Comparative Analysis of Classic Clustering Algorithms and Girvan-Newman Algorithm for Finding Communities in Social Networks

Jelena Ljucović
University "Mediterranean" Podgorica, Montenegro
Tijana Vujičić
University "Mediterranean" Podgorica, Montenegro
Tripo Matijević
University "Mediterranean" Podgorica, Montenegro
Savo Tomović
University of Montenegro Podgorica, Montenegro
Snežana Šćepanović
University "Mediterranean" Podgorica, Montenegro

Abstract
Nowadays finding patterns in large social network datasets is a growing challenge and an important subject of interest. One of current problems in this field is identifying clusters within social networks with large number of nodes. Social network clusters are not necessarily disjoint sets; rather they may overlap and have common nodes, in which case it is more appropriate to designate them as communities. Although many clustering algorithms handle small datasets well, they are usually extremely inefficient on large datasets. This paper shows comparative analysis of frequently used classic graph clustering algorithms and well-known Girvan-Newman algorithm that is used for identification of communities in graphs, which is especially optimized for large datasets. The goal of the paper is to show which of the algorithms give best performances on given dataset. The paper presents real problem of data clustering, algorithms that can be used for its solution, methodology of analysis, results that were achieved and conclusions that were derived.

Keywords: data mining, datasets, clusters, communities, graphs, social networks, ICT, Girvan-Newman algorithm, clustering algorithms
JEL classification: C8

Introduction
There is much information to be gained by analyzing the large-scale data that is derived from social networks. That process is known as social media mining and represents a rapidly growing new field. It is an interdisciplinary field at the crossroad of disparate disciplines, deeply rooted in computer science and social sciences. Finding patterns in large social network datasets is a growing challenge and an important subject of interest. One of current problems in this field is identifying clusters within social networks with large number of nodes (Zafarini et al., 2014). The uniqueness of social media data calls for novel data mining techniques that can effectively handle user generated content with rich social relations. An important question about a social network is how to identify "communities", that is, subsets

containing the nodes (people or other entities that form the network) with unusually strong or numerous connections.
The most common social network relation is "friendship". This type of relation can be found in one of the most popular social networks nowadays – Facebook. However, it is important to accentuate that these social networks, i.e. networks of "friendship", are not the only type and that the term "social network" is much broader than general knowledge assumes. By broader definition, social networks are communities that gather people, or entities on behalf of people, of similar interests or based on common interaction (Leskovec et al., 2014).
Social networks are naturally modeled as graphs, which is sometimes referred to as a social graph. The entities are the nodes, and an edge connects two nodes if the nodes are related by the relationship that characterizes the network (Barabasi, 2016). In this paper accent is put on collaborative social networks, in which the entities are authors of scientific papers and relations are collaboration between two authors that have at least one common paper.

In this paper authors will present comparative analysis of the most used clustering algorithms (standard hierarchical algorithm, k-means algorithm) and well-known Girvan-Newman algorithm (Leskovec et al., 2014). The analysis will be done on real data set clustering, using collaborative social network of University of Montenegro (further: UoM).
In chapter *Methodology* authors give detail description of data within UoM collaborative network, as well as description of used algorithms. The description and graphical representation of results of testing the algorithms are presented in chapter *Results*. In chapter *Discussion* authors give comparison of results between different algorithms. Key parameters to comparison are the number of nodes in the largest cluster, time of execution and algorithm complexity. In chapter *Conclusion* authors will give their view of the analysis and propose future work on this topic.

Methodology

In this paper authors use graph of collaboration network at the UoM as testing dataset. In this network nodes represent scientists from the UoM who have published research papers. There is an edge between two individuals who published one or more papers jointly. The clusters in this network should contain authors working on a particular topic (Ljucović et al., 2016).

In this dataset, authors outside the UoM are not considered, and consequently neither the links between authors from the UoM and authors beyond. Also, only research papers in journals or conferences indexed in SCI, SCIE, SSCI, A&HCI and SCOPUS categories over the period 2000-2015 are considered. These categories are selected because the number of publications in these categories is of great impact on assessing a lecturer career progress at the University of Montenegro. The data source for this study is portal www.nastava.ucg.ac.me, where one can find publications of researchers at the UoM since 1975. The database of the UoM research and scientific papers contains 237 authors, which are presented as 237 nodes of the graph. Mutual cooperation on at least one paper has made a total of 425 pairs of authors, which is represented as 425 edges between nodes of the graph (Ljucović et al., 2016).

