Graph mining - WS 2016 - Project Report

[First-name-1] [Last-name-1], [First-name-2] [Last-name-2]

Due date: 24 January 2017

Summary

Brief description of the project: used network and the performed analysis.

1 Summary of the data

Report the main characteristics of the dataset (or datasets) that you used. This includes: number of nodes, number of edges, whether directed/undirected graph, static/dynamic graph, number of node/edge labels (if labeled), presence of ground truth (e.g., communities, classes to learn ...), degree distribution.

[Optionally] You can report more fine grained information, such as the number of connected components in the graph, the size of the largest/smallest connected component, the diameter, the number of triangles.

2 Description of the algorithm

Briefly describe in your own words the algorithm or the algorithm(s) you used in terms of input and output, which kind of network it applies to, and what is the main idea behind it (three sentences). Reference the algorithm (e.g., [1])

3 Preprocessing and storing

Report all the preprocessing steps to obtain the final network. How do you store the data (graph database, main memory, disk)? In which format? Did you have to remove some information in order to clean the data?

4 Network analysis

In this section, explain thoroughly the steps of the analysis you performed and give a motivation for such an analysis (e.g. discovering the most influential node, finding interesting patterns, analyzing communities). Elaborate on why you find it challenging and useful. Report possible applications.

4.1 Qualitative Results

Comment the results of the analysis. You should also mention all the relevant details of your analysis, for instance specific values for parameters you used. It is preferable that you show some graphics and present some examples. If you use a visualization method to depict the nodes of the graph, make sure that you can explain at least a part of the network in the plot.

For instance for node classification, show some of the classified vertices. If you have a ground truth, compare it to your results. You can also show a use case by running the algorithm on a smaller network (like the Star Wars network) and show what you find.

For a graph query, show how the algorithm behaves with particular patterns (stars, triangles, cores ...). Is the algorithm producing useful and interpretable results? Are the results unexpected or is it something you can find by yourself? Is there some anomaly?

4.2 Time and scalability

Evaluate the used method regarding its run time and scalability. Report the time you need to preprocess the data (if there is such a phase, like an index construction) and to produce the results. On the same network and algorithm, consider incremental graph sizes (e.g. 10% of the network, 30%, etc.).

Alternatively, generate some random network with `http://www.cse.ust.hk/graphgen/` or any other graph generator and compare the results of the algorithm between the two graphs (the original one and the random one). Are there any important differences in the outcome?

4.3 [Optional] Memory usage

Report how much memory is consumed by the algorithm.

5 Limitations and interesting findings

Did you notice any limitation in the approach? Does it work as expected regarding the quality of the results? In terms of performance, is it too slow when the network is too big? Is it domain dependent or applicable to various types of graphs? To which kind of users are the results of the analysis useful? Describe what you have learned and propose ideas on how the analysis could improved.

Bibliography

[1] P. Zhao, J. X. Yu, and P. S. Yu. Graph indexing: tree + delta >= graph. In *Proceedings of the 33rd international conference on Very large data bases*, pages 938–949. VLDB Endowment, 2007.