products overview

Social network analysis software simplifies analysis by describing network features through a visual or numerical presentation. This article has reviewed the most popular software products on the market that allow you to analyze the graph of a social network and build visualization.

GraphX - is an Apache Spark library designed to analyze graphs. GraphX supports the Scala programming language. In GraphX graph is represented as a multigraph, at the vertices and edges of which there is additional information. Apart from the vertices and edges, the Triplet object is implemented there, in which, in addition to information about the edge, there is information about the vertices adjacent to it.

GraphX implements the PageRank reference ranking algorithm, that is, the algorithm ranks by importance all the vertices of the graph among all others. GraphX provides a static and dynamic version of the implementation of PageRank. In the static version, there is a fixed number of iterations, and in the dynamic version, the algorithm will work until the rating is closer to the specified value. In addition to the reference ranking, GraphX is able to search for connectivity components, count the number of triangles, and search for strong connectivity components, but there are no ready-made clustering algorithms (version 1.4.1).

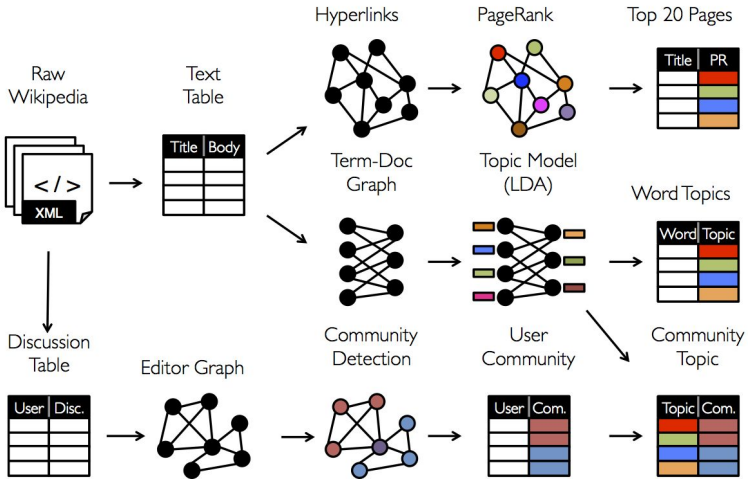

Fig. 1.3: Pipeline in GraphX

Okapi - is a library of machine learning and graph mining algorithms with algorithms built on top of Apache Giraph. Apache Giraph is an iterative graph processing system designed for high scalability. At the moment is used in large companies such as Facebook. Giraph is an add-on to Hadoop, designed to handle graphs. Giraph appeared as an alternative to Pregel with open source.
The okapi library includes modern collaborative filtering algorithms used in recommender systems, as well as graph algorithms such as sectioning and clustering. Okapi also uses its own development algorithm - Spinner, based on the Label Propagation algorithm.

Graph-tool - Python library aimed at manipulating and statistical analysis of graphs. The basic algorithms and data structures are written in C ++, which gives a high level of performance (both in terms of computation speed and memory usage). Also in this library is widely used metaprogramming, based on the Boost Graph Library. Many algorithms are implemented in parallel using OpenMP and for local computing (graphs up to a couple of hundreds of thousands of nodes) graph-tool is the fastest option. The library provides graph statistics (average shortest distance, a histogram of degrees, etc.); includes algorithms for finding the minimum spanning tree, connectivity components; generates a random graph.

Gephi - is a graph analysis software written in Java. It provides an easy way to visualize data, with the management of the structure, color and shape of the graph. In addition, there is a 3D rendering mechanism capable of displaying large networks in real time. Gephi implements algorithms for finding the density of a graph, connected components, clustering coefficient, PageRank, and so on.

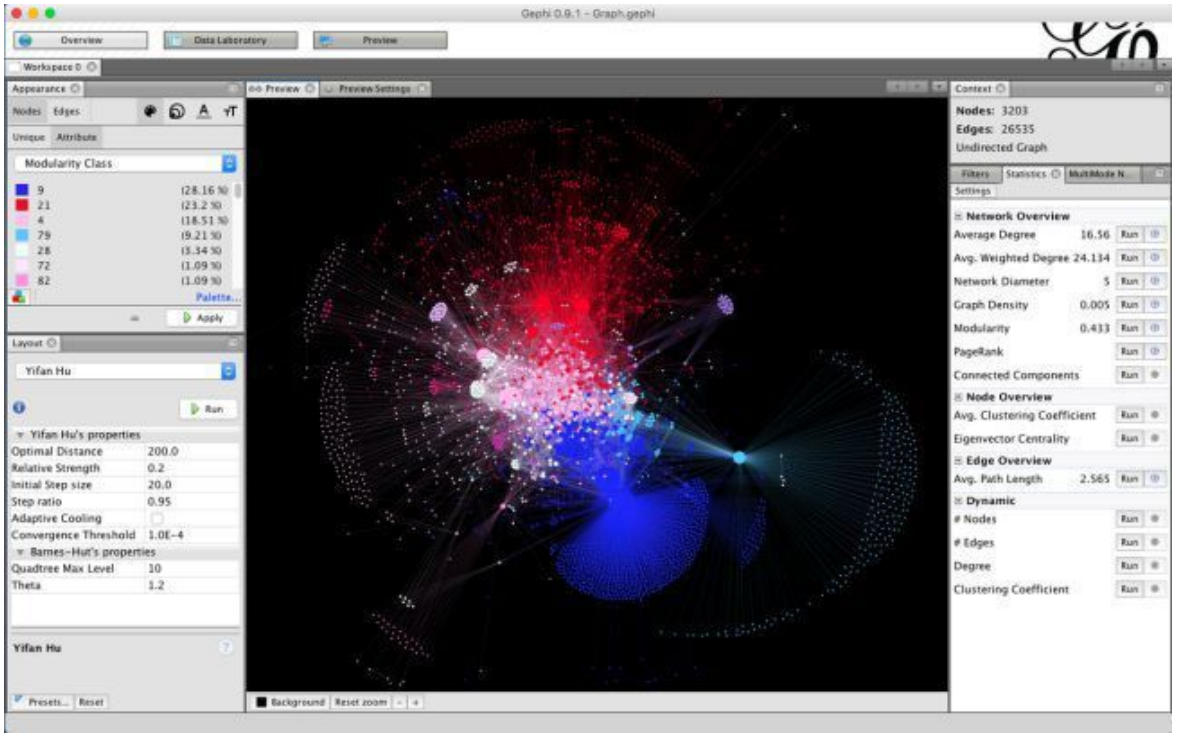

Fig. 1.4: Example of building a graph using Gephi

Networkx - library is written in Python. By speed and visualization it inferior graph-tool and gephi. NetworkX has many algorithms for graph analysis. The library allows us to build random graphs, find k-cores and clicks. It is also able to obtain such graph characteristics as the degree of vertices, the height of the graph, diameter, radius, center, etc. Despite the abundance of methods, networkX is not able to cluster the graph.

Chapter 2

Formation of a social network model as a weighted graph.

2.1 The structure of the graph. Estimation of weights of the graph's edges.

The goal of the thesis is to investigate the communities in which students of Kazan Federal University, Institute of Computational Mathematics and Programming are members. As the initial data, URLs of pages of personal accounts of students in the social network "VKontakte" were taken. It was necessary to collect all identifiers (IDs) of the communities in which they were members, and to analyze the resulting graph.

It was advisable to take the communities themselves as nodes of the graph. The problem was in defining the connections between these communities. The task was to determine the way communities interact in a social network. Each group had its own administrators and own topics. Perhaps some of them had overlapping materials or ideas [4] - [5]. The following approach was chosen to determine the connections between the communities. If members of community B were among members of community A, then a connection arises between these communities. To take into account the fact that one pair of groups has 50% of the total participants from the entire sample, while the other has less than 1%, it is necessary to introduce a certain criterion. This criterion is the similarity measure, expressed as the weight of the edge of the graph.

The similarity measure can be divided into three groups [6]:

- unary - one object is considered;
- binary - two objects are compared;
- N-ary - N objects are compared.

Since the weight of the edge is a mathematical formalization of the relationship between two objects, we consider the binary similarity measures.

As a similarity measure of the two groups, we will use the generalized formula proposed by B.I.Syomkin on the basis of the average Kolmogorov formula.

$$K_{\tau,\eta}(A,B) = \left(\frac{K_\tau^\eta(A;B) + K_\tau^\eta(B;A)}{2}\right)^{\frac{1}{\eta}} \qquad (2.1)$$

1) If we take τ, η equal to 1 and -1, respectively, then we get the most popular binary similarity measure - the **Jaccard index**.

The Jaccard index can be written as follows:

$$K_{1,-1}(A,B) = \frac{n(A \cap B)}{n(A) + n(B) - n(A \cap B)} = \frac{n(A \cap B)}{n(A \cup B)} \qquad (2.2)$$

where n(A) - number of users in group A

n(B) - number of users in group B

2) If we take τ, η equal to 0 and -1, respectively, then we get binary similarity measure - the **Braun-Blanquet coefficient**.

The Braun-Blanquet coefficient can be written as follows:

$$K_{0,-1}(A,B) = \frac{n(A \cap B)}{max[n(A), n(B)]} = min\left[\frac{n(A \cap B)}{n(A)}, \frac{n(A \cap B)}{n(B)}\right] = \frac{2*n(A \cap B)}{n(A) + n(B) + |n(A) - n(B)|} \qquad (2.3)$$

where n(A) - number of users in group A

n(B) - number of users in group B

3) If we take τ, η equal to 0 and 1, respectively, then we get binary similarity measure - the **Overlap coefficient**.