An important aspect of social networks is that they contain communities of entities that are connected by strong or numerous edges.Many algorithms for community detection in networks are mainly identical to clustering algorithms in graphs. Clusters are often useful summary of data that is in the form of points in some space. To cluster the points, we need a distance measure on that space. Ideally, points in the

same cluster have small distances between them, while points in different clusters have large distances between them. Social networks, instead of distances, usually use similarity as measure of strength of edge between nodes. In that case, shorter distances relate to more similarity and vice versa. Thus, algorithms and their measures that are intended for graphs in general often have to be somewhat modified in order to give logical results in social networks. There are two main clustering algorithm types: partitional and hierarchical (Maimon et al., 2010).

Partitional clustering algorithms partition the dataset into a set of clusters. In other words, each instance is assigned to a cluster exactly once, and no instance remains unassigned to clusters. K-means (Leskovec et al., 2014) is a well-known example of a partitional algorithm. The algorithm starts with k initial centroids, where k stands for a random, user generated natural number, that is lower than number of nodes. In practice, these centroids are randomly chosen nodes from the dataset. These centroids form the initial set of k clusters. Then, each unassigned node is assigned to one of these clusters based on its distance to the centroid of each cluster. The calculation of distances from nodes to centroids depends on the choice of distance measure. For k-means algorithm weighted graphs were used where weight was calculated as arithmetic mean of percentage of mutual papers in total number of papers of both authors. After assigning all nodes to some of the clusters, the centroids are recomputed by taking the average (mean) of all nodes inside the clusters (hence, the name k-means). This procedure is repeated until convergence, each time using the newly computed centroids. The most common criterion to determine convergence is to check whether centroids are no longer changing. Note that k-means is highly sensitive to the choice of initial k centroids – different clustering results can be obtained on a single dataset depending on that choice.

Previously discussed method considers communities at a single level. In reality, it is common to have hierarchies of communities, in which each community can have sub- or super-communities. Hierarchical clustering deals with this scenario and generates community hierarchies. Initially, n nodes are considered as either 1 or n communities in hierarchical clustering. These communities are gradually merged or split (agglomerative or divisive hierarchical clustering algorithms), depending on the type of algorithm, until the desired number of communities is reached. A dendrogram is a visual demonstration of how communities are merged or split using hierarchical clustering (Zafarani et al., 2014).

The simplest agglomerative hierarchical clustering algorithm is classic hierarchy clustering method (Leskovec et al., 2014), that starts by observing n nodes as n different clusters. In initial moment, clusters contain exactly one node, and it is considered as a clusteroid - the representative node of that cluster, i.e. the node that minimizes the sum of distances between other nodes in the same cluster. During each of the next iterations, the algorithm merges two least distant clusters (based on the distance between clusteroids) and recalculates the new clusteroid of merged cluster. The measure of distance used in this paper for this algorithm is the same as measure of distance used in k-means.

The best-known algorithm for finding communities in social networks that uses divisive hierarchical clustering is Girvan-Newman (further: GN) algorithm (Girvan et al., 2002). It is one of the most widely applied algorithms for social network graph clustering, based on detection of edges that are least likely to fall within the same cluster. To that purpose, the graph edge is denoted a new parameter - "betweenness". GN detects clusters by progressively removing edges from the original network. The connected components (Leskovec et al., 2007) of the remaining graph represent the resulting clusters. Connected component of a graph is a subgraph in which any two

nodes are connected to each other by paths, and which is connected to no additional nodes in the "supergraph".
Instead of trying to construct a measure that tells us which edges are the most central to cluster, GN focuses on edges that are most likely "between" clusters. In that purpose, GN is divided into two main phases - in the first phase it calculates "betweenness" for every edge in graph and in second phase it uses that "betweenness" to cluster the graph.
"Betweenness" of an edge (a, b) is the number of pairs of nodes x and y, such that the edge (a, b) lies on the shortest path between x and y. To be more precise, since there can be several shortest paths between x and y, edge (a, b) is credited with the fraction of those shortest paths that include the edge (a, b). The bigger the number, it suggests that the edge (a, b) runs between two different clusters; that is, a and b do not belong to the same cluster (Girvan et al., 2002).
Finally, GN clusters the graph using the calculated "betweenness". It starts by removing the edges from the graph in order of decreasing "betweenness":it begins with the graph and all its edges, then removes edges with the highest "betweenness", as many times as it is needed, until the graph has broken into a suitable number of connected components.

