|  |  |  |  |
| --- | --- | --- | --- |
|  | [Pick the date] |  | |
|  | |  | |
| ***Final Project***  *Distinguishing between graphs with no sensitivity to sampling* | | | |
|  |  | | Barak Ye’ela 208745919 and Boker Dor 209271279 |
|  |  | | Supervisor:  Dr. Lozoun and Prof. Yanz |

*Final Project*

Distinguishing between graphs with no sensitivity to sampling

# **Abstract**

Graph clustering is a well-known problem [1][2][3][4][5][6]. In this paper, we would like to distinguish between sampling of networks in time. To put it differently, given a sample of a network in a specific time, we would like to be able to classify to which network it belongs. This requires a method to characterize networks in a way that is not sensitive to the sampling, using measures We propose a novel attitude in that field. Regardless of current methods [7][8][9][10], we do not look for similarity in properties but in the relationships between them. We show significant results on several networks, both on real networks and on simulated networks, using a mathematical model, as detailed below.

# **Introduction (The Problem)**

The process of identifying this structure in terms of grouping the data elements is called clustering, also called data classification [152]. The resulting groups are called clusters. The grouping is usually based on some similarity measure defined for the data elements.

The ability to distinguish between graphs with no sensitivity to sampling can have a major impact in both military and civilian fields.

Let us assume that we have emails exchange between members in a few different institutions. Giving two networks, representing the email exchange in different institutions, in different times, we would like to classify the emails by institution.

By doing that, we can, only by two samples of two networks in time, conclude how close is the structure of these networks is.

In the security aspect, we could recognize an attack on our systems, by identifying that in a certain time the network is abnormal, by our criteria, compared to any other moment in time. Not only that, but also we could identify abnormal communication between hostile factors that might indicate a hostile action. The huge benefit of using this method is that it doesn’t require reading any knowledge of the content of the communication, only who talked to whom when, since it is not always possible because of lack of resources or encryption problems.

So far, most current distinction methods in networks look for similarity in properties of the network such as node degree, density centrality and so on[7][8][9][10].. However, these parameters are very dynamic. To emphasize, let us look at two networks: a group of friends and a professor with his students. It can clearly be seen how this methods fail dramatically. The number of messages, for example, can be higher or lower than average depends on the time checking, and cannot indicate well to which of the groups it belongs. Thus, we here suggest a novel method to distinguish between the networks that is not sensitive to sampling. We are using the same features, but instead of looking at the features themselves, we look at the ratio between them. This way, we don’t look at the average number of messages, but the connection between the average number of messages to the density. This way, we get a clearer look on the nature of the network. The proposed method can be applied to directed and undirected, weighted and unweighted graphs.

# **The Goal**

The goal of the project is to distinguish between graphs in a way that will not be sensitive to sampling. Given numerous graphs, which are samples in time of a group of networks, this algorithm divides the graphs and determines which of the graphs belong to each network

The goal of the project is to find a vector of parameters, which are functions of features of the graph that will position the graph and determine

**Mathematical Model**

We tested our method using both real world networks and stimulated ones.   
For the second part, we took a network and produced different graphs in different activation times in a probabilistic method, as follows:

Every edge receives a value such that . We chose a T parameter between 0 to 3. Each produces a different sample.   
For every edge we take a random number between 0 to 1:  
If : we shall turn the edge on, and consider it a part of the graph.  
If : we shall turn the edge off, meaning it is not a part of the graph.  
To point out, when T tends to infinity, and the statement is always true. On the other hand, when tends to , . Since, the graph has no edges. In short, this model represents all levels of activation in proportion to .

# Algorithm description

The algorithm consists of four steps. Each step will be described in detail below.   
There are two phases to this method: the training phase, and the testing phase.

* The first step is to calculate a list of features for each graph.
* In the second step we fit a regression line for every two features, which defines the connection between them.
* Then, we select pairs of features that separate the graphs optimally in the learning stage.
* At this point, each sample is defined as a vector of consts (each element is the constant computed in the regression for two features)
* Finally, in order to know whether two samples belong to the same graph, all is needed is to compute the [Euclidean](https://www.google.co.il/search?rlz=1C1CAFB_enUS602IL609&q=euclidean&spell=1&sa=X&ved=0ahUKEwiD7KKVm-jWAhXGJFAKHZm3BS8QvwUIIigA) distance between them. To receive the full division, machine learning can be used, as follows.