The Overlap coefficient can be written as follows:

$$K_{0,+1}(A,B) = \frac{n(A \cap B)}{min[n(A), n(B)]} = max\left[\frac{n(A \cap B)}{n(A)}, \frac{n(A \cap B)}{n(B)}\right] = \frac{2*n(A \cap B)}{n(A) + n(B) - |n(A) - n(B)|} \qquad (2.4)$$

where n(A) - number of users in group A

n(B) - number of users in group B

As an alternative to binary similarity measures it is possible to use the **affinity index**. This method of specifying the weights of the edges of the graph makes it possible to distinguish in the data a fairly clear structure of clusters of groups. One of its interesting features is that each of the largest groups (Kazan news communities, KFU student communities, programming communities) mostly do not have a large edge weight neither with one nor with the other node. This leads to the fact that those groups as nodes often remain isolated, not falling into any of the clusters.

$$affinity(A,B) = \frac{\left|n(A) \cap n(B)\right| * |U|}{|n(A)| * |n(B)|} \qquad (2.5)$$

where n(A) - number of users in group A

n(B) - number of users in group B

|U| - total number of users considered

2.2 Construction of the empirical law of distribution of the values of the weights of the edges.

The empirical distribution function in mathematical statistics is an approximation of the distribution function constructed using a sample of it.

Empirical distribution function is defined as:

$$\widehat{F}(x) = \frac{1}{n} \sum_{i=1}^{n} 1_{\{X_i \le x\}} \qquad (2.6)$$

where $X_1, X_2 \ldots . X_n$ - sample, where the edge weight's value acts as a random variable, $1_{\{X_i \le x\}}$ - indicator of event $\{X_i \le x\}$. The empirical distribution function at x is equal to the relative frequency of the elements of the sample not exceeding the value of x.

As the values of x were taken a set of n numbers. Then, were found this set by the following formula.

$$x_0 = min\,(W) - \varepsilon$$

$$t = \frac{min\,(W) - \varepsilon - (max(W) + \varepsilon))}{n-1} \qquad (2.7)$$

$$x_{i+1} = x_i + t\,,\ i = 0,\ \ldots .\,,\ n-1$$

The first element will be the minimum value among the weights of the edges, calculated using the appropriate coefficient. In order for the function chart to start at y = 0, it is necessary to take the initial value of x slightly less than the minimum value of the weights of the edges. Then we find the remaining values of the set x with step t.

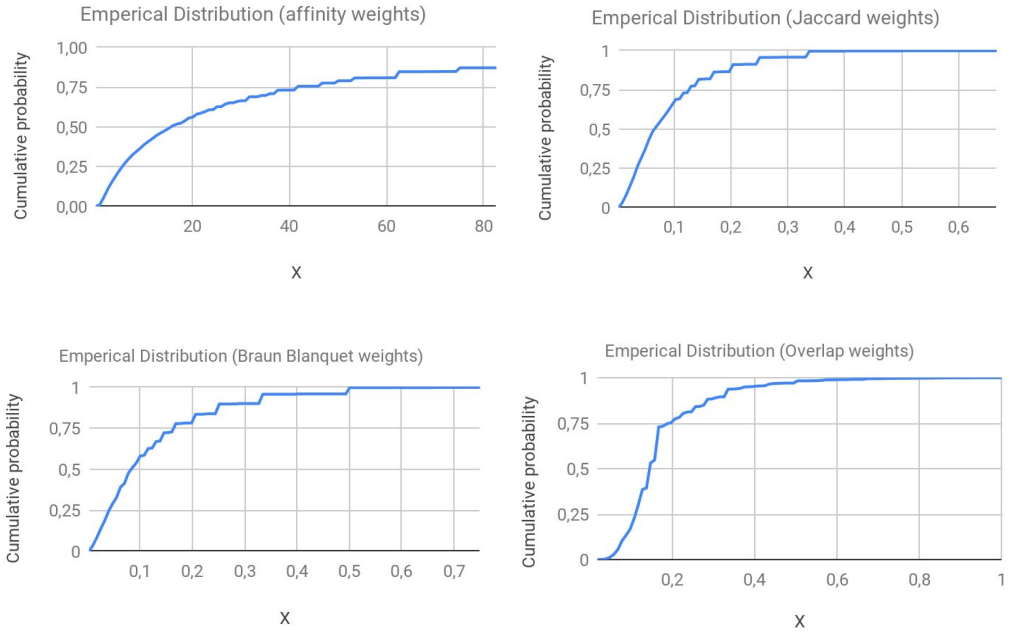

Fig. 2.1: Empirical probability distribution of edge weights

The obtained distribution laws will be used to analyze the structure of the graph. The empirical distribution function is sufficient statistics that retains the entirety of the sample information.

2.3 Estimation of the parameters of the theoretical law of the distribution of the weights of the edges.

Based on our sample, we describe the resulting variation curve using a mathematical function. The choice of mathematical dependence is carried out by selecting the appropriate mathematical model that determines the type of distribution function.

Consider the distribution function as such a model:

$$F(x) \approx b_0 + b_1 * ln(x) + b_2 * \sqrt{ln(x+1)} \qquad (2.8)$$

Using the method of least squares (OLS) [7], we find the best values of the parameters b_0, b_1, b_2 and values of F(x) maximum approximate to the actual values of y. So it is necessary to find such b for which the sum of squared deviations of e_i will be minimal.

$$\hat{b} = arg \min_{b} RSS(b) \qquad (2.9)$$

where RSS - Residual Sum of Squares.

$$RSS(b) = e^T e = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - f(x_i, b))^2 \qquad (2.10)$$

In the regression analysis probabilistic models of dependencies between variables are used

$$y_i = f(x_i, b) + \varepsilon_i = \sum_{j=}^{k} b_j x_{ij} + \varepsilon = X^T b + \varepsilon_i \qquad (2.11)$$

where y is a column-vector of the values of the empirical distribution function. X is the (n x k)-matrix of observations of the factors, n is the sample size.

For our model are used the general formula of OLS estimates in the case of linear regression:

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix} = \hat{b} = (X^T * X)^{-1} * X^T * y \qquad (2.12)$$

After finding the coefficients, we substitute them into the formula (*). Then using the coefficient of determination are determined the quality of the selection of the regression model.

The coefficient of determination is determined by the formula:

$$R^2 = 1 - \frac{\sigma^2}{\sigma_y^2} = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \overline{y})^2} \qquad (2.13)$$

where $\frac{\sigma^2}{\sigma_y^2}$ - is the proportion of variance, conditional on the factors x, of the dependent variable in the variance of the dependent variable,

$ESS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$ - error sum of squares,

$TSS = \sum_{i=1}^{n} (y_i - \overline{y})^2$ - total sum of squares, where $\overline{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$

The coefficient of determination (R^2) for a model with a constant takes values from 0 to 1. The closer the value of the coefficient to 1, the stronger the dependence, which means that the model is more consistent with the data. If R^2 > 0.8, then the model can be considered good enough. When R^2 = 1, the model provides a functional relationship between the variables.

As a result of our research, the following results were obtained:

Affinity weights	Jaccard weights	Braun Blanquet weights	Overlap weights
$R^2 = 0.93719$	$R^2 = 0.92699$	$R^2 = 0.94109$	$R^2 = 0.90364$

Table 2.1: Results of coefficients of determination

The values of the weights of the edges calculated using the Braun Blanquet coefficient are best suited for this theoretical function. For a more detailed analysis were found the probability density of the weights of the edges by the formula:

$$f(x) = F'(x)$$

$$f(x) = \frac{b_1}{x} + \frac{b_2}{2 * (x+1) * \sqrt{ln(x+1)}} \quad (2.14)$$

Display the results on the chart:

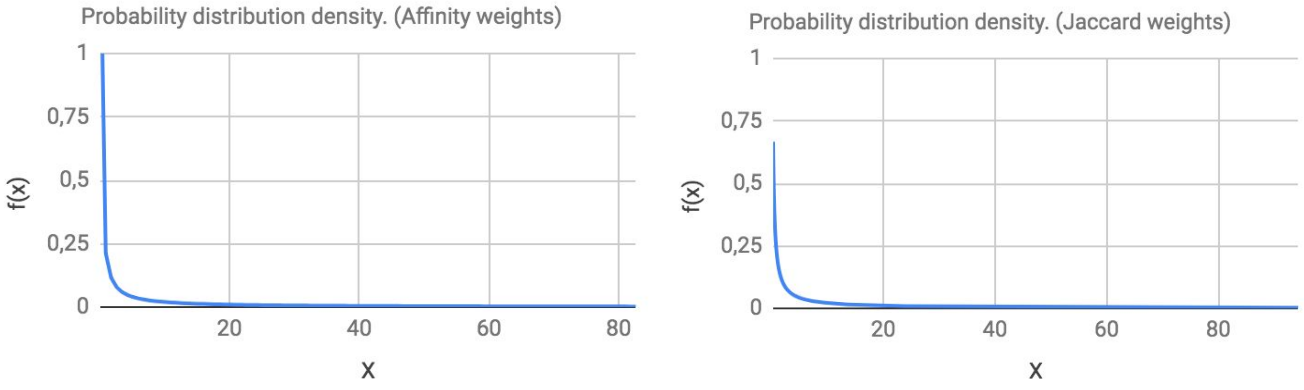

Fig. 2.2a: Affinity and Jaccard edge weights density distribution

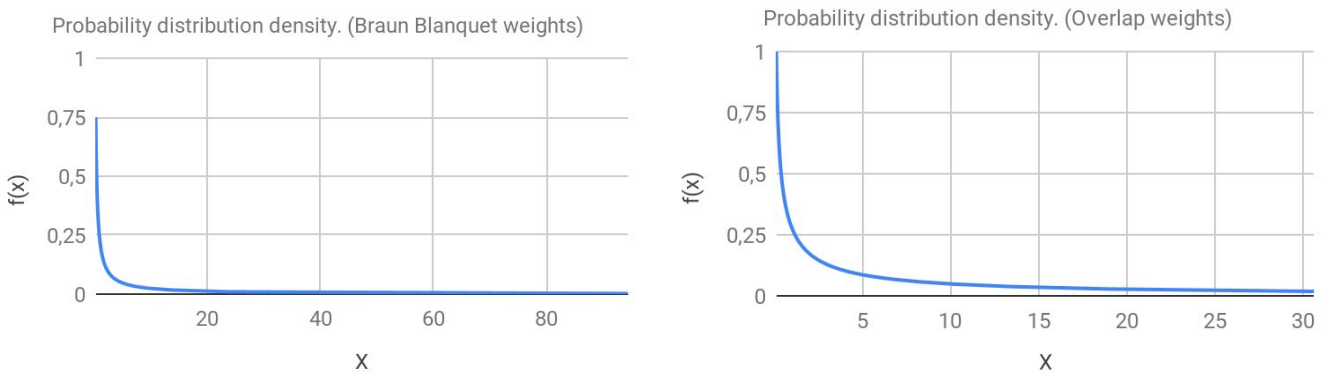

Fig. 2.2b: Braun Blanquet and Overlap edge weights density distribution

The studied values of the weights allowed us to approximate the empirical distribution function by the theoretical law of the probability distribution of the weights, which will facilitate the analysis of the structure of the graph.

