COMS 4995-1: Intro Social Networks — Due on Tuesday, October 23rd, 2012

Assignment #2 – Explaining and Extracting Important Nodes

Chris J. Riederer, Zhikun Ma (TAs) — A. Chaintreau (instructor)

Why there are 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 :** Being able to reproduce these concepts in 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 concepts or principles you have learned. Develop a critical eye to determine how the concepts introduced are useful in practice.

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

($\mapsto$) "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 practice: how far are you from the answer? Do you still feel uncomfortable with some of the concepts and definitions? which part could you complete quickly?

($\not\mapsto$) "tortuous": this question either requires an advanced concept, 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: Exercise 1: Analysis the copying model (8pt+0.5pt for ($\not\mapsto$)) Through the analysis of the Yule process, we have seen in class the consequence of reinforcement. Reinforcement here denotes the fact that a difference between two entities (e.g. the size of two genus, the number of links received by two webpages) is itself biasing the dynamics so that the difference continues to increase. As a consequence, even starting from a small initial set of equivalent entities, minor difference created by randomness could further lead to major differences. In the case of the Yule process, it provided a simple model explaining the imbalance of species among genus which is characterized by a power law.

In this exercice, we conduct a very similar analysis to model edges created in a graph. The main result is to show that a very simple copying strategy leads to big imbalance, characterized by a power law degree distribution of nodes' in-degree.

The copying model We analyze the following dynamics. We start at time $t = 1$ from a directed graph containing $N(1)$ nodes such that each of these nodes has exactly one outgoing edge. We introduce at each time step $t = 2, 3, 4, \ldots$, a new node $v(t)$ with a single outgoing edge $e(t)$ that is initially connected to another node chosen uniformly at random (that we denote by $u(t)$). We then assume the following evolution:

- with probability p, the process stops there, and the new edge connects $v(t)$ to $u(t)$.
- otherwise (hence, with probability $(1-p)$), $v(t)$ examines the edge that is starting at $u(t)$ and decides to *copy* this edge. This means that the edge from $v(t)$ to $u(t)$ is changed to one that goes from $v(t)$ to the destination of the edge starting in $u(t)$.

Evolution of node's degree Since the graph is directed all nodes both have an out-degree and an in-degree. The out-degree of all nodes in the graph remains constant equal to 1. The interesting problem is to analyze the evolution of the in-degree of nodes in the graph as t becomes large.

Let us denote for any $i \geq 0$ by $X_i(t)$ the number of nodes in the graph with an in-degree equal to i.

1. ($\rightarrowtail$) How many nodes (denoted by $N(t)$) and edges (denoted by $E(t)$) are there in the graph as a function of t?

   To simplify we assume that at time $t = 1$ there is a single node and a single edge. Note that it means this edge is a loop from this node to itself.

   Prove that no other loops will be created later. What are now $N(t)$ and $E(t)$?

For the rest of this exercise, we will assume that at time $t = 1$ here is a single node and a single edge. This makes several results below simpler, although a very similar proof applies to the general case.

2. ($\curvearrowright$) Assuming that $X_0(t)$ (*i.e.*, the number of nodes with no incoming edge) is known, what are the possible values of $X_0(t+1)$ and what are the probability that these values occur?

3. ($\curvearrowright$) Derive from the previous question the evolution equation giving the expectation $\mathbb{E}[X_0(t+1)]$ as a function of $\mathbb{E}[X_0(t)]$.

   N.B.: As seen in class, you can either assume that $X_0(t)$ can be treated as a constant equal to its average, or if you have seen conditional expectation in class before, you can prove this statement rigourously.

For the next questions, we assume that $p < 1$.

4. ($\curvearrowright$) Let us introduce, for a given constant c_0, the sequence $\Delta_0(t) = \mathbb{E}[X_0(t)] - c_0 t$. Show that there exists a value of the constant c_0 such that:

$$\forall t \geq 1\,,\ |\Delta_0(t)| \leq |\Delta_0(1)|\ .$$

   What is the value of c_0? (*i*) $c_0 = \frac{1}{1-p}$ (*ii*) $c_0 = \frac{1}{2-p}$ (*iii*) $c_0 = \frac{1}{1+p}$

5. ($\rightarrowtail$) Deduce that the following hypothesis is true for $i = 0$:

$$\forall \varepsilon > 0, \exists A > 0 \ \text{ such that } \ |\Delta_i(t)| \leq A t^{\varepsilon}\ .$$