**Basic Definitions**

Graphs are structures formed by a set of vertices (also called nodes) and a set of edges that are connections between pairs of vertices.  
Clustering of sets of graphs is the task of grouping the different graphs into clusters taking into consideration the structural similarity between them.

A directed graph is a graph in which edges have orientations.  
A simple graph is an undirected graph in which both multiple edges and loops are disallowed.

Strongly connected component is a subgraph in which any two vertices are connected to each other by paths and are not connected to any vertices which is not in the subgraph.

#### Adjacency matrix is a square matrix whereas Aij is 1 if there is an edge from the node I to the node j and zero otherwise and zeroes on its diagonal.

#### Laplacian matrix- the elements of the laplacian matrix are defined as:

Note that the Eigen values of both the laplacian matrix and the adjacency matrix are always positive.

*Triad*- is two edges with a shared vertex.

**The Features-**

The following are the features we used in order to define a network. Please note that the features listed are for both directed and undirected graphs, our method will work for both cases.

Average degree- average of the degree on the number of nodes.

##### Degree centrality-

Such that: are the nodes, is the node with the highest degree and

##### Betweenness- Node with maximum betweenness. Betweeness of a node is defined as:

where V is the set of nodes,  is the number of shortest  paths, and  is the number of those paths passing through some node  other than . If , , and if , .

##### Normalized Betweenness- The node with the highest load centrality. Load centrality is defined as the fraction of all shortest paths that pass through that node.

##### Edge betweenness- Maximum betweenness centrality on all edges in the graph.

Betweenness centrality of an edge e is the sum of the fraction of all-pairs shortest paths that pass through e:

where  is the set of nodes; is the number of shortest  paths;  is the number of those paths passing through edge .

Number of strongly connected components

##### Maximum size of connected component-We shall take all the strongly connected components, and look for the one with maximum nodes

##### Average number of nodes in strongly connected components- of all the strongly connected components we computed the average of the number of nodes

##### Average distance between every pair of nodes- We computed the average distance between every two nodes that have a path in the graph.

##### Diameter- biggest distance between two vertices in the graph

##### Density- in a directed graph and in an undirected graph

##### Maximum Eigen value of the adjacency matrix

##### Minimum Eigen value of the adjacency matrix

##### Maximum Eigen value of the laplacian matrix

##### Minimum Eigen value of the laplacian matrix

##### Small world- Typical distance L between two randomly chosen nodes

##### Motifs of size 3 -This feature reflects the ratio between directed tringles and undirected tringles. Directed tringle is a tringle with importance of direction. Undirected tringle is a tringle without importance of direction. In other words, three edges where each one is connected to the other two.

Undirected Triangle

Directed Triangle

That is to say, number of tringles in the graph divided by the number of undirected tringles.   
Note these feature is only available in a directed graph.

Transitivity- the fraction of all possible triangles present in G. Possible triangles are identified by the number of triads.

**Regression**

**Instead of defining a graph by its features and find similarity in that aspect, we look at the** relationships between the different features. We look for exponential connection between two features. Therefore, we fit a linear regression line to every two features (the data observed is on outcome of the feature of all the graphs). In detail, we find such that is **the fitted linear equation**. As can be seen, each pair has. The next step is to choose the pairs that separate well the graphs. After doing so, for each sample we calculate the free coefficient, for all the we chose. Altogether, for each sample we get a vector of free coefficient. This is the vector that defines the position of the sample compared to others.

**Selection of pairs of features - the learning stage**

We now would like to select pairs of features that will set apart best the different graphs given. At this stage, in the context of a pair , each graph is represented as a const , which is defined as . Must be remembered that this step is being done only in the learning stage.  
Let us define matrix, where indicates how well the pair divides well the graphs. We will do that by using coefficient of variation, which is defined as the ratio of the standard deviation to the mean. As mentioned earlier, in this step we know the classifications of each graph. With that in mind, for each network we compute the mean (the center of all its samples) and the standard deviation for all the graphs that are samples of the same network. For every pair of networks we calculate the ratio between the distance of their means and the sum of their standard deviations. Finally, will be the minimum of all the ratios of the pairs. This way, the CVij represents a lower bound of the (i,j) features separation. Therefore, The higher the result is, the better the separation. As a result, we look for the maximum values in the matrix. The features they represent are the features chosen.  
The intuition behind checking the coefficient of variation is that we would like to get circles that contain most of the samples. Moreover, we would like the circles to be further away from each other as possible, and that the radius is the smallest needed to contain most of the graphs of that network. More precisely, we would like to maximize the ratio between the distances of the center of the circles, and their radius’. The center of a circle is the mean of the samples and the radius is the std. the ratio mentioned above is the exactly the coefficient of variables.

