

Exercise Sheet 2 – Random Graphs

Max Planck Institute for Software Systems / Saarland University

Instructions

1. You have to submit this assignment (either digitally or on paper) by 14:00 on 13th May 2016.
2. Irrespective of your submission preferences (digital or paper), you have to email the code of the coding assignments to the instructors.
3. To submit your assignments digitally, email your solutions to sma-ss16-instructors@mpi-sws.org, with the subject as “[SMA] HW 2 : <last name 1> <matriculation number 1>, <last name 2> <matriculation number 2> solutions”.
4. Add your full name(s) and the matriculation number(s) in the beginning of all types of assignment solution submissions (in digital copy, on paper, inside the code as comment etc).
5. **You can submit solutions to both section A and section B. Your grade for this exercise will be maximum of the grade for section A and grade for section B).** For example if you score 15 in section A and 0 in section B, your final grade for this exercise will be 15.
6. Instructions for coding assignment - Section A, Problem 2 and 3:
 - (a) You are free to use any programming language you are comfortable with. Here is one library that handles large scale graphs efficiently and provides a large number of functions for analyzing graphs: <http://snap.stanford.edu/>. Its is called SNAP and has two versions : one for C++ and another for Python.
 - (b) You should be able to run your code in a computer with a reasonable configuration (for instance 2GB or more RAM).
 - (c) Please include the results outputted from your code into your solution sheet for these problems (in a table with columns as shown in the problem or as graphs). Additionally, you should also submit your code and instructions on how to run your code in a text file via email to the instructors. For problem 2 The name of the code file should be: **gen.randomgraph.py** (or cpp or some other programming language). Similarly the name of the running instruction file should be: **gen.randomgraph.howtorun.txt**. If there are any additional resources (eg. library) that are required to run your code, please mention those in the how to run file.
 - (d) We should be able to run the submitted code on our machine. If it fails to run on our machine, you will not get any marks for the coding part of the assignment.
 - (e) If any part of your code takes a long time to run (e.g., more than 10 minutes) report that in the instruction file with an estimate of time required.
7. References : Chapters 7 and 12 Networks - An Introduction, Mark Newman

Section A

Note: You can submit the solution for problem 1 digitally (in latex/doc etc.) or even in hard copy (written pen and paper format). For problems 2 and 3 add the textual answers (non-code, e.g., comment on this or that) in the same file/answer sheet.

Problem 1 (10 marks)

Consider the random graph $G(n, p)$ with mean degree c .

1. Show that in the limit of large n the expected number of triangles in the network is $\frac{1}{6}c^3$. Note that, it means, the number of triangles is constant, neither growing nor vanishing in the limit of large n .
2. The clustering coefficient C is given as follows (Eq. 7.41 in Newman's book):

$$C = \frac{3 \times \text{number of triangles}}{\text{number of connected triples}}$$

- (a) Show that the expected number of connected triples in the network is $\frac{1}{2}nc^2$.
- (b) Calculate the clustering coefficient C according to this definition (using the above results), and confirm that it agrees for large n with the value given by $C = \frac{c}{n-1}$.

[4 + [4 + 2]]

Problem 2 (15 marks)

Consider a random graph $G(n, p)$ where giant component covers exactly half of the nodes

1. What is the mean degree of this graph?(Hint : Use the relation $c = -\frac{\ln(1-S)}{S}$, c is mean degree and S is the fraction of nodes in the giant component (Newman's book, section 12.5))
2. Express the number of edges of this random graph as a function of only n .
3. Write and submit a code named **gen.randomgraph.py** (or cpp or some other language) to generate 5 random undirected graphs with the mean degree deduced in 1 and with $n = 100, 1,000, 10,000, 100,000, 1,000,000$ nodes. Also submit a file called **gen.randomgraph.howtorun.txt** to tell us how to run this code.
 - (a) For each random graph, compute the following in your code. :
 - Fraction of nodes in the largest connected component
 - Effective diameter of the random graph
 - Average clustering coefficient

For each random graph, your code should print out: *#nodes: <value>*, *node fraction in largest component:<value>*, *Effective diameter: <value>*, *Average clustering coefficient: <value>*. Also enumerate your code output for your run in **gen.randomgraph.howtorun.txt** (i.e. 5 lines, one for each random graph).

- (b) Compare the above generated values for the random graphs, with the corresponding values generated for the real-world graphs in Assignment 1. Point out any similarities or differences that you observe between the two.
- (c) Comment on whether the output of your code matches with the fact that theoretically the giant component covers half of the nodes.

[3 + 1 + [7.5 + 2.5 + 1]]

Problem 3 (10 marks)

Consider the random graph $G(n, p)$ with mean degree c .

1. Write a code to generate 5 random undirected graphs with the n and p shown in Table 1. Note that the mean degree for all these networks is $(n - 1) \times p$ and have the value of 10.

You don't need to submit your code, but submit the graphs you generated as edgelist. The names of these graphs should be: **randomgraph.meandegree.10.nodes.<numOfNodes>.elist**

Random Graph	n	p
1	101	0.1
2	1001	0.01
3	10001	0.001
4	100001	0.0001
5	1000001	0.00001

Table 1: Values of n and p to generate random graphs.

2. Plot the degree distributions for the above 5 graphs (X-axis degree k , Y-axis the **probability** that a node will have degree k). In each of the figures, also plot the poisson distribution (which is given by $\frac{e^{-c} \cdot c^k}{k!}$, where c is mean degree (take it as 10) and k is the degree of nodes). Submit 5 png figures containing these results named as: **degree.randomgraph.meandegree.10.nodes.<numOfNodes>.png**
 - Hint: Those of you who use snap, note that the degree distribution snap creates by default has the count of nodes in Y axis, you have to divide it by number of nodes in the graph to get the degree distribution we ask.
 - Hint: Again for snap users, note that, snap uses "gnuplot" to generate the graphs, the gnuplot command file is created with extension ".plt" and the data file is created with the extension ".tab" in the same directory. You can use the gnuplot command file with command "gnuplot <.plt file>" to get the ".png" file (of course you need to have gnuplot in your system). Play around with these data/plot files and gnuplot to create the desired figures. You might have to create new data files for plotting the poisson distribution.
3. Comment (based on the your figures) how the degree distributions and poisson distribution compare as the number of nodes (n) in the graphs increase and why?

[2.5 + 5 + 2.5]

Section B

Problem 4 (Research, 35 marks)

Identify a novel and significant problem based on the topics discussed in the class related to Random Graphs. The problem may be about devising some new model, proving a new theorem, identifying a new application or any other idea that you might have. You must first clearly identify the problem you are interested in solving, then motivate why you think this problem is important to answer and non-trivial to solve. Finally, you must propose / sketch a solution that you have in mind to address this problem.

Note that this part of the Exercise is aimed for the students who enjoy research and are motivated to identify and solve problems that remain unaddressed in today's world.

The grading for this section will be highly subjective and will be judged based on the novelty of the identified problem, motivation of the problem and the solution sketch that you provide. If you identify a problem that also interests us, we would be happy to follow it up with you. Additionally, we would invite some students who propose an exceptional problem, to present their work briefly in one of the tutorials.