Chapter 3

Clustering the data under study.

3.1 Description of the clustering algorithm and determination of its complexity.

Description of the clustering algorithm

Finding clusters in a graph is an important tool for studying and analyzing social networks.

The task of clustering is the task of splitting a set of objects into groups called clusters. Inside each group, there should be “similar” objects, that is, objects that have a certain similar characteristic. On the contrary, the objects of different groups should be as distinct as possible. The number of clusters is usually not known in advance, so it is necessary to independently determine its optimal value.

There are many different clustering methods. For example, there are hierarchical methods (ascending and descending), (based on the definition of cluster centers) centroid-based, (based on data distributions) distribution-based, (density methods) density-based, and others. A specific method is chosen separately for each tasks.

When analyzing the links between Vkontakte network groups, it is assumed that clustering will help to group communities by common topics. The number of topics is not known in advance, but it can be assumed that each group has only one of its main topics.

In this thesis, the k-medoids method was chosen as the clustering algorithm.

This algorithm resembles the k-means algorithm, but, unlike it, not the arbitrary point of space, but only an element from the set of nodes of the graph can be chosen as the central points of our clusters. We add such a restriction due to the fact that with the standard application of the k-means algorithm, the set of clustered elements lies in a certain continuous space (the elements are given by the coordinates of this space), and then, by setting the distance measure, you can determine the distance to any arbitrary point. Therefore, any element of this space will be suitable as a cluster center. In our case, the graph itself defines the space: the elements are the nodes themselves, and the distance between them is the weight of the corresponding edge. Therefore, only nodes of the graph are considered as the center.

This condition increases the algorithmic complexity of the algorithm itself. In the case of classic k-means, the averaging of cluster elements defined by a vector is taken to calculate the cluster center.

$$\forall i = 1, \ldots, k : \mu_i^{(t)} = \frac{1}{|S_i|} \sum_{x \in S_i} x \quad (3.1)$$

where k - amount of clusters,

$|S_i|$ - elements of the cluster)

The complexity of such a step: $O(|S_i|)$ - for each cluster or $O(N)$ - for the entire graph. (N - number of the elements). [8]

In k-medoids, it is necessary to “try” each node of the cluster as a medoid, and measure the average distance from it to each other node of the cluster.

$$\frac{1}{|S_i|} \sum_{x \in S_i} \sum_{y \in S_i} d(x, y) \quad (3.2)$$

The difficulty became: $|S_i| * O(|S_i|)$ - for each cluster, or in the worst case: $O((n-k)^2)$

The k-medoids algorithm divides N given objects into K classes, where the number K is specified at the beginning of the algorithm as a hyperparameter. A matrix of distances between objects (weights in the graph) is provided to the algorithm as the input. One of the most common methods for implementing the k-medoids algorithm is Partitioning Around Medoids (PAM). The goal of PAM is to minimize the distance between objects belonging to the same class [9]. This is expressed by the formula below:

$$F(x) = minimize \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \, b_{ij} \quad (3.3)$$

w_{ij} is the distance between objects i and j (the weight of the edge in the graph)

$b_{ij} = \{0, \textit{the elements } i, j \textit{ belong to different clusters}; 1, \textit{the elements } i, j \textit{ belong to the same clusters}\}$

The algorithm looks like this (pseudocode):

```
G - graph with a set of nodes V and a measure of the distance W(v_i, v_j)
between the nodes
M - set of medoids (cluster centers)
C - dictionary of mapping of the node and the nearest cluster
k - number of clusters

# PAM algorithm
====================
def k_medoids(G, k): # input: graph G; specified number of clusters k
```

```
      M = {m_1, m_2, , … , m_k} # initialization of medoids: k random nodes of
the graph
      C = select_clusters(V, M) # distributes nodes between clusters
      M_new = M # keep updated centers at each iteration
      for epoch in range(n_epoch): # limited number of iterations
            # update cluster centers
            for i in range(k):
                  S_i = {v | C[v] = i} # elements of the cluster i
                  m_new_i = argmin(mean_distance(v, S_i)) , v ∈ S_i
            C = select_clusters(V, M_new) # redistribute nodes between
clusters taking into account new centers
            if M == M_new:  # if clustering has stabilized, we stop
iterating
                  M = M_new
                  break
            M = M_new
      return C, M     # output: mapping of clusters to nodes and centers of
the clusters themselves
======================
# finding the nearest cluster for each node
======================

def select_clusters(V, M):
      for v in V:
            C[v] = argmin(W(v, m_i)), i ≤ k
      return C
======================
# count the average distance from the center at the node v to each node of
the set S_i (cluster i elements)
======================
def mean_distance(v, S_i):
      return   1/|S_i| * Σ_{s ∈ S_i} W(v, s)
======================
```

The overall complexity of the algorithm is O(n * (n - k)) ~ O(n^2). Since at each iteration, the vertices are first distributed between the centers, and therefore the distance of all (n - k) vertices to all k centers is measured, i.e. the complexity of this step is O ((n - k) * k). Then optimal centers are selected: this is O (((n - k) 2) operations in the worst case.

3.2 Graph conversion methods for more informative clustering used in the work.

Cutting off groups with a small number of subscribers

For the analysis of the social graph, groups with a small number of users were removed, since such groups may insert too much noise in the data. To select the size of the group, which cut off the unpopular ones, a histogram of the size of the groups and a chart of the connection of the selected threshold with the average degree of nodes were constructed. On the one hand, to remove a lot of noise, but on the other hand not to make the graph too sparse, thereby losing useful information. Finally minimum users amount was chosen as 5.

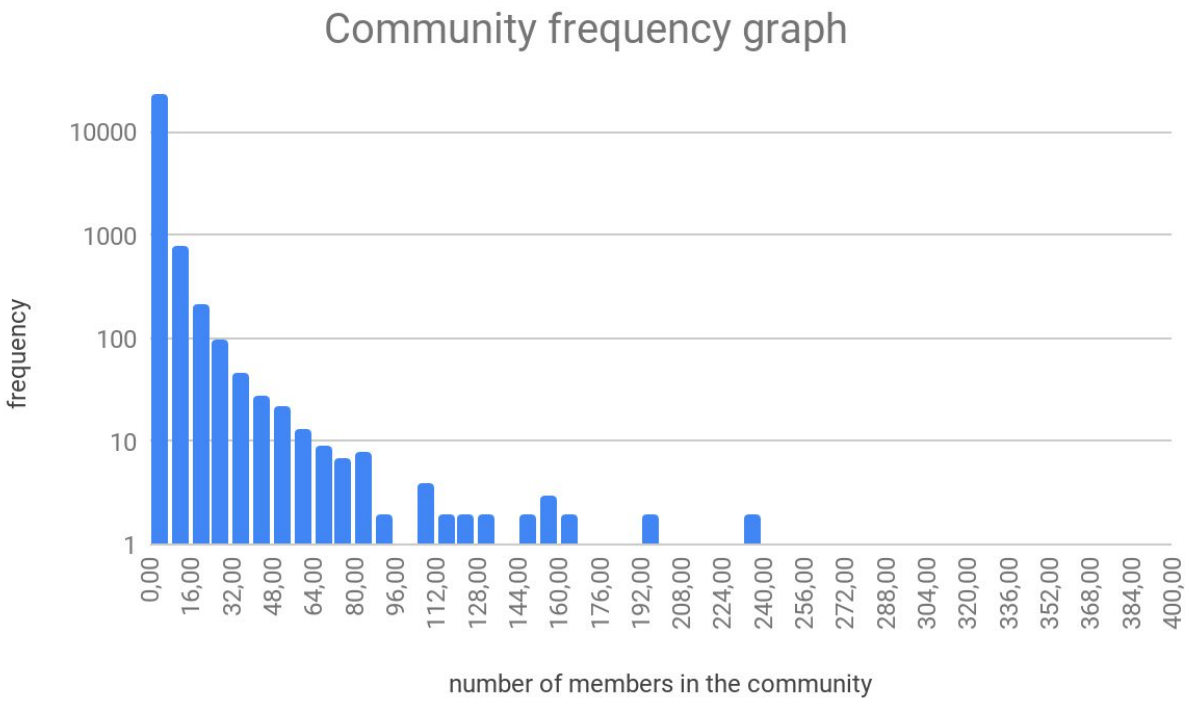

Fig. 3.1: Frequency of community members

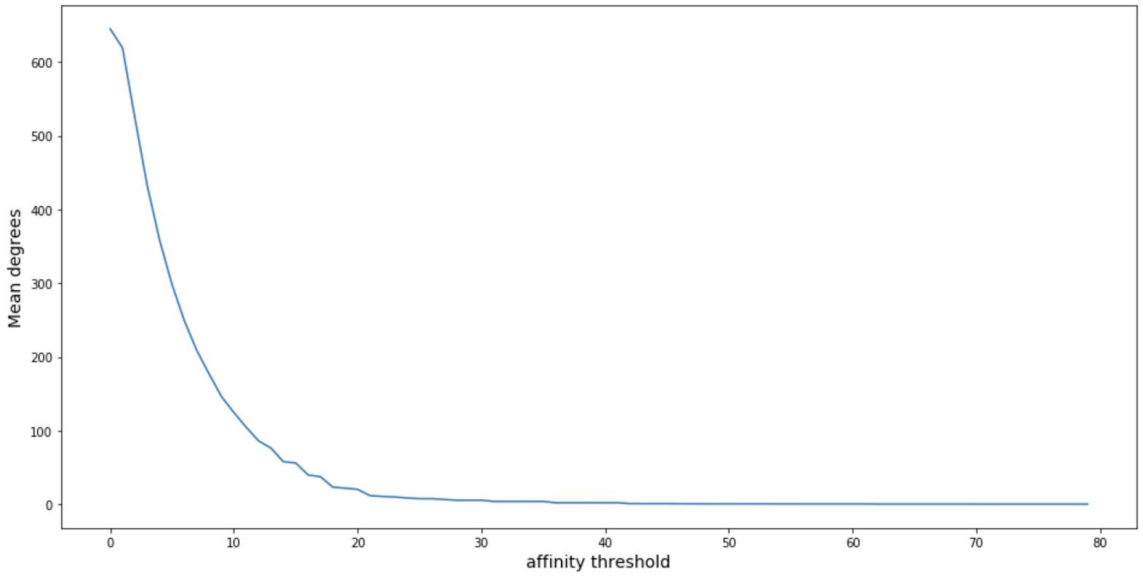

Fig. 3.2: The affinity threshold relationship with the average degree of vertices

