Network Analysis: Assignment

Please submit your solutions via duo by 2pm on 24 February 2017 as a zipped archive. Written answers should be in a single pdf document. At the top of the first page of this document identify yourself.

You must reference any sources used (other than the lecture notes). You can use any programming language you choose. Any code you write should be submitted but will not be directly assessed. You may reuse any code in the public domain but should clearly document what you have done. You may reuse code from the exercises done during the course (available on duo) without comment.

Some tasks are for level 4 students only. These are written **in bold**.

The marks available for correct answers to each question are indicated. Partial credit will be given for good attempts.

Your marked work will be returned no later than 17 March.

Question 1 [30 marks]

A *Group Graph* is defined by four parameters m, k, p and q and is constructed as follows:

- Create mk vertices. The vertices are partitioned into m groups each of size k.
- For each pair of vertices that belong to the same group, add an edge between them with probability p.
- For each pair of vertices that belong to different groups, add an edge between them with probability q.

Investigate the degree distribution of Group Graphs for $p + q = 0.5$, $p > q$. Decide which values of m, k, p and q to investigate. You should report on how the structure changes as p and q vary and whether the same effects are found for different values of m and k. Use plots to illustrate your observations. Investigate the relationship between the diameter of Group Graphs and p (for fixed q). **Level 4 students: Investigate the relationship between the clustering coefficient of Group Graphs and p (for fixed q).**

Question 2 [30 marks]

Construct the graph defined in `coauthorship.txt` (also available under datasets on duo). Ignore edge weights.

A k-cycle is a set of k vertices $\{1, 2, \ldots, k\}$ such that 1 is joined to 2, 2 is joined to 3, ..., and k is joined to 1 (it does not matter whether or not other vertices are adjacent). For the graph from `coauthorship.txt`, for $k = 4$ and $k = 5$, find out how many k-cycles each vertex belongs to and plot the distribution (that is, plot number of cycles vs number of vertices that belong to that many cycles).

For each of the following types of graph

- Random Graphs (see Lecture 2),

- PA Graphs (see Lecture 3),
- Group Graphs (see Question 1 above), and
- **Level 4 students: WS Graphs (see Lecture 5)**

create examples with the same number of vertices and approximately the same number of edges as the graph from `coauthorship.txt` and create the same plots. Comment on what you find.

Question 3 [40 marks]

This question is about the problem of searching in graphs. The input to this problem is a *start* vertex and a *target* vertex. The aim is to reach the target vertex from the start vertex by moving along the edges of the graph, but at each step the decision about which vertex to move to next must be made using only local information; that is, it is possible to look at all the neighbours of the current vertex and then decide which one to move to. The *search time* is the number of steps taken (in fact, it is the average over all pairs of start and target vertices). The aim is that the search time is as small as possible. For each of the following types of graph

- Random Graphs (see Lecture 2),
- PA Graphs (see Lecture 3), and
- Group Graphs (see Question 1 above),

describe a strategy for searching in the graphs. That is, describe how, at each step, to decide which vertex to move to next. The information available about vertices is as follows:

- for Random Graphs, each vertex is labelled with a unique integer between 1 and n,
- for PA Graphs, each vertex is labelled with a unique integer between 1 and n, and vertices $1, \ldots, m$ form the complete graph formed initially and then vertices $m+1, \ldots, n$ are added in that order, and
- for Group Graphs, each vertex is assigned a unique integer and the identity of its group.

When deciding which neighbour of the current vertex to move the only information about the neighbours listed above can be used; a search strategy can also record which vertices have been visited already.

You should explain why you believe your strategy might be effective and implement and test it on many instances (you can choose the parameters yourself as long as you are not perverse — for example, the groups in the Group Graph should not be of size 1 or n). Plot search time against the number of instances that achieve that time. Comment on your plots.