COMS 6998: Social Networks — Due on Monday, September 22nd, 2014

Assignment #1 – Warming up with Graphs and Randomness

Xiao Zhu, (TAs) — A. Chaintreau (instructor)

Why there is three parts in this assignment: Each part fulfills one of the objectives of the class:

- **Manipulate concepts:** Getting Familiar with the technical concepts used in class, by reproducing similar arguments. Being proficient by manipulating the object to answer some small-size problem.
  You are expected to answer this question rigorously, the answer can be quite short as long as it contains all the required argument to justify your answer.
- **Experience the concepts in real case:** Being able to reproduce these concepts on real or synthetic data. Study their properties in real examples.
- **Connect the concepts to real-life:** Interpret a problem you find in light of the notions you have learned. Develop some critical eye to determine how the concepts introduced are useful in practice.

How to read this assignment : Exercise levels are indicated as follows

($\rightarrowtail$) "elementary": the answer is not strictly speaking obvious, but it fits in a single sentence, and it is an immediate application of results covered in the lectures.
Use them as a checkpoint: it is strongly advised to go back to your notes if the answer to one of these questions does not come to you in a few minutes.

($\curvearrowright$) "intermediary": The answer to this question is not an immediate translation of results covered in class, it can be deduced from them with a reasonable effort.
Use them as a practice: how far are you from the answer? Do you still feel uncomfortable with some of the notions? which part could you complete quickly?

($\nrightarrow$) "tortuous": this question either requires an advanced notion, a proof that is long or inventive, or it is still open.
Use them as an inspiration: can you answer any of them? does it bring you to another problem that you can answer or study further? It is recommended to work on this question only AFTER you are done with the rest!

Part A — Manipulating the Concepts

Exercise 1: Cliques in random graphs (15pt) A k-clique in a graph is a subset of vertexes such that any two of them are directly connected by an edge. Cliques are important objects in many computing problems and they also have simple combinatorial properties that allows to analyze them well. This exercise guides you towards understanding how they appear in a random environment.

In this exercise, we always work with a sequence of undirected uniform random graphs, $G_n = (\{1,\dots,n\}, E_n))_{n\geq 0}$. For each $n \geq 0$ the graph G_n has n vertexes and a random collection of edges such that every edge $e = \{i, j\}$, where $i \neq j$, appears independently from others with a probability $p(n)$.

We wish to find, for any clique size k, a threshold for $p(n)$, defined as a function $t(n)$ such that:

(i) When $\lim_{n\to\infty} \frac{p(n)}{t(n)} = 0$, then $\mathbb{P}\left[G_n \text{ contains a clique of size } k\right] \to_{n\to\infty} 0$.

(ii) When $\lim_{n\to\infty} \frac{p(n)}{t(n)} = \infty$, then $\mathbb{P}\left[G_n \text{ contains a clique of size } k\right] \to_{n\to\infty} 1$.

1. ($\rightarrowtail$) What can you say about cliques of size 1 and 2? Can you define a threshold function for them.

We now consider the case $k = 4$, which means 4 vertexes connected by 6 edges together. It is unfortunately not as easy to characterize directly the probability of having at least one 4 cliques. Hence we need to use the probabilistic second moment method.

We denote by N_n the number of 4-cliques in the graph G_n. Since the edges of G_n appears randomly, N_n is a random variable (with values in $\{0,1,2,\dots\}$). To be more precise, there are $L_n = \binom{n}{4}$ number of choices of 4 elements in the n vertexes of G_n, and any of these can potentially form a clique in G_n, provided that the associated edges happen to be present, which is a probability event.

Let us denote $C_1, C_2, \cdots, C_{L_n}$ all these possible choices of 4 elements and for a subset C_l, let X_l denote the following random variable: $X_l = \begin{cases} 1 & \text{if } C_l \text{ is a clique in } G_n \\ 0 & \text{otherwise} \end{cases}$.

$$\text{We can now rewrite the variables } N_n \text{ as a sum: } \quad N_n = \sum_{l=1,\dots,L_n} X_l \,. \tag{1}$$

2. ($\rightarrowtail$)Are variables $(X_l)_{l=1,\dots,L_n}$ mutually independent?

3. ($\rightarrowtail$)What is the expectation of X_l (for a given l)? What is the expectation of N_n?

4. ($\curvearrowright$)Using Markov's inequality, show that when choosing $t(n) = \frac{1}{n^{2/3}}$ the property (i) of threshold is satisfied.

5. ($\rightarrowtail$)For this value of the threshold and if we assume that $\lim_{n\infty} \frac{p(n)}{(t(n)} = \infty$, what can you say about the expectation of N_n as n goes large? Why is that not sufficient to conclude that the property (ii) is satisfied?