Cutting off the edges of the graph by affinity

Also, all the edges, the weight of which by affinity was below a certain threshold (affinity_threshold), were initially cut off. Affinity is a selective estimate of how close the probability of an event (subscribing to two specific groups by the same user) to independence.

So, if affinity << 1, then the groups most likely have completely different almost non-overlapping audiences. For example, a group about car repair and a group about cosmetics. If the user is interested in auto repair, then most likely he will no longer subscribe to the women's makeup group. This is an example of opposing groups. If the value is close to 1, then the event (subscription to both groups) is independent. For example, a group about humor and a group about books. The user may like both of these interests, and each of them separately. If the value >> 1, then such groups are closely related to each other, then when subscribing to one group, they will most likely subscribe to another. For example, two similar popular groups of the same subject.

Based on this, the edges with a small affinity were removed, thus saying that there is no connection between the groups at all. Such a change, firstly, significantly reduced the size of the graph itself, and hence the complexity of the algorithm calculations. Secondly, groups with a very large number of subscribers were automatically cut off, because if almost everyone subscribes to such a group, then it ceases to carry information about the distinctive features of the users of this group and about its connection with other groups. Thirdly, such a cut-off of large groups is not a coarse filtering on the threshold of the number of users in it, and also takes into account their connectivity, thereby either the edges with smaller groups with a diverse unrelated audience can be filtered, or connections with larger groups that have an audience with strong interest may remain.

The *affinity_threshold* value was chosen equal to 5. This threshold cuts off 17% of the edges. If you look at the relationship between the size of the group and the average value of the affinity of adjacent edges (figure below), you can see that really small groups have affinity weights on average more than large groups.

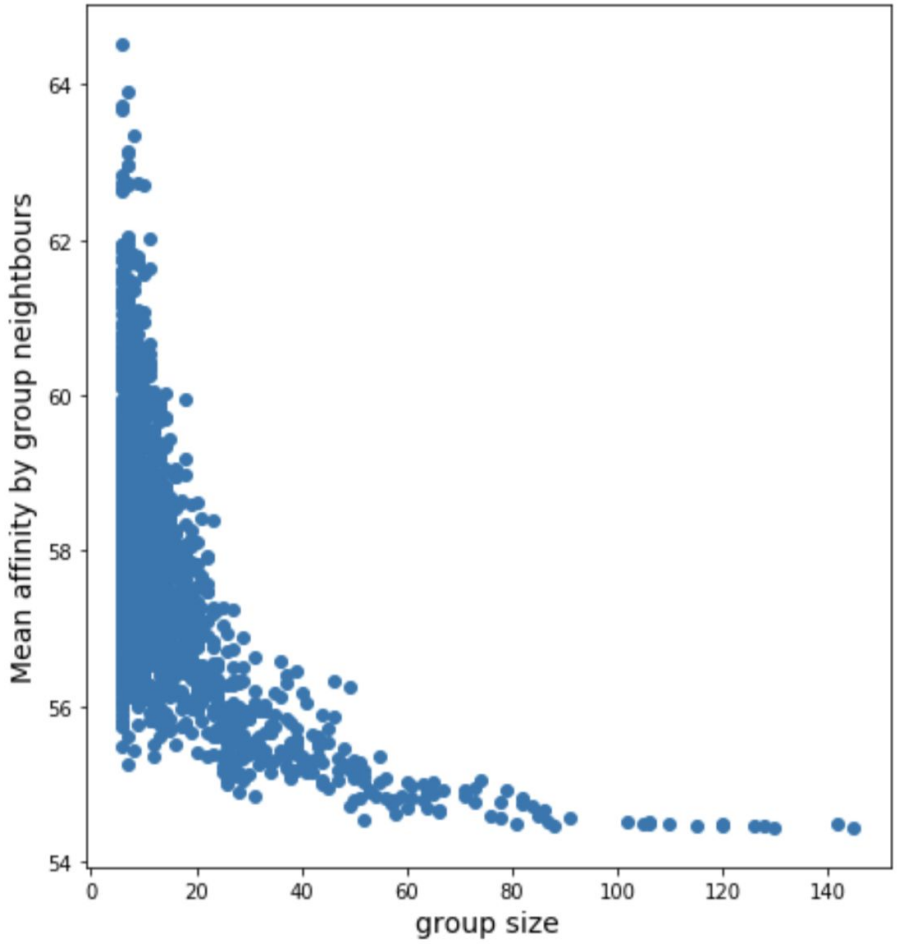

Fig. 3.3: Relationship between the size of the group and the average value of the affinity

You can also look at the dependence of the degree of connectivity of a vertex with its size. In this case, it is also clear that small groups have connections with a large number of other vertices.

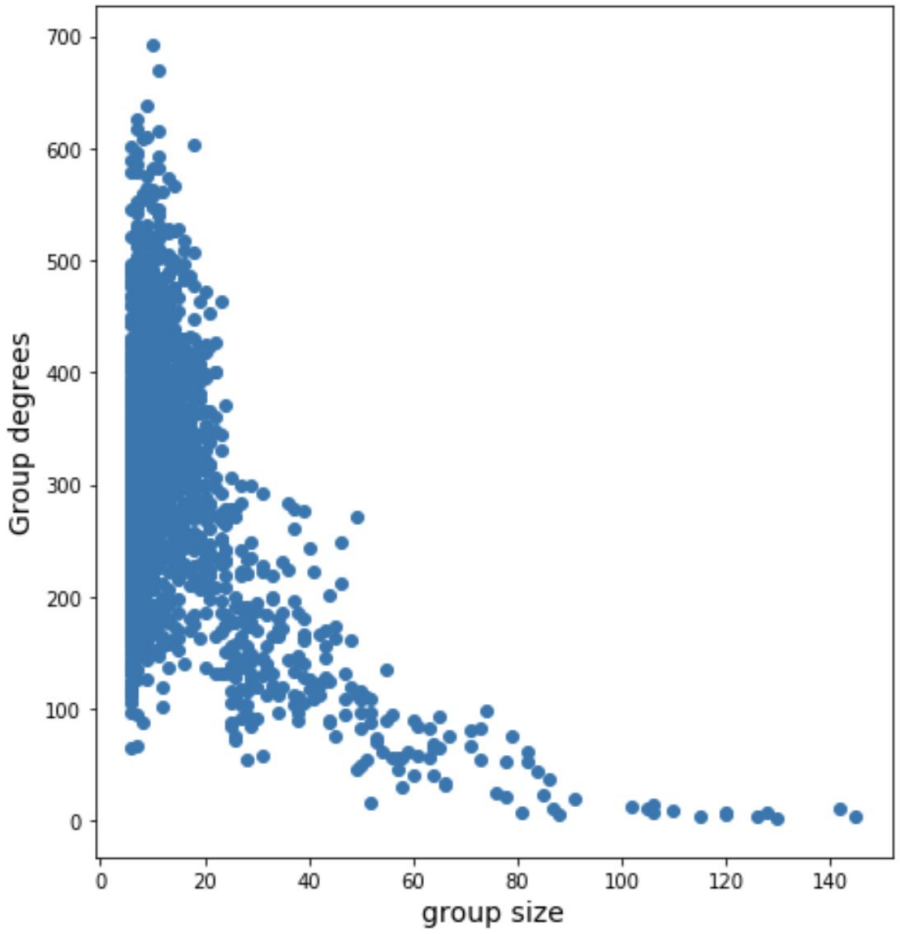

Fig. 3.4: Dependence of the degree of connectivity of a vertex with its size

Edge sampling by various methods

After clipping nodes by size and edges by affinity, the resulting graph remained connected. Moreover, the average degree of nodes is ~ 299. Problem of visualization of such a graph appears, as well as problem of its analysis. In this thesis, 3 data reduction approaches were considered: the cut-off by the threshold of the coefficient of connectivity, the local sparsification method, and nonlinear sampling depending on various coefficients of connectivity.

The first and easiest method: the cut-off by the threshold. Similar to affinity cut-off, you can set a threshold for one of the three coefficients of connectivity we are considering: Jaccard, Braun-Blanquet, and Overlap. Affinity gives us an estimation of the independence of the two groups, and the connectivity coefficient shows how strongly the groups are connected, not giving an advantage to small groups compared to large ones. Such coefficients estimate what proportion of users of the first group intersects with the second, and at the same time what proportion of users of the second group intersects with the first.

The second method: local sparsification. When using this method, not all edges are cut below a certain threshold, regardless of the groups themselves, but a local sparsification of the edges of each group occurs. To do this, first, for each node v its degree d_v is determined, then all edges of this node are sorted in increasing order of the connectivity coefficient and only the top max(1, d_v^e) edges are left, where e - an arbitrary coefficient from 0 to 1. The edge remains in the graph if it hit the top edges of at least one of its nodes. Thus, at each node, only part of its edges is removed. This method ensures that the graph's connectivity is guaranteed, the appearance of isolated nodes is avoided, and the average degree of nodes and the complexity of the graph itself are significantly reduced.