##### At this point the learning stage is completed; we can now distinguish, for any given sample, to which network it belongs. Additionally, no more need for previous knowledge.

**Cluster Method-תמונה**

Given a graph, we shall compute the free coefficient vector, as was mentioned before in the section about regression.   
When given two graphs, the distance of the vectors suggests how near the networks are.  
Moreover, when given a numerous of samples, each one is represented as the vector of coefficients, machine learning can be used for the full division.   
In order to do so, we used hierarchical clustering algorithm. Hierarchical clustering groups data over a variety of scales by creating a cluster tree or dendrogram. The tree is not a single set of clusters, but rather a multilevel hierarchy, where clusters at one level are joined as clusters at the next level. This allows you to decide the level or scale of clustering that is most appropriate for your application.

Note that the specific algorithm for the clustering is not critical.

# Results

**Networks**

The networks we tested are from Stanford University. They were generated using email data from a large European research institution. We have anonymized information about all incoming and outgoing email between members of the research institution. The e-mails only represent communication between institution members (the core), and the dataset does not contain incoming messages from or outgoing messages to the rest of the world.

At first, we examined four sub-networks corresponding to the communication between members of four different departments at the institution at different times.

Since there is lack of data in a specific moment in time, we created bins in time, so that each sample is emails exchange in a period of time. Moreover, we analyzed how the size of the bins affects our ability to distinguish between the different graphs. Not only that but we also implemented the temperature model that was mentioned above.

Another network we examined with the temperature model is a signaling pathways and cellular machines in the hippocampal CA1 neuron from different years- 2004 and 2006.

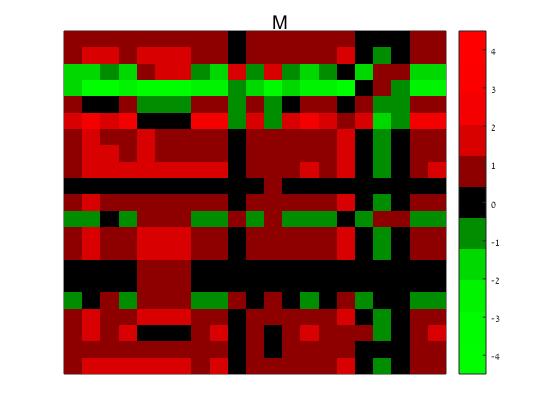
|  |  |  |  |
| --- | --- | --- | --- |
| **Networks examined using Temperature Model** | | | |
| **Name** | **Nodes** | **Edges** | **Description** |
| **Department1** | **309** | **3031** | **All e-mails between members of department at a** large European research institution**.** |
| **Department2** | **162** | **1772** |
| **Department3** | **89** | **1506** |
| **Department4** | **142** | **1375** |
| **Signaling Pathways 2004** |  |  | Signaling pathways and cellular machines in the hippocampal CA1 neuron. |
| **Signaling Pathways 2006** |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Networks **examined** using Bins in Time | | | | |
| **Name** | **Nodes** | **Temporal Edges** | **Time span** | **Description** |
| **Department1** | **309** | ***61046*** | **803 days** | ***All e-mails between members of department at a large European research institution.*** |
| **Department2** | **162** | ***46772*** | **803 days** |
| **Department3** | **89** | ***12216*** | **802 days** |
| **Department4** | **142** | ***48141*** | **803 days** |

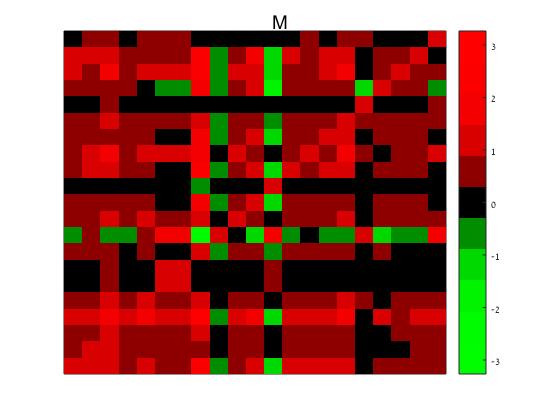
For each network we created samples in time with both the temperature model and by using bins in time. For each sample we calculated all the features as mentioned.

**Regression Results**

We fit a linear regression line to every log of two features (the data observed is all the graphs). In detail, we find m, n such that is the fitted linear equation. As can be seen, each pair has .