6. ($\curvearrowright$) For a sequence of constant $c_0, c_1, \ldots$, let us define $\Delta_i(t) = \mathbb{E}[X_i(t)] - c_i t$. Show that for any $i > 0$, if the sequence satisfies $c_i = c_{i-1}\left(1 - \frac{2-p}{(1+p)+i(1-p)}\right)$ then we have:

$$\Delta_i(t+1) = \Delta_i(t)\left(1 - \frac{p + i(1-p)}{N(t)}\right) + \Delta_{i-1}(t)\frac{p + (i-1)(1-p)}{N(t)}\,. \tag{1}$$

7. ($\curvearrowright$) We assume that the hypothesis of question 5 holds for any $i \geq 0$. What does this tells us about the evolution of degree in this system?

   Show that for $i > 0$ we have (remember that we assume $p < 1$):

$$c_i = c_{i-1}(1 - \frac{\beta}{i} + \varepsilon(i)) \;\text{ where }\; |\varepsilon(i)| \leq \frac{A}{i^2} \;\text{ and }\; \beta = \frac{2-p}{1-p}\,.$$

   As a consequence, as shown in the lecture, if we neglect the error term $\varepsilon(i)$ we have that c_i is approximately following a power-law with coefficient β.

   For which values of p does the power law becomes the most imbalanced? Does this correspond to your intuition about the dynamics of copying.

8. ($\curvearrowright$) Assuming now $p = 1$, how could you characterize the decrease of c_i as i gets large? Relate this behavior to the dynamics of the copying model.

9. ($\looparrowright$) Complete the proof by showing that the hypothesis of question 5 is true for all $i \geq 0$.

Exercise 2: Manipulating Importance metrics (6pt)

Imagine you work in a small joint venture whose website contains only 6 webpages about various projects that points to each other as seen in Figure 1. You would like to create a new project and, assuming that

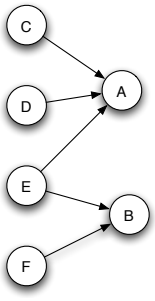

Figure 1: Elementary Network of Hubs and Authorities

Google follows the HITs algorithm, have it placed first whenever the name of your company is searched. We will neglect here the links from webpages of other website (which are longer to obtain, and are more difficult to control). We will also assume that, due to the competitive nature of projects in your company, you cannot ask any project to point to you.

1. Compute the score as obtained by the Hubs and Authorities after two complete iterations in the original network.

2. Now imagine that you create a new project page X. Creating outgoing links from X is not going to help you getting a higher authority score. So you another fake project Y and points to X, as a way to provide some authority to X. You can also decide what project(s) Y links to.

   Compare the score you will obtain after two iterations in the HITs algorithm if Y points only to X or if Y points to A, B and X. Which one will you choose.

3. Imagine you can now create two fake projects Y and Z. Find a configuration that will place your project X as second in the order of importance. Again you will stop after two iterations.

Exercise 3: Issues with Basic Pagerank (0.5pt)

Basic pagerank is the first version of pagerank, without a restart or, equivalently, with no "dumping factor". The importance metric of a node is hence the probability that a random walk in steady state is currently visiting this node. We have seen that this can create black hole whenever a node (or a small group of nodes) only receives links from other, but this does not seem to be the case if the graph is undirected.

1. ($\looparrowright$) Prove that basic pagerank on an undirected graph is not intrinsically better since it is equivalent to another metric we have seen.

2. (no credit, just for fun) Can you imagine a situation in which this algorithm is still useful?

Part B — Experiencing the Concepts

Coding Assignment Submission All programing should be written in one file. The name of this file should be {UNI}_homework2.py, with {UNI} replaced with your UNI. For example, if my UNI is cjr2149, I would name my assignment cjr2149_homework2.py. This file should be uploaded into your Drop Box on Courseworks before the deadline. Please name the file correctly, paying attention to the extension, and do not compress your file before uploading. Points may be subtracted if you do not follow these procedures.

Exercise 4: Coding Problem 1: Pareto In class, we discussed the Pareto principle, an illustration of a power law relationship, which states that 80% of the effects come from 20% of the causes. We'll investigate if that's true in a few data sets. From SNAP (http://snap.stanford.edu/data/index.html), you should download the following files:

- email-EuAll
- web-Stanford
- soc-Epinions1

You should solve these problems in Python using NetworkX.