Results

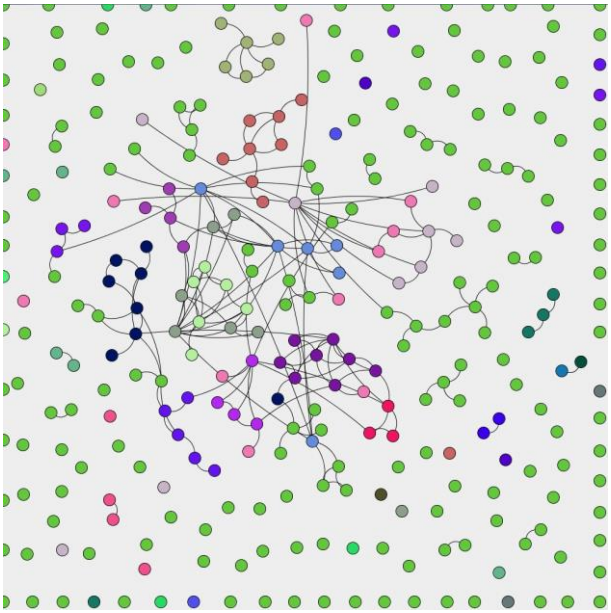
K-means algorithm was implemented in Java programming language using JUNG software library (O'Madadhain, 2016). Number of iterations was set to 500 and convergence threshold to 0.001. Number of clusters was set to 28 in order to get more relevant data for comparison with GN, as it returned 28 as minimal number of clusters for used dataset. Resulting clusters are displayed on figure 1. Nodes in each cluster are marked with different color. Numbers of nodes per cluster are shown in table 1. Average execution time for this algorithm, without visualization included, was 60 milliseconds.

Table 1
Number of Nodes per Cluster Resulting from K-means Algorithm Implementation

Cluster	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Nodes	6	1	6	10	3	4	4	4	12	4	5	4	3	7	1	8	9	169	4	1	1	2	8	2	8	10	3	4

Source: Authors' results

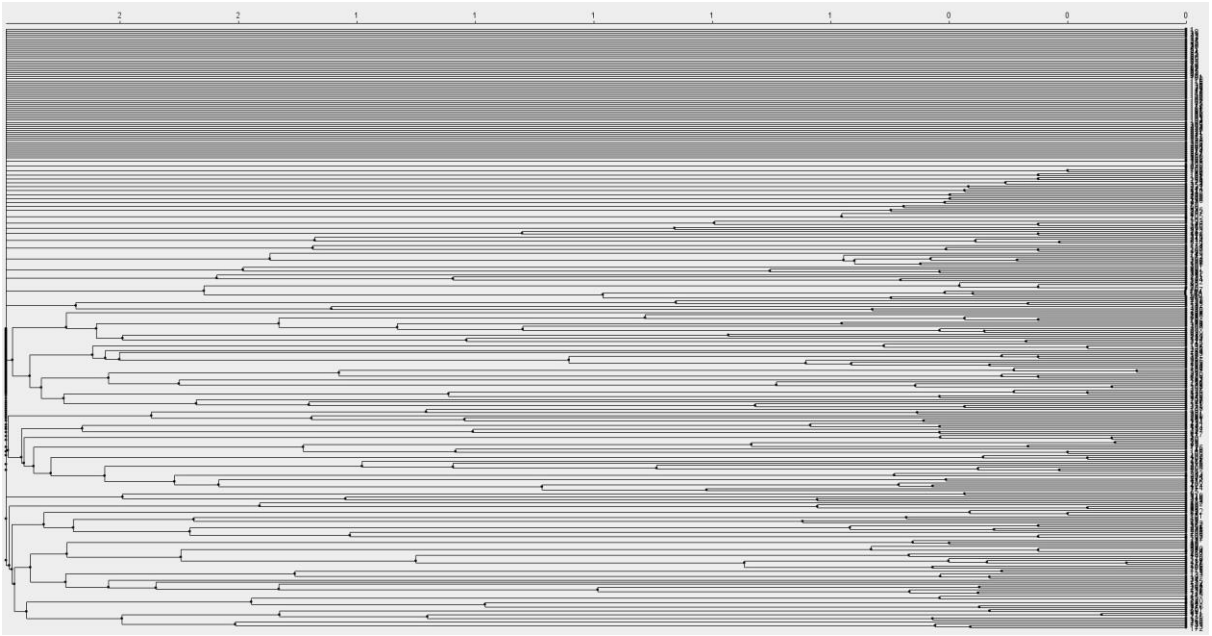
Figure 1
Clusters Resulting from K-means Algorithm Implementation