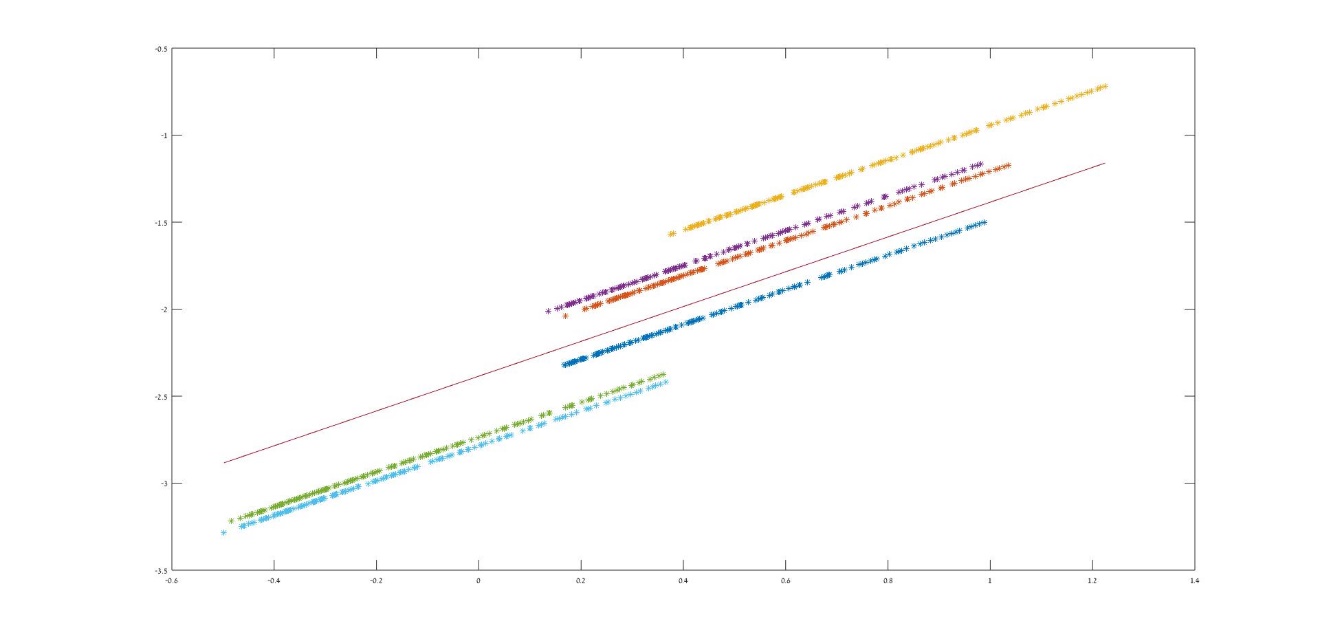


The matrix, as was detailed above (in case of the temperature model).



The matrix in case of bins size 323,000.

For each sample we calculate the free coefficient, for every pair .Then, we obtain a constants vector that includes all the free coefficients of the pairs.



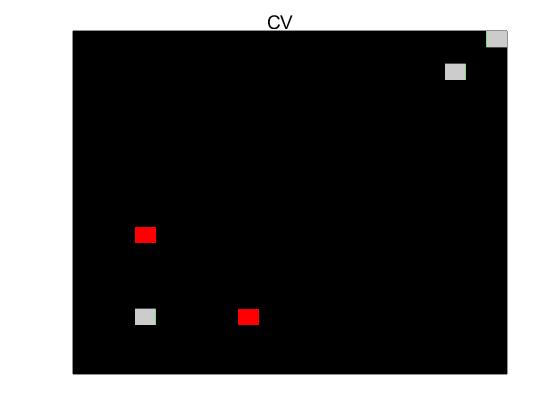
The dots here are (log(value of feature 4), log(value of feature 9)) – each color represents different graphs. The line is the linear function fitted to the dots by linear regression. (in case of the temperature model)

As can be seen, using as samples of the same graphs, we get that are related.

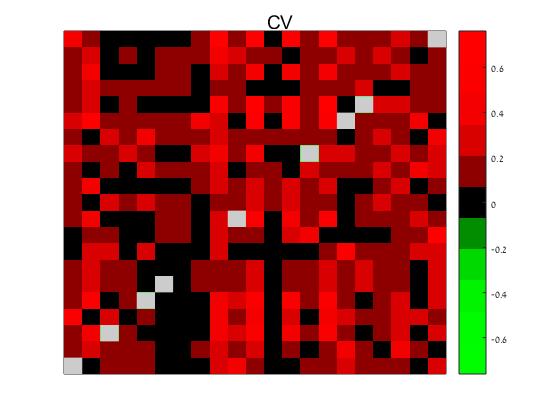
**The features Selection**

After receiving constants vector for pair of features , we would like to choose the pairs that provide maximum separation between the different networks.