1. (↣)Write a Python function named problem_one_a. The function should take no inputs are return (not print!) a list. The function should read and analyze the file email-EuAll. The returned list should contain three floats. Each of these floats should be the fraction (not percent!) of nodes that have a certain percentage of the email correspondants. In order, the three floats should be the fractions corresponding to 95%, 90%, and 80% of the email correspondants.

   To calculate "percentage of email correspondants", consider the degree of a node to be the number of that node's correspondants. Thus, the total number of correspondants will be twice the degree of the graph.

2. (↣)Write a Python function named problem_one_b. The function should take no inputs are return (not print!) a list. The function should read and analyze the file web-Stanford. Note that this is a directed graph! The returned list should contain three floats. Each of these floats should be the fraction (not percent!) of nodes that have a certain percentage of the incoming links correspondants. In order, the three floats should be the fractions corresponding to 95%, 90%, and 80% of the incoming links.

3. (↣)Write a Python function named problem_one_c. The function should take no inputs are return (not print!) a list of floats. The function should read and analyze the file soc-Epinions1. Each of these floats should be the fraction (not percent!) of nodes that have a certain percentage of the "trust". In order, the three floats should be the fractions corresponding to 95%, 90%, and 80% of the "trust". Count each outgoing edge as "trust".

Exercise 5: Coding Problem 2: Traceroute and the Fake Power Law We learned in class that what appears to be a power law may actually just be biased data. We will explore that phenomenon in this problem.

1. ($\curvearrowright$)We will create a new graph by running traceroute on one we've already used. Load the graph from web-Stanford using NetworkX. Randomly choose a node from the graph that you will use as your source node. Randomly choose 100 other nodes that are connected to the source node. Create a new graph that consists only of those nodes, and the nodes and edges between them. The function "shortest_path" can be used to find a path between two nodes in a graph.

   After doing the above, on your new graph, return a list similar to the one you did previously. In order, the three floats in the list should be the fractions of nodes that correspond to 95%, 90%, and 80% of the incoming links.

2. ($\curvearrowright$)All work for this problem should be done inside a function named problem_two_b. The function should take no input and return a list of floats, much like previous problems. Inside the function, generate a random graph using NetworkX's fast_gnp_random_graph function. Use values of $n = 70000, p = (\frac{1}{20000})$ and a seed value of 8273696869826982. Return a list similar to the one you created previously. In order, the three floats in the list should be the fractions of nodes that correspond to 95%, 90%, and 80% of the graphs total degree.

3. ($\curvearrowright$)Create the random graph as you did in the previous problem. After creating the graph, pick a random node which you will use as your source node. As you did previously, choose 100 other nodes, and run shortest_path to find a path from your source node to them. Create a new graph made up of these paths. Return a list similar to the one you created previously. In order, the three floats in the list should be the fractions of nodes that correspond to 95%, 90%, and 80% of the graphs total degree.

Part C — Concepts at large

Exercise 6: Effect of Recommendation Engine (5 pt) You are working on a start-up that deals with a large number of user-generated videos. An important part of your website is that users are able to post recommendation for others, in which they "endorse" videos.

Based on your study of some users feedback, you have noticed that users on the website typically prefer videos that receive some endorsement as opposed to none, essentially because it avoids spam. You have also noticed that not so many users are interested by the most endorsed videos, that happen to be hugely popular, because they tend to be somewhat too broad to really be in their interest. You would like to keep this popularity somewhat balanced so that users can enjoy the breadth of your catalog.

One person suggests to introduce a random catalog browsing, in which every user will be shown an endorsement that would be chosen uniformly at random among all endorsements that exist at this time, together with a quick link to endorse this videos as well.

1. Using elements from the lecture, can you predict how the popularity will change as you introduce this feature?

Exercise 7: Big old stars in mathematics (5 pt)

Pick 10 mathematicians arbitrarily who graduated in 2012, using the interface of the Math genealogy project: `http://genealogy.math.ndsu.nodak.edu/search.php` For diversity we recommend you pick ten starting with the same initial as your lastname, and try to avoid picking all of them from institutions in the same country.

Follow their ancestors, by clicking on their advisor (if you see two advisor, pick the first one), and note for each step: (1) Their name, (2) affiliation country (3) their number of students, (4) their number of descendants.

1. Are all ancestor chains disjoint? If not draw the tree on a piece of paper and comment on its topology? Are there important names or countries of affiliations that you observe?

2. Comment on the number of students and ancestors that you observe during this process? Does this seem to correspond to a preferential attachment scheme?