Source: Authors' illustration

Classic hierarchical clustering algorithm was implemented in Java programming language using source code written by Behnke (2015). Resulting dendrogram is displayed on the figure 2. Average execution time for this algorithm, without visualization included, was 220 milliseconds.

Figure 2
Clusters Resulting from Classic Hierarchical Clustering Algorithm Implementation

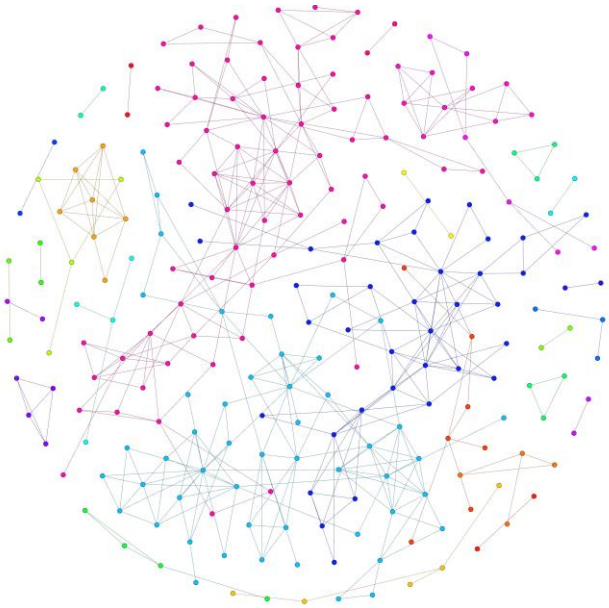


Source: Authors' illustration

Girvan-Newman algorithm was implemented in Gephi open source tool for analysis and visualization of graphs (Gephi Consortium, 2016) using the Girvan-Newman

Clustering plugin for Gephi (Kuchar, 2014). Minimal number of resulting clusters was 28 and visualization for that clustering level is displayed on figure 3. Average execution time for this algorithm to find all possible clustering levels (from 237 to 28) on used dataset is 2900 milliseconds. However, when desired number of clusters is selected afterwards, average execution time per level is 20 milliseconds.

Figure 3
Clusters Resulting from Girvan-Newman Clustering Algorithm Implementation



Source: Authors' illustration

Discussion

Comparison of results between different algorithms is displayed in table 2. Key parameters to comparison are: the number of nodes in the largest cluster (NNL), execution time (ET) andalgorithm complexity (AC).

Table 2
Results of Comparative Analysis

Algorithm	ET [ms]	NNL	AC
K-means	60	169	O(i k n)
Hierarchical clustering	220	N/A	$O(n^3)$
Girvan-Newman	20	64	$O(n^3)$

Note: In algorithm complexity formula n stands for number of nodes, k -number of clusters, i - number of iterations

From above it is easy to conclude that GN has shortest average execution time (per clustering level), but its disadvantage is the fact that it has to find all possible clustering levels first, which lasts longer (2,9 seconds) and only than it is possible to choose desired level of clusterization.
In the theory of social graphs it is known that there exists a continuous phase transition with increasing density of edges in a graph at which a "giant component" forms, i.e. a connected subset of nodes whose size scales extensively. From above, it

can be seen that the giant component exists in both K-means and GN. On the other hand, this parameter is not applicable for hierarchical clustering.
K-means algorithm has the lowest complexity, while GN and hierarchical clustering algorithms have equal complexity because GN is derived from hierarchical.
Although it could be concluded that K-means is the best solution, it suffers from serious shortages: desired number of clusters must be predefined and results largely depend on randomly chosen first centroids.