The third method: non-linear sampling. With this approach, a nonlinear function of the weight (connectivity coefficient) of the edge, which will give the probability with which the edge will remain in the graph, is selected. So, edges with smaller weights will be removed with a much greater probability than strongly connected edges. In this thesis, the function was considered as such a nonlinear function:

$$f(x) = ((x - weight_{min}) / (weight_{max} - weight_{min}))^{\alpha} * (p_{max} - p_{min}) \quad (3.4)$$

where α is an arbitrary parameter. If α is greater than 1, then the increase in probability depending on the size of the weight will be slower, thereby throwing more edges with small weights. In this thesis, α was chosen to be 5.

The first experiments were carried out with the Jacquard coefficient. The threshold for the "by threshold" method was experimentally chosen to be 0.1. The left shows the distribution of weights before sparsification (without sampling), as well as after sparsification in three ways separately (by threshold, local sparsification, by nonlinear function). On the right are the same histograms, but already with a logarithmic scale, in order to more clearly consider all the changes.

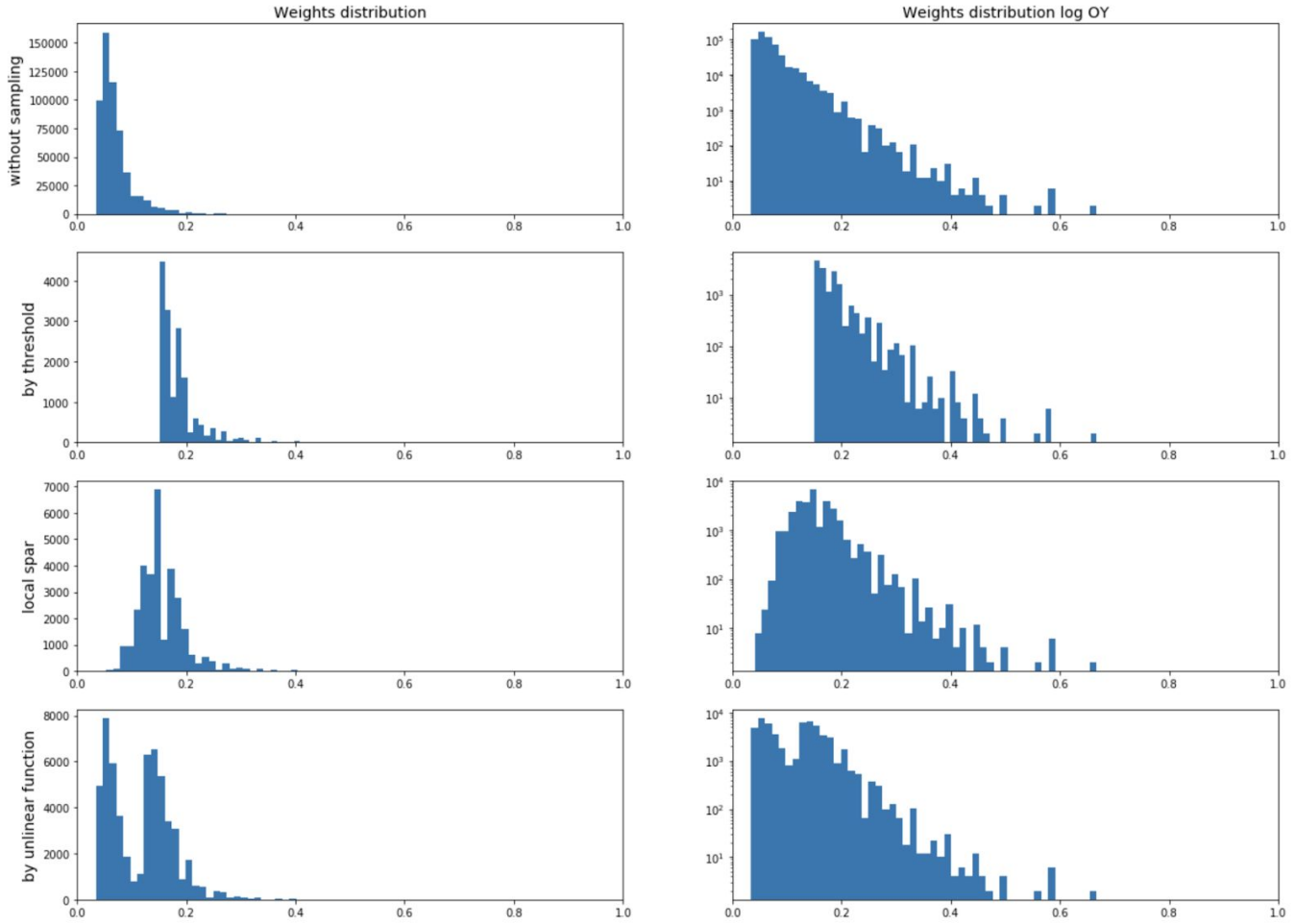

Fig. 3.5: Sampling with weights according to the Jaccard coefficient

The data on the distributions obtained are presented in the table:

	edges count	mean	standard deviation	mean node degrees	separated nodes	connectedness	R^2
without sampling	549816	0.069	0.031	299.31	0	true	0.83491
by threshold	59690	0.138	0.037	32.81	46	false	0.92649
local sparsification	56648	0.135	0.041	30.84	0	true	0.84919
by nonlinear function	55296	0.117	0.058	44.26	5	true	0.83491

Table 3.1: The data on the distributions with weights according to the Jaccard coefficient

After any of the three samples, the average value of the coefficient is shifted to the larger side, since the smallest values were filtered with a larger coefficient. The variance also increased, which means a greater variety of weights, which means that the graph is obtained with more pronounced connections.

Only when using the threshold cut-off method, separate connected components appeared, but non-linear sampling also separated several nodes. This can be explained by the fact that the threshold cut-off method roughly filters edges, not paying attention to connectivity, unlike the *local sparsification* method, and the *nonlinear sampling* gives a chance to small weights, thereby reducing the probability of new components appearing.

The average degree of the node in all experiments decreased markedly, which greatly facilitated the visualization of the graph itself.

Similarly, sparsification was constructed using two other connectivity metrics: Braun-Blanquet and Overlap.

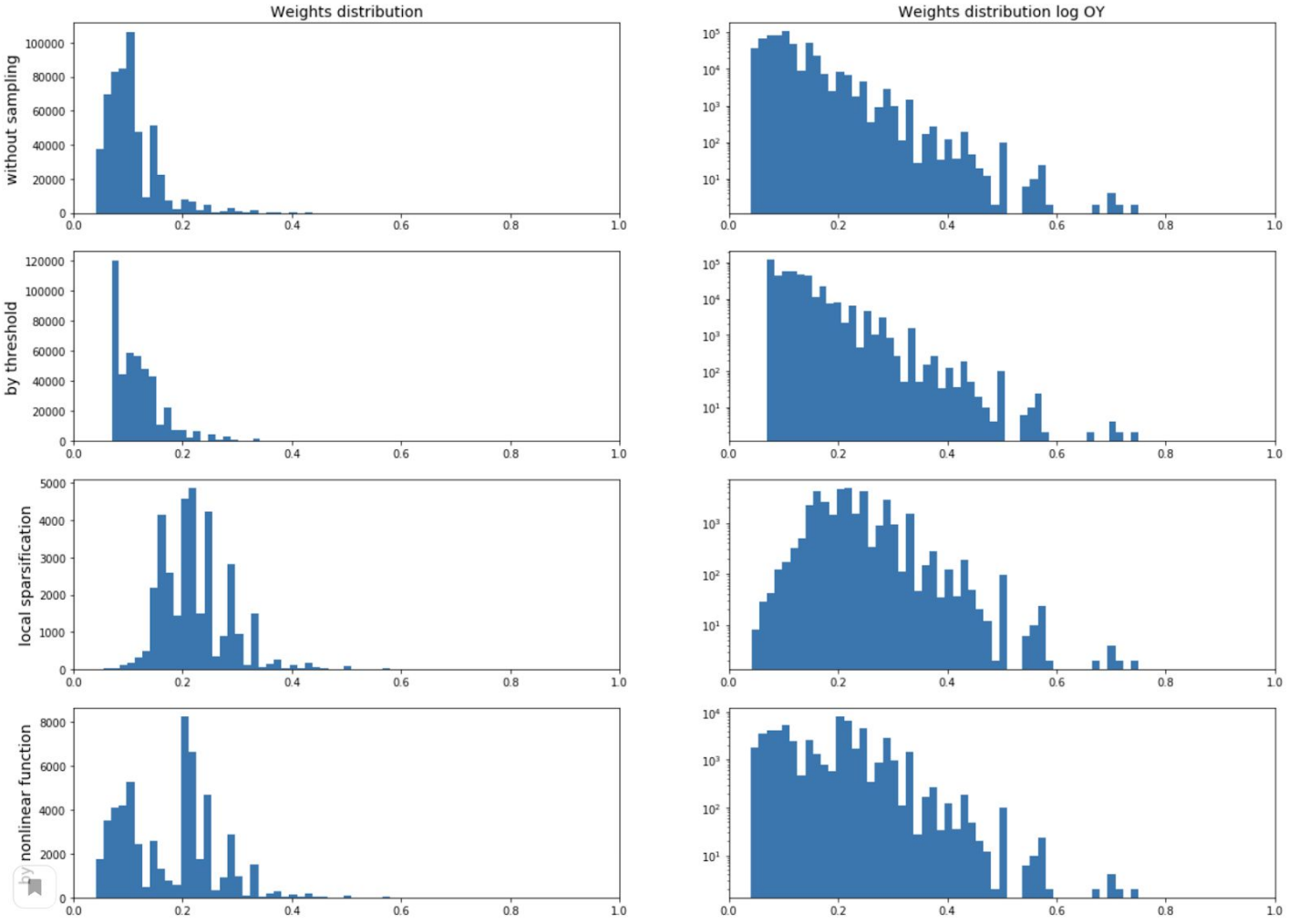

Fig. 3.6: Sampling with weights according to the Braun-Blanquet coefficient

	edges count	mean	standard deviation	mean node degrees	separated nodes	connectedness	R^2
without sampling	549816	0.106	0.047	299.31	0	true	0.85299
by threshold	59848	0.197	0.051	40.11	19	false	0.95776
local sparsification	56280	0.201	0.059	30.63	0	true	0.85441
by nonlinear function	56380	0.175	0.084	44.92	9	true	0.85299

