PageRank and k-core

Maximilien Danisch

Exercise 1 — *PageRank (directed graph)*

Download the Wikipedia dataset from:
`http://cfinder.org/wiki/?n=Main.Data#toc1`.
Download two files: (i) the list of directed hyperlinks and (ii) the name of the pages (corresponding to each node ID).
Implement PageRank using the power iteration method and test your algorithm on the Wikipedia network with $\alpha = 0.15$.
How many iterations seem necessary to reach convergence?
Report the 5 pages with the highest PageRank and the 5 pages with the lowest PageRank.

Exercise 2 — *Correlations*

Draw six scatter plots such that, for each node in the graph:

1. x = PageRank with $\alpha = 0.15$, y = in-degree;
2. x = PageRank with $\alpha = 0.15$, y = out-degree;
3. x = PageRank with $\alpha = 0.15$, y = PageRank with $\alpha = 0.1$;
4. x = PageRank with $\alpha = 0.15$, y = PageRank with $\alpha = 0.2$;
5. x = PageRank with $\alpha = 0.15$, y = PageRank with $\alpha = 0.5$;
6. x = PageRank with $\alpha = 0.15$, y = PageRank with $\alpha = 0.9$.

Should you use linear or log scales?
What can you say about the correlations between the values?

Exercise 3 — *k-core decomposition (undirected graph)*

Implement an efficient algorithm to compute the k-core decomposition (that is to compute a k-core ordering and the core value of each node in the graph).
Test your algorithm on the 5 graphs downloaded in practical 1. For each graph, report the running time of your algorithm as well as the core value of the graph.

Exercise 4 — *Graph mining with k-core*

Download the google scholar dataset at:
`https://drive.google.com/open?id=0B6cGK503Ibt0dXA3Z2lJcHlLX28`.
Download two files: (i) the list of undirected co-authorship links and (ii) the names of the authors (corresponding to each node ID).
Using the google scholar dataset, make a plot similar to the ones shown on slide 11 of the course. Try to find some "anomalous" authors.