We recall the property of the variance of the sum of 0-1 variable, which implies from Eq.(1) that

$$\text{Var}\,[N_n] \leq \mathbb{E}\,[N_n] + \sum_{l,m|l\neq m} \text{Cov}\,[X_l, X_m]\,; \quad (\text{Cov}\,[X,Y] = \mathbb{E}\,[(X - \mathbb{E}\,[X])(Y - \mathbb{E}\,[Y])])\,.$$

And the inequality $\text{Cov}\,[X_l, X_m] \leq \mathbb{E}\,[X_l X_m]$ true for any non-negative variables.

6. ($\rightarrowtail$)Show that when C_l and C_m are either disjoint or share a single vertex, then X_l, X_m are independent? What can you conclude on $\text{Cov}\,[X_l, X_m]$?

7. ($\curvearrowright$)Assuming that C_l and C_m share exactly two vertexes, give a bound on $\text{Cov}\,[X_l, X_m]$? what if these subsets share exactly three vertexes?

8. ($\curvearrowright$)Using the second moment method, conclude that the property (ii) of threshold is satisfied.

9. ($\curvearrowright$)Without providing too much details, can you explain how you would characterize threshold for $k = 5, 6, \cdots$. Assuming that a concentration argument holds (a variable is most likely around its mean) propose a reasonable candidate for threshold function $t(n)$ for large value of k.

10. ($\nrightarrow$)Imagine we wish to apply the same methods to determine when *chordless cycles* appear in random environment. A cycle is a sequence of edges in the graph forming a path starting and ending in the same node, it is chordless if there does not exist an edge that connect nodes in the cycle except immediate neighbors (i.e. this cycle cannot be made shorter). Would it work?

Part B — Experiencing the Concepts

Instructions: This part includes programming exercises. In this class, you can pick your favorite programming language to solve programming exercises as long as it's runnable on our testing environment. We will test your code on CLIC machines. If you do not have a CS account (thus have no access to CLIC machines) please make sure your code is compatible with the following environment:

```
$ clisp --version
GNU CLISP 2.49 (2010-07-07) (built on allspice.buildd [127.0.1.1])
Software: GNU C 4.6.2
$ gcc --version
gcc (Ubuntu/Linaro 4.6.3-1ubuntu5) 4.6.3
$ java -version
java version "1.6.0_32"
OpenJDK Runtime Environment (IcedTea6 1.13.4) (6b32-1.13.4-4ubuntu0.12.04.2)
OpenJDK 64-Bit Server VM (build 23.25-b01, mixed mode)
$ matlab -r 'ver;exit'
MATLAB Version: 8.0.0.783 (R2012b)
MATLAB License Number: 650045
Operating System: Linux 3.2.0-58-generic #88-Ubuntu SMP Tue Dec 3 17:37:58 UTC 2013 x86_64
Java Version: Java 1.6.0_17-b04 with Sun Microsystems Inc.
Java HotSpot(TM) 64-Bit Server VM mixed mode
$ perl --version
This is perl 5, version 14, subversion 2 (v5.14.2) built for x86_64-linux-gnu-thread-multi
$ python --version
Python 2.7.3
$ python3 --version
Python 3.2.3
$ R --version
R version 2.14.1 (2011-12-22)
Copyright (C) 2011 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: x86_64-pc-linux-gnu (64-bit)
$ scala -version
Scala code runner version 2.11.2 -- Copyright 2002-2013, LAMP/EPFL
```

If you would like to use any other language please check with TAs. You can apply for a CS account at `https://www.cs.columbia.edu/~crf/accounts/cs.html`.

Unless specified otherwise, you're allowed to use any third-party libraries to do graph operations and network analyses, for example, igraph for C/C++/R, NetworkX for Python, JGraphT for Java, and MatlabBGL for Matlab. However the TAs have very limited support on them.

For each programming exercise, put all files of your solution into a folder named exercise[exercise#], e.g. exercise2.1. Include a README file in the folder to tell the TAs how to compile (if needed) and run your code, and what libraries are used for the question. Makefiles are appreciated.

Pack all your folders into a zip file named [UNI]_hw[assignment#].zip, e.g. ab123_hw1.zip when you have finished all questions and submit the zip file to CourseWorks.

For example, you will have a zip file with file structure looks like this,

```
abc_hw1.zip
| exercise2.1
| | solution.c
| | README
| | ...
| exercise2.2
| | solution.c
| | README
| | ...
| ...
```

Exercise 2: Programming warm-up (2pt)

1. ($\rightarrowtail$) Write a program that prints out 42.

2. ($\rightarrowtail$) Write a program that prints out all odd numbers between 50 and 150.

Exercise 3: Graph warm-up (6pt) In this and the next exercise we use a real dataset that shows coauthorship in papers about General Relativity and Quantum Cosmology (sounds complicated!). The nodes in this social network are authors. The nodes are connected by an edge if the authors are both listed as authors on the same paper. You should download this network here: http://snap.stanford.edu/data/ca-GrQc.html. Download the only file under the "files" section, ca-GrQc.txt.gz. The format .txt.gz is a compression format that can be unzipped with 7-Zip. You can get 7-Zip here: http://www.7-zip.org/. You can assume that this input file, ca-GrQc.txt, is in the working directory when we test your code.