Table 3.2: The data on the distributions with weights according to the Braun-Blanquet coefficient

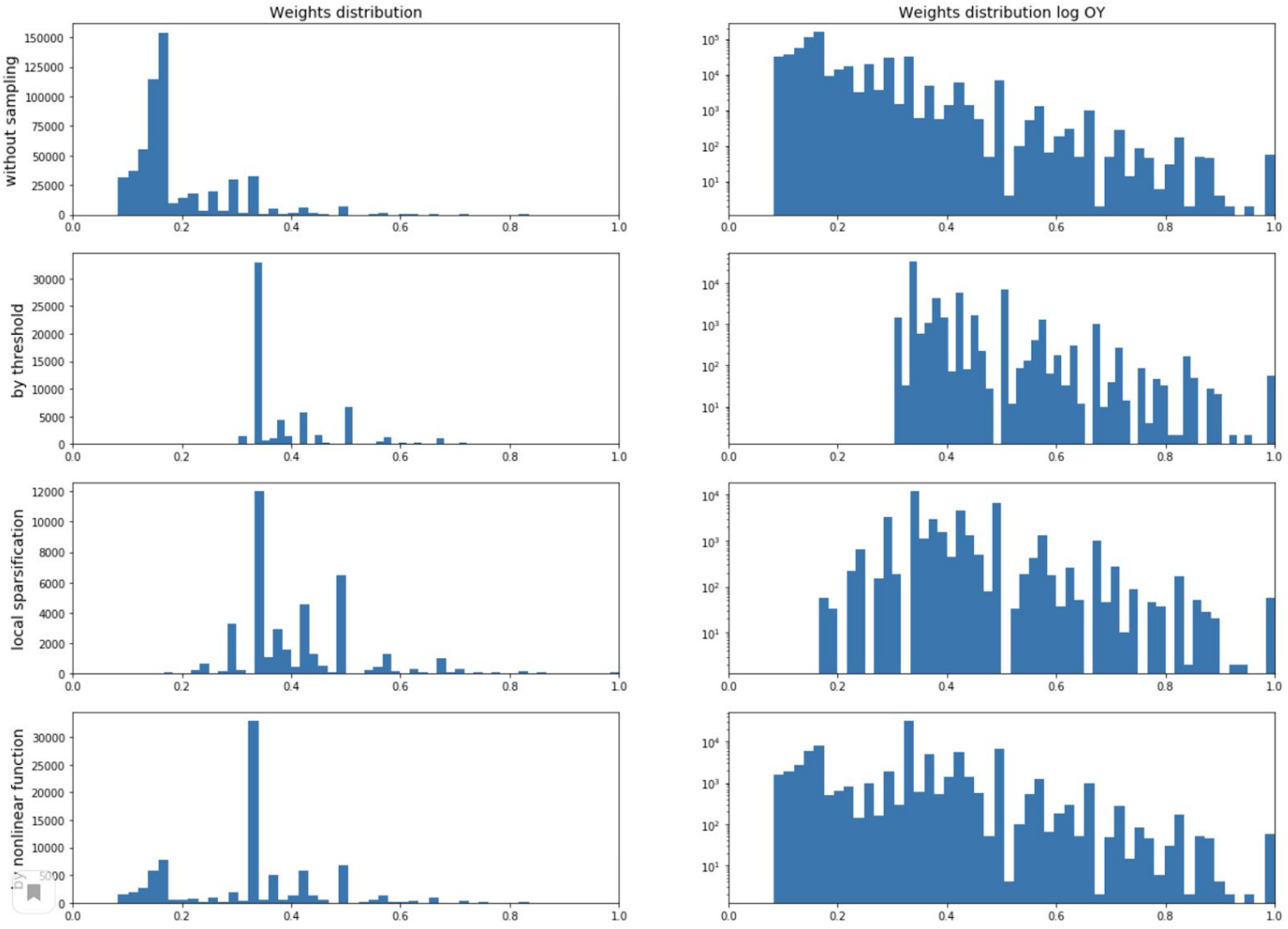

Fig. 3.7: Sampling with weights according to the Overlap coefficient

	edges count	mean	standard deviation	mean node degrees	separated nodes	connectedness	R^2
without sampling	549816	0.188	0.089	299.31	0	true	0.91301
by threshold	59380	0.393	0.092	32.34	1	false	0.99132
local sparsification	53440	0.386	0.105	29.09	0	true	0.96080
by nonlinear function	58940	0.325	0.133	58.38	0	true	0.91301

Table 3.3: The data on the distributions with weights according to the Overlap coefficient

As can be seen from the charts, the Jaccard metric has lower values compared to other metrics, while the Overlap metric gives the highest values. This result is justified by the fact that in the denominator of the Jaccard coefficient is the union of the power of groups, and therefore the proportion of intersection takes into account, both with the first and the second groups. While the other two metrics estimate only the fraction of the intersection with one group: Braun-Blanquet - the highest power, Overlap - the lowest power (visually visible in the figure below). Since the initial size distribution of groups is greatly biased downward, the proportion of edges connecting any group with a “small” one is much larger than any group with a “large” one. That is why the distribution of the Overlap coefficient, the most uniform among the others, has the largest value of the mean and variance.

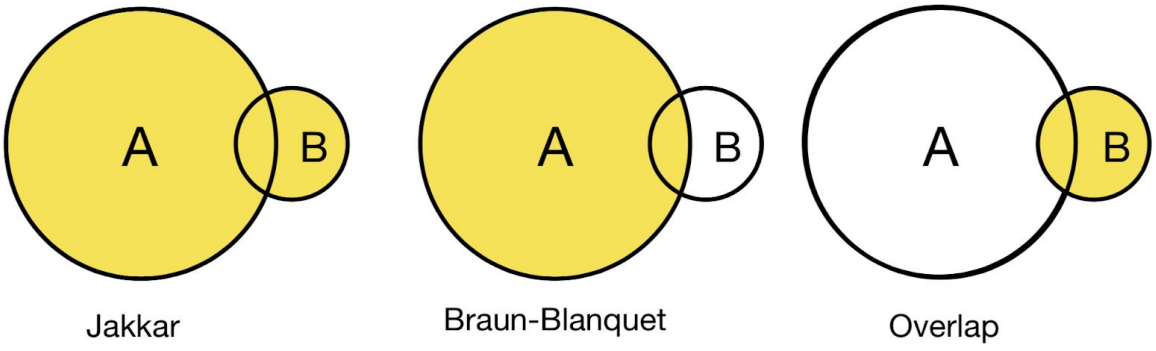

Fig. 3.8: The yellow color indicates the set, whose power in the denominator of each of the coefficients

The closest distribution to the selected function (8) from all sets of weights obtained by different sampling methods is given *by the threshold* method on Overlap coefficients.

If you look at the distribution of degrees of connectivity of nodes, you can see that the *local sparsification* method greatly reduces the degree of each node, which is not surprising based on the formula. But then during clustering a situation when the node has no edge to any of the cluster centers will often arise. Therefore, this method is not suitable for clustering, but it does an excellent job with improving visualization: each mode retains its most characteristic neighbors, which allows for quite informative placement of all groups in a two-dimensional space, while significantly reducing the number of edges.

For clustering, on the basis of the distributions obtained, the *by threshold* method is more suitable. Since it has an average weight shift to the higher side than the method *by nonlinear function*, it also reduces the average degree of the node more strongly. Therefore, this method has shown in practice the best result in graph clustering.

3.3 The results of data clustering when applying various methods of graph conversion.

Selection of the number of clusters

Since the clustering of k-medoids requires the initial assignment of the number of clusters as a hyperparameter of the model, it was necessary to understand how many clusters the graph under study is better divided to. A measure of the quality of clustering is set arbitrarily, based on a specific task. Usually, some ideas about the structure of the graph itself, about the expected distribution of clusters and their other characteristics are made. Therefore, to create a quality criterion for the model, the following assumptions were made:

- a distinctive theme stands out in each cluster
- the number of brightly expressed clusters is not very large (up to 10), since the number of non-overlapping interests is not large (for example: humor, auto, cosmetics ...)
- clusters do not differ much in size, since groups with a small number of subscribers and groups without a peculiar specific audience were filtered out, thereby leaving only groups with a clearly expressed theme (and there are not many of them in general).

Based on these assumptions, the measure of the mean deviation of the cluster size was chosen as a criterion, at the same time trying to isolate as many clusters as possible.

Clustering stability

As it is known, the k-means algorithm, and therefore the k-medoids, is very unstable to the choice of the initial position of the centers of the clusters. Therefore, in this thesis, we considered two ways to select the initial medoids.

The first method: you can spend several iterations of randomly determining the initial position of the clusters and choose the clustering, where the variance value is less.

The second method is more iterative.

- the first center is selected randomly
- for the second center, N random nodes are taken and the center is assigned to the node that is farther from the first center
- To determine the k-th center, N nodes are also taken first and the node furthest from the previous (k-1) centers is assigned as the k-th center.

The second method gave the best result. N was taken equal to 100.

Cluster analysis by received graph

As a graph for the analysis, we took a graph with weights calculated by the Braun-Blanquet coefficient, with preliminary sampling using the local sparsification method. Because this choice of conversion best cuts the edges for visualization. Clustering was carried out on a graph with sampling by the threshold method on Jaccard scales.