The selection of pairs of features, as described in the algorithm section, is being done by CV matrix, which gives an indication of the quality of separation using these pairs.



Here we can see the heatmap of the CV matrix for the networks in the temperature model



Here we can see the heatmap of the CV matrix for the networks using bins (bin's size is 555,000).

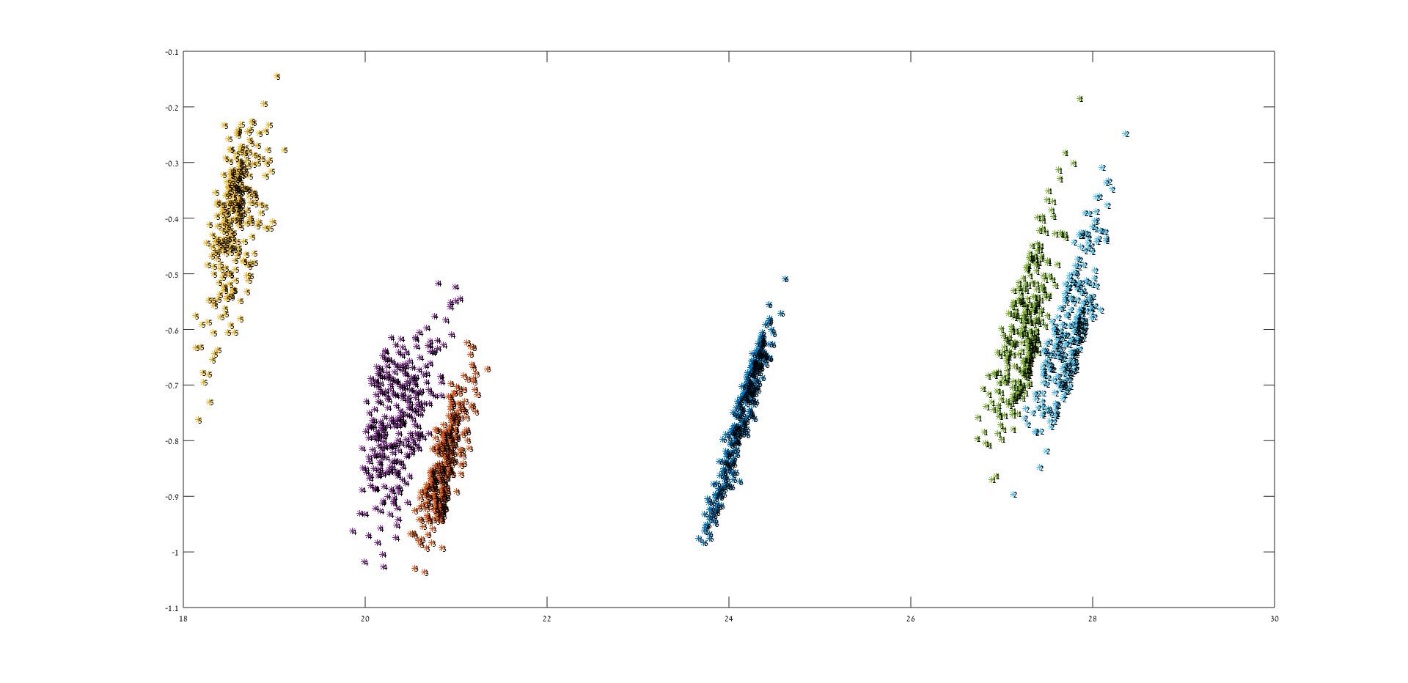
Notice that the highest values of the matrix obtained in places: .

Therefore, in this case, we will choose these pairs of features to achieve good separation of the networks later on.

After choosing the features, the learning stage is complete and we can distinguish between the graphs with no further use of a training set.

