Exercise Sheet 12: Centrality Measures
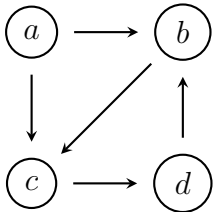Maximilian Marx, Markus Krötzsch
Knowledge Graphs, 2019-01-22, Winter Term 2018/2019

Exercise 12.1. Consider the directed graph G given by the diagram below. Compute the first three partial sums of the Katz centrality values for each vertex of G (use $\alpha = 0.5$):

Exercise 12.2. The Girvan-Newman algorithm for community detection requires that, in each step, the betweenness values for all edges are recalculated. Show that this is necessary to avoid counter-intuitive "communities" in the result: find a small graph for which not re-calculating betweenness after removing an edge results in a substantially different clustering.

Exercise 12.3. Write a program that takes as input

- a directed graph in METIS format

- and a dictionary file in the format of Exercise 11.4 mapping vertex IDs to labels,

and outputs the vertex labels of the given graph, ordered by (decreasing) PageRank.

Modify the program from Exercise 11.4 to only consider events that were part of the 2018 FIFA World Cup (Q19317), and compute the PageRank of this graph. How do you interpret the results?

Hint: You can exploit the fact that most of the entries in the adjacency matrix are 0 by using *sparse matrices* as implemented by, e.g., SciPy.[1] NetworkX[2] offers `pagerank_scipy`,[3] an implementation of PageRank using sparse matrix arithmetic.

Exercise 12.4. Write a program that takes as input

- a directed graph in METIS format

- and a dictionary file in the format of Exercise 11.4 mapping vertex IDs to labels,

and outputs the vertex labels of the given graph, ordered by (decreasing) Betweenness.

Run your program on the Routing Information Base data set[4], a snapshot of the global Internet routing table: every vertex corresponds to an independently operated network (a so-called *Autonomous System*), such as a university, a company, or an internet provider, and an edge from v to w is present if there is some route in which traffic flows from v to w.

What are the twenty most important *Autonomous Systems* in terms of Betweenness?

Hint: NetworKit[5] is a Python library that has an efficient implementation of Betweenness.[6]

---

[1] https://www.scipy.org/

[2] https://networkx.github.io/

[3] https://networkx.github.io/documentation/stable/reference/algorithms/generated/networkx.algorithms.link_analysis.pagerank_alg.pagerank_scipy.html#networkx.algorithms.link_analysis.pagerank_alg.pagerank_scipy

[4] https://github.com/knowsys/Course-Knowledge-Graphs/tree/master/test-data/rib

[5] https://networkit.github.io

[6] https://networkit.github.io/dev-docs/python_api/centrality.html