To visualize the graph obtained, the force-graph JS library was used. Force-graph library is a web component to represent a graph data structure in a 2-dimensional canvas using a force-directed iterative layout. Uses HTML5 canvas for rendering and d3-force for the underlying physics engine. Supports canvas zooming/panning, node dragging and node/link hover/click interactions. [10]

The following methods are available in this library: node identification, link object accessor attribute referring to ID of the source node, set background colour, node colour and especially important is the availability of the method, that allows specifying the distance for edges according to the specified parameter.

Below is a script for visualizing and an example of the input file for it.

```
<script>
    fetch('../datasets/clusters_new_small_update.json').then(res => res.json()).then(data => {

        const elem = document.getElementById('graph');
        const Graph = ForceGraph()(elem)
            .backgroundColor('#000000')
            .nodeAutoColorBy(
               node => node.cluster
            )
            .nodeLabel(node => `${node.id}: ${node.diameter}`)
            .linkColor(() => 'rgba(255,255,255,0.5)')
            .zoom(0.2)
            .linkWidth(0.05)
            .onNodeClick(node => window.open(`https://vk.com/public${node.id}`, '_blank'))
            .graphData(data)
            .d3Force("link", d3.forceLink().distance(d => d.distance));
    });

    function getRandomArbitary(min, max)
    {
        min = Math.ceil(min);
        max = Math.floor(max);
        var res = Math.floor(Math.random() * (max - min + 1)) + min;
        return res;
    }

</script>
```

Fig. 3.9: Graph visualization code

clusters_new_small_update.json


```
{"nodes": [{"diameter": 145, "id": "61615047", "cluster": 10}, {"diameter": 142, "id": "107978388", "cluster": 10},
{"diameter": 130, "id": "63731512", "cluster": 10}, {"diameter": 128, "id": "101965347", "cluster": 10},
{"diameter": 126, "id": "72495085", "cluster": 10}, {"diameter": 120, "id": "88647129", "cluster": 10}, {"diameter":
 120, "id": "57846937", "cluster": 10}, {"diameter": 115, "id": "29573241", "cluster": 10}, {"diameter": 110, "id":
"4808406", "cluster": 10}, {"diameter": 106, "id": "30022666", "cluster": 10}, {"diameter": 106, "id": "56106344",
"cluster": 10}, {"diameter": 105, "id": "34215577", "cluster": 10}, {"diameter": 102, "id": "31976785", "cluster":
10}, {"diameter": 91, "id": "41437811", "cluster": 10}, {"diameter": 88, "id": "6527899", "cluster": 3},
{"diameter": 87, "id": "52537634", "cluster": 10}, {"diameter": 86, "id": "60028089", "cluster": 10}, {"diameter":
85, "id": "65", "cluster": 10}, {"diameter": 84, "id": "28905875", "cluster": 10}, {"diameter": 82, "id":
"38683579", "cluster": 10}, {"diameter": 82, "id": "49439086", "cluster": 10}, {"diameter": 81, "id": "22798006",
"cluster": 10}, {"diameter": 79, "id": "29559271", "cluster": 10}, {"diameter": 78, "id": "76982440", "cluster":
9}, {"diameter": 78, "id": "26750264", "cluster": 10}, {"diameter": 76, "id": "115608422", "cluster": 10},
```

```
"links":[{"source":"54343520","target":"70841494","Affinity":"82.555557","Jaccard":0.500000,"BraunBlanquet":0
.666667,"Overlap":0.666667},{"source":"104613914","target":"82724249","Affinity":"79.607140","Jaccard":0.666667,
"BraunBlanquet":0.750000,"Overlap":0.857143},{"source":"35623118","target":"37867179","Affinity":"75.816330",
"Jaccard":0.555556,"BraunBlanquet":0.714286,"Overlap":0.714286},{"source":"70061151","target":"61115705",
"Affinity":"70.761902","Jaccard":0.444444,"BraunBlanquet":0.571429,"Overlap":0.666667},{"source":"135731379",
"target":"97015440","Affinity":"70.761902","Jaccard":0.444444,"BraunBlanquet":0.571429,"Overlap":0.666667},
```


Fig. 3.10: Example of the input file for graph visualization

Figure 3.10 - 3.11 shows the results of visualization of the resulting graph. In a circle, there are groups, all the edges of which were removed after sampling. When constructing the graph, the length of the edge was chosen in inverse proportion to its weight, so that groups with a high coefficient of connectivity were located close, and strongly differing groups were, on the contrary, far away.

