CPA. Densest subgraph

Maximilien Danisch, Clémence Magnien, Lionel Tabourier

In this practical, we use the following technical vocabulary.

- The average degree density of a subgraph refers to the value of its average degree divided by 2 (that is its number of edges divided by its number of nodes).
- The size of a subgraph refers to its number of nodes.
- The densest subgraph is the subgraph with maximum average degree density and maximum size.
- The edge density of a subgraph refers to the number of edges divided by the maximum number of edges that could exist between the nodes of that subgraph.

Exercise 1 — *k-core decomposition*

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).
What are the core values of the five following graphs?:

- `http://snap.stanford.edu/data/email-Eu-core.html`
- `http://snap.stanford.edu/data/com-Amazon.html`
- `http://snap.stanford.edu/data/com-LiveJournal.html`
- `http://snap.stanford.edu/data/com-Orkut.html`
- `http://snap.stanford.edu/data/com-Friendster.html`

For each graph, give (i) the average degree density, (ii) the edge density and (iii) the size of a densest core ordering prefix[1]

Exercise 2 — *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.

Exercise 3 — *Densest subgraph*

Make an efficient implementation of the algorithm given in slide 17.
Fix the number of iterations t to 10.
For the four graphs given in Exercise 1, give (i) the average degree density, (ii) the edge density and (iii) the size of a densest prefix for a non-increasing density score[2] ordering. Compare these values to the ones obtained in Exercise 1. Same question for $t = 100$ and $t = 1000$.
Prove that the highest density score is an upper bound of the average degree density of the densest subgraph.
For each of the four graphs, report the highest density score and compare it to the average degree density of a densest prefix obtained in the previous question.

[1] Meaning a subgraph with the highest average degree density among the subgraphs induced on the p first nodes of a core ordering for any p.

[2] the density score of a node refers to the final value associated to the node (cf slide 17)

Exercise 4 — ***(Optional) Graphs not fitting in main memory***

Implement the algorithm of Exercise 3 without storing all edges in main memory, but reading them from disk several times.

Consider the following graph `http://snap.stanford.edu/data/com-Friendster.html` and fix the number of iterations t as large as you can.

Report (i) the value of t, (ii) the average degree density of a densest prefix for a non-increasing density score ordering and (iii) the highest density score.

Exercise 5 — ***(Optional) Triangle densest subgraph***

Make an efficient implementation of the algorithm you guessed in slide 20 leading to "a triangle density score".

Fix the number of iterations t to 100.

For the four graphs given in Exercise 1, give (i) the average degree density, (ii) the edge density and (iii) the size of a triangle densest prefix following a non-increasing triangle density score ordering[3]. Compare these values to the ones obtained in Exercise 3.

[3]Meaning the prefix maximising the fraction between the number of triangles and the number of nodes