Conclusion

Standard graph clustering algorithms usually give poor results in social network graphs, because of sheer number of nodes (complexity of the network) and complexity of relations between nodes. Thus, there arises the need to adapt them or to think of new ones and also define new applicable measurement model for each individual scenario. In this paper, while presenting three different types of clustering algorithms, authors had intention to compare their performance on a specific social network graph. From results of the comparative analysis authors can conclude that for observed dataset Girvan-Newman algorithm gave best average performance. For future work it is planned to perform analysis on datasets with larger number of nodes, and to test more clustering algorithms. Also, different scenarios may require adaptation of measurement model. All of these variables will certainly influence performance and end results.

References

1. Barabasi, A. L. (2016), Network Science, Cambridge University Press, Cambridge, UK.
2. Behnke, L. (2015), "Implementation of an agglomerative hierarchical clustering algorithm in Java", available at: https://github.com/lbehnke/hierarchical-clustering-java(April 15, 2016).
3. Gephi Consortium (2016), "GEPHI", available at:https://gephi.org/(March 10, 2016).
4. Girvan, M., Newman, M.E.J. (2002), "Community structure in social and biological networks", in Proceedings of the National Academy of Sciences of the United States of America, pp. 7821-7826.
5. Kuchar, J. (2014), "The Girvan Newman Clustering plugin for Gephi", available at: https://github.com/jaroslav-kuchar/GirmanNewmanClustering(March 10, 2016).
6. Leskovec, J., Kleinberg, J., Faloutsos, C. (2007), "Graph evolution: Densication and shrinking diameters", ACM Transactions on Knowledge Discovery from Data, Vol. 1 No.1.
7. Leskovec, J., Rajaraman, A., Ullman, J. D. (2014), Mining of Massive Datasets, Palo Alto, CA, USA.
8. Ljucović, J., Tomović, S. (2016), "Analyzing clusters in the University of Montenegro collaboration network", in Proceedings of MECO 2016, Budva, Montenegro.
9. Maimon, O., Rokach, L. (2010), Data Mining and Knowledge Discovery Handbook, Springer, USA.
10. O'Madadhain, J. (2016), "JUNG - Java Universal Network/Graph Framework", available at: http://jung.sourceforge.net/ (April 05, 2016).
11. Zafarani, R., Abbasi, M.A., Liu, H. (2014), Social Media Mining: An Introduction, Cambridge University Press.

About the authors

Jelena Ljucović is teaching assistant at the Faculty of Information Technology, University "Mediterranean" Podgorica, Montenegro, and is attending M.Sc. studies at Faculty of Natural Sciences, University of Montenegro. Her research interests include:

database and data warehouse systems, data mining and artificial intelligence. The author can be contacted at **jelena.ljucovic@unimediteran.net**.

Tijana Vujičić, M.Sc. is teaching assistant at the Faculty of Information Technology, University "Mediterranean" Podgorica, Montenegro, and student of Ph.D. studies at Faculty of Organizational Sciences, University of Belgrade, Serbia. Her research interests include: software engineering, programming, intelligent software systems, data mining and technology-enhanced learning. The author can be contacted at **tijana.vujicic@unimediteran.net**.

Tripo Matijević is teaching assistant at the Faculty of Information Technology, University "Mediterranean" Podgorica, Montenegro, where he is attending M.Sc. studies. His research interests include: software engineering, database systems and technology-enhanced learning. The author can be contacted at **tripo.matijevic@unimediteran.net.**

Savo Tomović received his Ph.D. in computer science from University of Montenegro in 2011. He is associated professor at Faculty of Natural Sciences, University of Montenegro. His primary research interest is in the areas of data mining and artificial intelligence. The author can be contacted at **savotom@rc.pmf.ac.me.**

Snežana Šćepanović, Ph.D. is Associate Professor at the Faculty of Information Technology, University "Mediterranean" Podgorica, Montenegro. Within different projects at the University level she is also responsible for development of software for e-learning and online study programs at different levels of study. Her research interests include: system requirement analysis, usability and design of GUI, human computer interaction and e-learning. The author can be contacted at **snezana.scepanovic@unimediteran.net.**