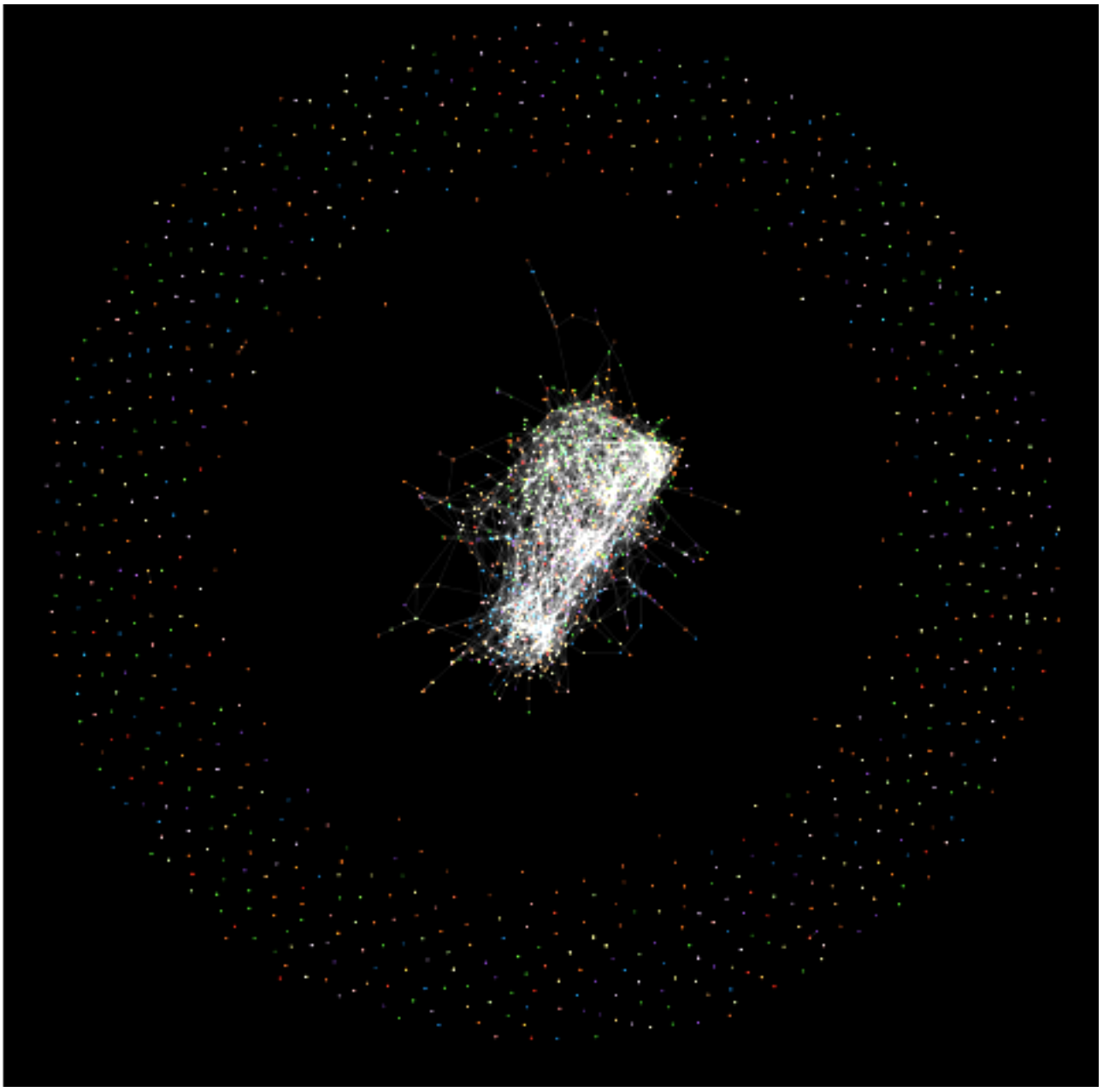

Fig. 3.11: Visualization of the resulting graph

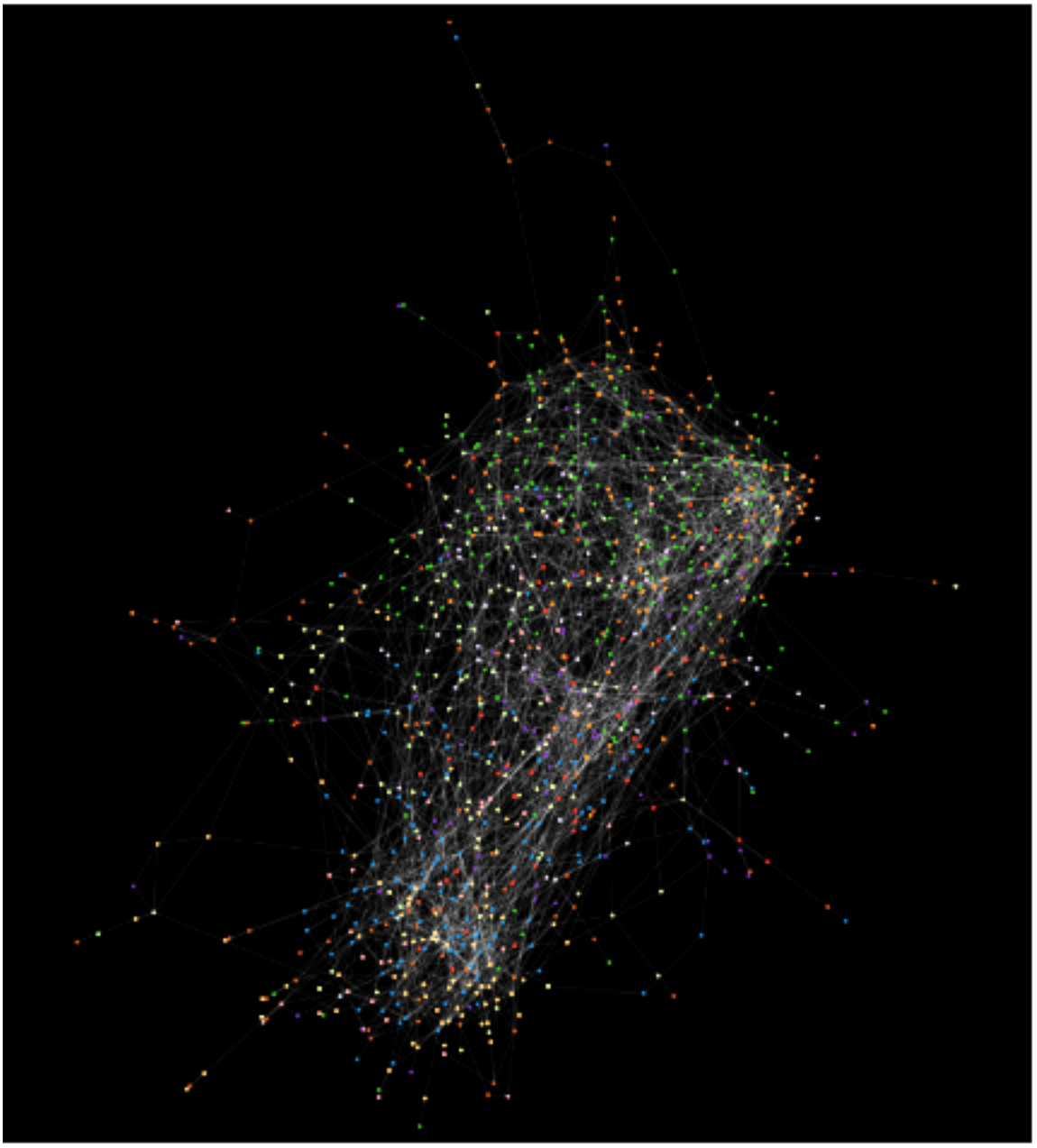

Fig. 3.12: Visualization of the resulting graph (zoomed view)

Let us analyze the results obtained in which types of communities are students of Kazan Federal University and if there is a cluster of professional communities among them. You can get all community IDs in each cluster and go to these “VKontakte” pages.

In Figure 3.11, the orange, green, purple, yellow, blue and pink clusters stand out well. Consider what type of community prevails in each cluster.

In the upper right corner of the orange cluster, you can clearly highlight the **beauty communities**.

Examples of community names in the orange cluster:

Manicure | Nails, Fashion - fashion and style, Beauty salon, Beauty secrets, Studs | Women's Magazine, Modern Girl, Fashion blog, Ideas for beauty, Beautybook - secrets of female beauty,

manicure, Women's Tricks, Hairstyles | Haircuts | Coloring, fashionista, Woman, Quotes and statuses (Women's magazine)

The same orange cluster at the top divided our communities into communities **about humor and sports communities.**

Examples of communities in this cluster are:

Clear Jokes, Sports Articles | Fitness, Smiled, BBET sports predictions, Fun Time - male humor, Like - Crazy fun, Laughter Corporation, Awesome fun, Funny Before Pain, Indecent Jokes, Funny Pictures, 4MEN, Men's Thoughts, Black Science, Football 24 | Champions League, Sports Forecasts.

Another big cluster is the **communities about hobbies** - green color (cooking, traveling, handicrafts).

This cluster includes the following communities:

Fitness and Health, Interior Design, High Kitchen, Travel | Interesting facts, Needlework Club, Russian for lazy people, Tasty - Quick recipes, Live In Tattoo | Tattoos, IDEAS for creativity, Creative | The hands, Interesting planet - travel and tourism, I love sports

Implicitly in the center of the graph highlighted purple cluster. It includes completely different **non-intersecting communities** about shopping, science and other things:

Personality Psychology, Learn English, Girl's AliExpress, Intellectuals - Smart Magazine, Read, AliExpress, LolySweet Online Clothing Store

Consider the lower part of the graph. It clearly distinguishes yellow and blue clusters.

In a yellow cluster with a strong predominance, **professional communities of programmers** appear, such as:

CyberForum.ru - Forum of programmers and system administrators, System administrator, Timeweb: everything about hosting, IT and Web, VK Gaming, Games for Android, Learning Java from scratch!, Frontend Raccoon, IT Daily, Webtackles - web development, front-end, I am a DEVELOPER, ProgHub | Community of Top Programmers, Master in MIPT, Java, Hexlet (Practical Programming Courses), Droider.ru - A Gadgetless Website, Web Programmer - PHP, JS, Python, Java, HTML 5, National Open Education Platform, TargetHunter, Infogra.ru - Best for Designer, Toaster (Question and Answer Service for IT Specialists), Business Club, Dribbble - community of designers who show their projects and process, Harvard Business Review Russia, Web Design Workshop

In the blue cluster there are communities for **self-enhancement**, **business** and again **programming**:

Freud | School of psychology and self-development, Psychology on the couch | Psychology, BUSINESS Online, Area of opportunities, localhost, I am a designer, Son of a Programmer, Entrepreneurs of a new generation, Code magazine (Programming without snobbery), PSD | Design Space, Data Mining | Data Analysis, VK Designers, motivation to learn.

3 clusters are scattered throughout the graph. In the light orange cluster, there are communities **about music, art and cinema:**

E: \ music \, MUJUICE (musician), Academic Music, T-Fest (music performer), Art, Museum of Fine Arts of the Republic of Tatarstan, MOTHERLAND / Music of local bands, other movies, Masterpieces of cinema, STUDIO 21 (radio station)

In the red cluster, there are communities of **narrow interests** (comics, books in English, advertising):
MyComics | Marvel, Practical Psychology, English Books, Where to fly budget, GLOBAL FASHION, Smart gifts, Looking for a Kazan model.

In the light-violet cluster you can see the communities about **Kazan and the Tatar language**: Islam is a religion of peace and good, Kazan: city and citizens, TYPICAL TATARIN | Kazan, Tatarstan, Work in Kazan.

Based on this analysis, certain conclusions can be drawn. It should be noted that if the graph is conditionally divided into two parts, then in the upper part there will be large clusters of students' personal interests, not related to education. These are mainly entertainment communities about humor, sports and beauty. While the clusters associated with education are located far away from them and are located in the lower part of the graph.

If we talk about the share of user communities about education, then the graph shows that quite a few students are interested in programming.

Conclusion

The main results of this thesis are as follows:

1. In this thesis, methods of social network analysis were studied.
2. First, the topic of social network analysis theory was considered. Formulated problems that may arise in the analysis.
3. In the master's thesis, the main features and approaches of social network analysis are investigated.
4. Existing programs for researching the social network graph were reviewed.

The main goal of the work was to create a weighted graph model and implement data clustering based on their specifics based on social network materials. Thus, the main aspects of the graph analysis were investigated and analyzed.

1. Based on the materials studied and the analysis of the social network, a social network model was created in the form of a weighted graph. Analysis was carried out selecting the weights in the graph.
2. For the analysis of the weights obtained, an empirical and theoretical law of the distribution of the weights of the edges was constructed. This made it possible to facilitate the analysis of the structure of the graph in the future.
3. In the course of work for more informative clustering it was necessary to convert the graph. For this, various methods of graph sampling were applied and a suitable method for calculating the weights of the edges was identified.
4. In addition, according to the obtained model, the graph was clustered. After visualization of the clustered graph, the obtained results were analyzed on what types of communities the students of Kazan Federal University are in and whether there is a cluster of professional communities among them. The study revealed a cluster of professional communities, analyzed the structure of this cluster and its relationship with others.

As for further research, there are several possible directions.The first is to create a methodology for optimally selecting the clustering of a social network graph, depending on the nature of the data being processed. The second is the identification of the dependence of the student's progress on the cluster to which the interesting pages to which the student is subscribed belong.The third is the construction of a graph, at the vertices of which there are other data of the network user profile for a more complete analysis of the progress. In the future, it is planned to consider more complex forms of dependencies.

This will require the construction of a joint law of multidimensional probability distribution of the values of the quantities under study.

Bibliography

[1] **N.F. Gusarova.** "**Analysis of social networks. Key concepts and metrics".** University of Information Technologies, Mechanics and Optics (ITMO), St. Petersburg, 2016. pp. 4-5, pp. 52-55 (in Russian).

[2] **M. Newman. " Modularity and community structure in networks"**, PNAS Vol. 103, N 23, 2006, pp 8577-8578.

[3] **K. Crowston, J. Howison, A. Wiggins. "Validity Issues in the Use of Social Network Analysis for the Study of Online Communities"**. Under Second Round Revision for the Journal of the Association of Information Systems (JAIS), 2010. pp. 12 - 18, pp. 20-23.

[4] **M. Newman. "The physics of networks".** Physics Today, Vol. 61, Issue 11, 10.1063/1.3027989, 2008.

[5] **Porter, Mason A., Jukka-Pekka Onnela, and Peter J. Mucha. "Communities in Networks"**. Notices of the AMS, Vol. 56 N 9, 2009, pp.1083 - 1091.

[6] Similarity measure. Avaliable at [https://ru.wikipedia.org/wiki/Коэффициент_сходства] (in Russian)

[7] The method of least squares. Avaliable at [https://en.wikipedia.org/wiki/Least_squares]

[8] **Hae-Sang Park, Jong-Seok Lee and Chi-Hyuck Jun. "A K-means-like Algorithm for K-medoids Clustering and Its Performance."** Department of Industrial and Management Engineering, POSTECH San 31 Hyoja-dong, Pohang 790-784, S. Korea

[9] **Hae-Sang Park, Chi-Hyuck Jun. "A simple and fast algorithm for K-medoids clustering" ,** H.-S. Park, C.-H. Jun / Expert Systems with Applications 36 (2009) 3336–3341

[10] Force-graph library. Avaliable at: [https://github.com/vasturiano/force-graph]