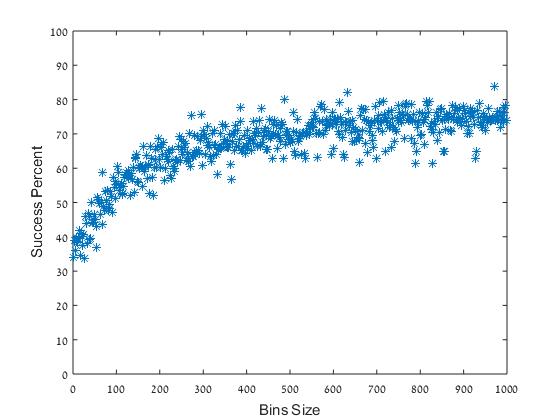
**The Separation**

With the pairs of features we chose before, we got for each sample a vector of constants. This vector defines the position of the sample compared to the others. Note that each sample is related to a network. We would like to divide the samples in groups according to the networks. In order to do that we used machine learning algorithm named Hierarchical clustering algorithm.



In the graph above we can see the PCA 2D casting of the data of samples with the temperature model. Each point represents a sample and each color represents different network. The numbers over the point are the results of the clustering algorithm. Notice that there is 100 percent accuracy in this clustering.

We would like to see how well our algorithm works with the different bin's sizes. We wish to analyze how the size of the bins affects our ability to distinguish between the different graphs. Hence, we calculated the success rates of the proposed method as a function of the bin's size.



An improvement in distingtion is obvious and intuitive. This accures due to the fact that the smaller the bin size is,the less edges there are, the less the network is defined. A graph with more than 200 edges, which is considered to be a small graph can already be separated at most.

Here we can see the final result after PCA. The different colors represent the different graphs. The numbers are the way the hierarchical clustering identifies the separation. This graph has \_\_ edges. We showed here a strong feasibility, that can be improved by choosing better features and better clustering algorithm.

references

[1]H. Bunke, P. Foggia, C. Guidobaldi, M. Vento, Graph clustering using the weighted minimum common supergraph, in: E.R. Hancock, M. Vento (Eds.), Proceedings of the Fourth IARP International Workshop on Graph Based Representations in Pattern Recognition, in: Lecture Notes in Computer Science, vol. 2726, Springer-Verlag GmbH, Berlin, Heidelberg, Germany, 2003

[2]A. Hlaoui, S. Wang, Median graph computation for graph clustering, Soft Computing — A Fusion of Foundations Methodologies and Applications 10 (1) (2006) 47–53.

[3]B. Luo, R.C. Wilson, E.R. Hancock, Spectral feature vectors for graph clustering, in: T. Caelli, A. Amin, R.P. Duin, M. Kamel, D. de Ridder (Eds.), Proceedings of the Joint IARP International Workshops on Syntactical and Structural Pattern Recognition and Statistical Pattern Recognition, in: Lecture Notes in Computer Science, vol. 2396, SpringerVerlag GmbH, Berlin, Heidelberg, Germany, 2002.

[4]B. Luo, R.C. Wilson, E.R. Hancock, Spectral clustering of graphs, in: G. Goos, J. Hartmanis, J. van Leeuwen (Eds.), Proceedings of the Tenth International Conference on Computer Analysis of Images and Patterns, CAIP, in: Lecture Notes in Computer Science, vol. 2756, Springer-Verlag GmbH, Berlin, Heidelberg, Germany, 2003.

[5]A. Robles-Kelly, E.R. Hancock, Graph edit distance from spectral seriation, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (3) (2005) 365–378

[6]S.E. Schaeffer, Algorithms for nonuniform networks, Ph.D. Thesis, Helsinki University of Technology, Espoo, Finland, April 2006

[7]Grover A, Leskovec J. node2vec. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD ’16*. New York, New York, USA: ACM Press; 2016:855-864. doi:10.1145/2939672.2939754.

8. Muchnik L, Itzhack R, Solomon S, Louzoun Y. Self-emergence of knowledge trees: Extraction of the Wikipedia hierarchies. *Phys Rev E - Stat Nonlinear, Soft Matter Phys*. 2007. doi:10.1103/PhysRevE.76.016106.

9. Rosen Y, Louzoun Y. Topological similarity as a proxy to content similarity. *J Complex Networks*. 2014;4(1):38-60. doi:10.1093/comnet/cnv012.

10. Itzhack R, Muchnik L, Erez T, et al. Empirical extraction of mechanisms underlying real world network generation. *Phys A Stat Mech its Appl*. 2010;389(22):5308-5318. doi:10.1016/j.physa.2010.07.011.

[152] J.M. Kleinberg, E. Tardos, Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov random fields, Journal of the ACM 49 (5) (2002) 14–23. [

להעזר במאמרים:

<http://www.leonidzhukov.net/hse/2015/networks/papers/GraphClustering_Schaeffer07.pdf>