

The stability and robustness of the Github network of collaborators

Jan Kenda^a and Andraž De Luisa^b

^aUniversity of Ljubljana, Faculty of Computer and Information Science, Ljubljana, Slovenia, jk3977@student.uni-lj.si ; ^bUniversity of Ljubljana, Faculty of Computer and Information Science, Ljubljana, Slovenia, ad9366@student.uni-lj.si

This manuscript was compiled on June 8, 2020

Social networks maintain a fragile equilibrium that can be easily disturbed by removing key individuals called linchpins. This is no less true in the network of Github collaborators (1). We propose a measure of robustness for social networks, that how vulnerable a network is when it comes to losing a percentage of its key individuals. The measure is based on the stability of the community structure of the social network, specifically by finding individuals which we deem most important to maintaining the current social structure. We first evaluate the performance of the proposed methods on Lancichinetti–Fortunato–Radicchi benchmark graphs (2) and then on the Github network itself.

networks | communities | robustness | node centrality

Sharing ideas, problems and solutions is a fundamental reason why humans are as advanced as they are today. This is true in all fields of science: biology, physics, mathematics and potentially even more so in computer science. With the advancements in technology and specifically the internet in the last couple of decades, cooperation between researchers and innovators has never been easier. This is in large part due to Git and other version control systems, which allow both researchers and software developers to interact and collaborate on projects with ease, in turn leading to faster learning, innovation and development of scientific and technological advancements.

But this network has a big drawback: it is a social network. And as a social network, it has its linchpins - key members of a society which hold everything together. Without them the network might fall apart. These are the people who introduce others. The keys that connect people into a community. And this is what we are interested in.

If we are able to find a community structure in the GitHub network of collaborators (3), we can then try to measure how "well connected" each community is. Using this measure as a criterion, we can then start looking for linchpins - in this case nodes, which if removed, would destabilize the community structure of the network as much as possible. In the real world social network this means removing individuals that we think contribute most to the development of new ideas and projects. Not necessarily provide solutions themselves, but by connecting others and creating (or just being the core of) communities, contribute to further technological advancements the most.

In this paper we analyse the ability of several node centrality measures to highlight these crucial nodes for the robustness of the community structure in social networks.

Related work

A lot of work has been done in detecting communities and evaluating their stability. A quantification of the community

structure robustness and the method for its computation is proposed by Karrer et al. (4). A somehow similar approach to statistical validation of such partitions of networks into communities has been analysed by Carissimo et al. (5). Both are based on random perturbations of edges in the network, analysing the induced changes in the community structure. Our approach differs from them, as instead on the edges we are more concerned on the importance of single nodes in the communities.

A node centrality measure for classification of nodes with respect to their importance in the community structure, based on the spectrum of the adjacency matrix, has been recently proposed by Wang et al. (6).

Methods and Data

For this project we use the data gathered from SNAP (1), which contains a social network of collaborators on Github up until August 2019. The nodes of the network are Github users who starred popular machine learning and web development repositories (only repositories with at least 10 stars are taken into account). The edges of the network are their follower relationships. It is an undirected graph.

We start by generating Lancichinetti–Fortunato–Radicchi benchmark graphs with planted communities. These graphs serve as testing networks for our project. First we find the communities within the network using various community detection algorithms, such as Louvain (7), Infomap (8) and Label propagation (9).

Since the Github network is a social network, the communities are not as well-defined and clearly separated as is the case with Lancichinetti–Fortunato–Radicchi benchmark graphs. Because of this, we also require algorithms which allow for overlapping communities, such as Demon (10), (11), which uses a local-first approach to community discovery, or if overlaps are significant, a k -clique algorithm based on (12), which works by locating all cliques in the network and then identifying communities based on the analysis of the clique-clique overlap matrix.

The next step is measuring how well-connected and stable a community actually is. The most basic method would be to simply measure the number of connections between nodes within a community and compare it with the number of connections that lead outside of it. A more advanced method is to measure the conductance (13) of communities of the network.

Once we have this, we start looking for key nodes that, if removed, would most disturb the community structure of the network. We do this by employing various methods of node im-

¹ Both authors contributed equally to this work.

portance. We use different methods of node centrality, namely degree centrality, closeness, bridgeness, betweenness, pagerank and eigenvector centrality. We also consider if, instead of removing nodes, we could just remove edges (e.g. sever the collaboration between two users). For this we calculate the edge betweenness centrality measure for all edges.

Using the above measures as criteria, we start testing by removing the nodes (or edges) with the highest measure and again calculating the conductance of communities, or even testing if some communities fall apart due to the removal on an individual. As such we aim to find the best measure, or even a combination of measures, to identify key linchpin individuals, without which the community structure of a social network would change drastically.

Results

Using the aforementioned measure of finding linchpins, we aim to provide a measure of robustness for social networks, with regards to how well they can handle the loss of key individuals. We will test this measure by removing the top 1%, 5%, 10%, 15% and 20% of individuals (based on our selected measure of node importance) and seeing how well the community structure of the graph holds.