Note: You can implement all basic graph algorithms by yourself. But we suggest you get used to a third-party graph library to save your time for this and future exercises.

1. ($\curvearrowright$)Write a program that reads the network from the input file and prints the maximum degree and minimum degree found in the network.

2. ($\curvearrowright$)Write a program that reads the network from the input file and prints out the following numbers:

   - Percentage of authors with only 1 coauthor.
   - Percentage of authors with 10 or fewer coauthors.
   - Percentage of authors with 20 or fewer coauthors.
   - Percentage of authors with 40 or fewer coauthors.
   - Percentage of authors with 80 or fewer coauthors.

Exercise 4: Weak and Strong Ties (7pt) We will define a *strong tie* in the following way: for authors A and B, an edge is a strong tie if A and B share a coauthor. That is, there exists an author C such that A and B both have edges to C (A and B have written a paper with C). An edge that is not a strong tie is a *weak tie*.

1. ($\curvearrowright$)**Strong Tie Function:** Write a program that takes two numbers besides the input file, and prints out a boolean value. The two numbers will be the labels of two nodes. The output should be True if the two nodes have a strong tie, and it should be False if the two nodes don't share an edge or if the two nodes have a weak tie. You can assume the nodes are in the graph.

2. ($\curvearrowright$)**How Strong?** Write a program that reads the collaboration network from the input file and prints, in order, the number of edges, the number of strong ties, and the number of weak ties. It may be helpful to use the code you wrote for the previous question.

3. ($\curvearrowright$)**The Strength of our Weaknesses:** Write a program that reads the collaboration network from the input file, removes all weak ties from the graph (or make a new graph with only the strong ties of the original graph), and prints in order:

   - The number of connected components originally
   - The number of connected components without weak ties
   - The number of nodes in the largest connected component originally.
   - The number of nodes in the largest connected component without weak ties.

Part C — Concepts at large

Exercise 5: Efficiency of social marketing (5 pt)

Imagine you are working for a company that sells online data storage like dropbox, which is cheaper and has better interface than previous products. You product is already well recognized in a small community of early adopters, and now you would like it to get known to the most people. You decide to follow a social recommendation approach:

You start the following promotion: every current active user will be offered 3 invitations that can be forwarded to Facebook friend. When one invitation is accepted by this friend, and she sign up for the product, both users receives 2GB of storage for free. After one week, you see a significant increase of the product adoption through this technique, but it remains moderate. You would like to boost this sale, especially as the competition is quickly getting up to speed.

Your marketing team proposes to you two strategies:

- One called "grass root" in which, in addition, when two invitations get accepted and these users are friends, the three of you receives an additional 500MB free storage, to recognize you are a promoting cluster.
- One called "forward" in which, in the same situation, the 500MB free storage is only given to the originator when two invitations are accepted and these people are not friends in Facebook.

1. (↷) Using concept seen in class, which policy would you suggest to use? How would you predict the ranking between "grass root", "forward" and the baseline strategy?

2. (↷) Your company is also developing an app for online social reviews and recommendations on health, housework and babysitting. You would like your product to be known and plan on offering discount on these products via a similar invitation scheme. Would you provide the same recommendation for which rewarding strategy to use?

NB: You do not need to provide a very long answer, a short paragraph is sufficient, but it is important that your answer clearly describes your argument(s) in English without to someone who would not know the concepts in advance. You are welcome to reuse class readings by citing relevant facts.

Exercise 6: Getting to know your online social environment (5 pt)

Choose a number N between 40 and 80. Look at the last N posts on your Facebook wall. For each post, give a point to the friend who who posted it. For each comment or like, give 0.5 points to the friend who posted it. Now, for each of these friends, look at your number of mutual friends.

NB: We recommend you take N sufficiently high so that you see a diversity of people (including some with more points than others). It should overall take no more than 15mn.

NB: We assumed that you can use a Facebook account. If that's more appropriate you can do the same exercise with RenRen, Google+, Twitter. Please contact us if that creates any difficulty.

1. (↷) Do you see any correlation among these two metrics? Can you interpret it?

2. (↷) Repeat the same experiment but counting points for the last N posts on your newsfeed, and for the last N messages in your Facebook inbox. Can you comment on any common patterns or differences you observe?

Please report these two metrics (in an anonymous manner) in a file. The file has one line per friends, with four columns separated by comma: The first columns contains the number of friends in common, The second columns contains how many points for this friend on the wall The third columns contains how many points for this friend on your newsfeed The second columns contains how many points for this friend on your Facebook inbox

For instance, if one of your friend A has three friends in common with you, 5 post in your wall, 1 comment in your newsfeed and 1 message in your inbox, one line in the file will be:

```
3, 5, 0.5, 1
```