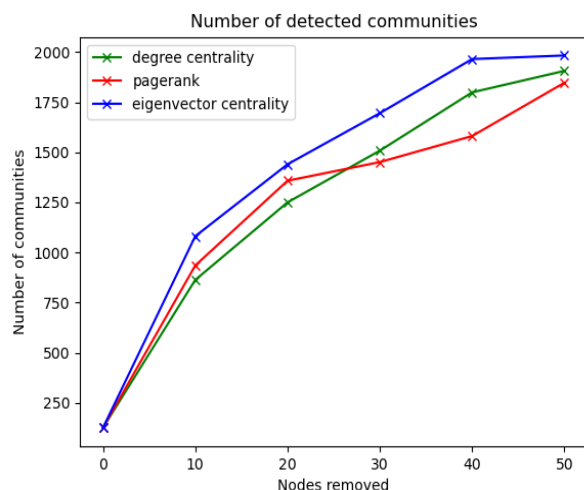


Fig. 1. The number of detected communities based on how many nodes we remove from the network.

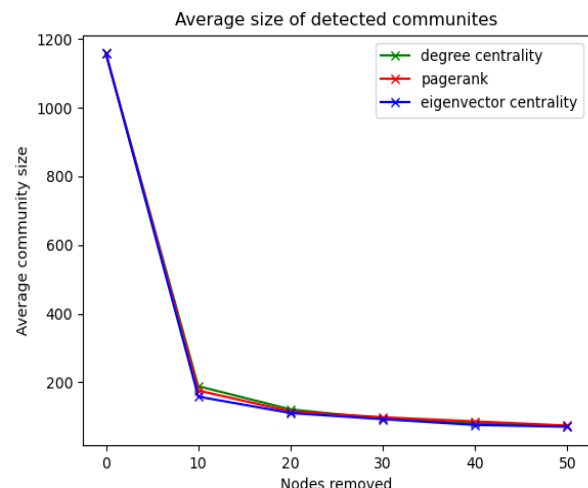


Fig. 2. The average size of detected communities based on how many nodes we remove from the network

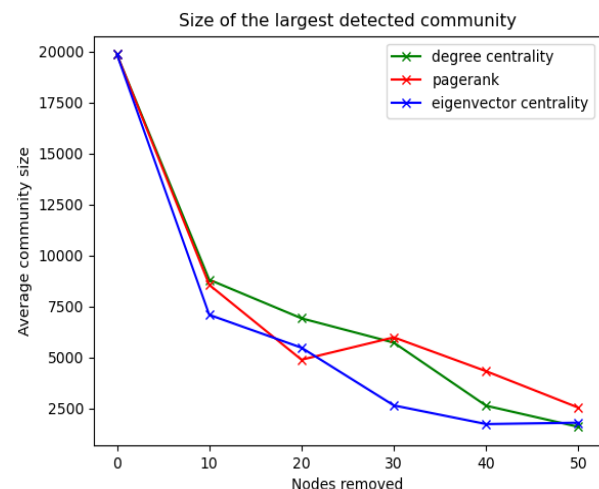


Fig. 3. The size of the largest detected community based on how many nodes we remove from the network

1. Rozemberczki B, Allen C, Sarkar R (2019) Multi-scale attributed node embedding.
2. Lancichinetti A, Fortunato S, Radicchi F (2008) Benchmark graphs for testing community detection algorithms. *Phys. Rev. E* 78(4):046110.
3. Rozemberczki B, Allen C, Sarkar R (2019) Multi-scale attributed node embedding.
4. Karrer B, Levina E, Newman MEJ (2008) Robustness of community structure in networks. *Physical Review E* 77(4).
5. Carissimo A, Cuttillo L, Feis ID (2018) Validation of community robustness.
6. Wang Y, Di Z, Fan Y (2011) Identifying and characterizing nodes important to community structure using the spectrum of the graph. *PLoS one* 6:e27418.
7. Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008(10):P10008.
8. Edler D, Eriksson A, Rosvall M (year?) The mapequation software package.
9. Raghavan UN, Albert R, Kumara S (2007) Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E* 76(3).
10. Coscia M, Rossetti G, Giannotti F, Pedreschi D (2012) Demon: A local-first discovery method for overlapping communities in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12*. (Association for Computing Machinery, New York, NY, USA), p. 615–623.
11. Coscia M, Rossetti G, Giannotti F, Pedreschi D (2014) Uncovering hierarchical and overlapping communities with a local-first approach. *ACM Trans. Knowl. Discov. Data* 9(1).

12. Palla G, Derényi I, Farkas I, Vicsek T (2005) Uncovering the overlapping community structure of complex networks in nature and society. *Nature* 435:814–818.
13. Leskovec J, Lang KJ, Mahoney MW (2010) Empirical comparison of algorithms for network community detection.

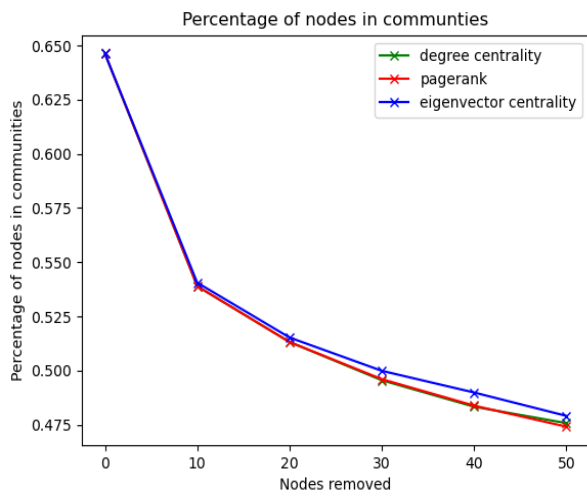


Fig. 4. The percentage of all nodes that are found in communities based on how many nodes we remove from the network.