CS224W: Social and Information Network Analysis **Fall 2012**

Problem Set 3

Due 9:30am November 8, 2012

General Instructions

These questions require thought, but do not require long answers. Please be as concise as possible. Please fill the cover sheet (http://cs224w.stanford.edu/cover.pdf) and submit it as a front page with your answers. Include the names of your collaborators (see the course website for the collaboration or honor code policy). You are allowed to take maximum of 1 late day (see course website for a definition of a late day).

Regular (non-SCPD) students should submit hard copies of the answers either in class or in the submission cabinet (see course website for location). You should also include source code and any other files you used.

SCPD students should submit their answers through SCPD. The submission must include all the answers, the source code and the usual SCPD routing form (http://scpd.stanford.edu/generalInformation/pdf/SCPD_HomeworkRouteForm.pdf).

Questions

1 Influence Maximization [25 points – Jacob]

Recall the greedy hill-climbing algorithm for the vertex cover problem discussed in class for directed graphs. In the algorithm, we add nodes to the current set one at a time. At step 0, we have an empty set S_0. At step $i > 0$, we pick the node which is most influential given S_{i-1}, i.e. the node u maximizing $f(S_{i-1} \cup u) - f(S_{i-1})$, where $f(S)$ denotes the number of nodes influenced by S. Then we add that node to S_{i-1} to get a new set S_i. However, the hill climbing algorithm cannot guarantee that its solution is optimal. In other words, there might exist a set T with $|T| = i$ and $f(S_i) < f(T)$. Parts (a) and (b) of this problem ask you to construct some examples where this is the case. Your answer should consist of: (1) a directed graph where an edge from A to B indicates that node A influences node B with probability 1. (2) S_i, the set of nodes that a greedy hill climbing would choose, and (3) T, the optimal set of i nodes.

(a) [8 points] For $i = 2$, construct an example where $f(S_i) < f(T)$.

(b) [8 points] For $i = 3$, construct an example where $f(S_i) \leq 0.8f(T)$

(c) [4 points] Give a property of S_i such that $f(S_i) = f(T)$. By this we mean: what is a general property of influence sets of nodes such that greedy hill climbing always outputs the optimal solution. It must be a property of S_i and the nodes that it influences **only**. Note properties like "the network has only i nodes" are not valid as correct answers.

(d) [5 points] Assume that we stop after k steps and $|S_k| = |T| = k$. Recall that in the class we proved a bound in the form of

$$f(T) \leq f(S_k) + \sum_{i=1}^{k} \delta_i, \tag{1}$$

where $\delta_1, .., \delta_k$ are the largest k values of $\{f(S_k \cup u) - f(S_k)|u\}$ (See the course slides). Construct a family of examples for which $f(S_k) + \sum_{i=1}^{k} \delta_i - f(T)$ can be arbitrarily large.

2 Empirical Power Laws [25 points – Wayne]

In this problem, you will generate a dataset following power-law distribution and test various methods to estimate the α exponent. For convenience, you are free to use any third party library.

(a) [5 points] Recall from class, the probability density function (pdf) of power-law distribution, where $h(x) \propto x^{-\alpha}$, is:

$$P(X = x) = \frac{\alpha - 1}{x_{min}} \left(\frac{x}{x_{min}} \right)^{-\alpha}$$

Please derive the expression of $P(X \geq x)$, the complementary cumulative distribution function (ccdf), in terms of α.

(b) [5 points] Generate a dataset of 100,000 values following a power-law distribution with exponent $\alpha = 2$ and $x_{min} = 1$. Plot on a log-log scale $P(X = x)$. Your plot can be a normalized histogram of the data you generated. To check if you generated your data correctly, you can optionally include the actual probability density function as a line on the same plot.

(c) [3 points] One way to fit a power law distribution to data is to create a histogram of the frequencies of x. On a log-log plot this becomes $ln(h(x)) \propto -\alpha ln(x)$. You can extract the value of α by performing a least-squares linear regression on the log-log histogram data.

Report the value of α you find.

(d) [**2 points**] Similar to (c), now run a least-squares linear regression on the ccdf. Report the value of α you find.

(e) [**5 points**] Run a maximum likelihood estimate of α on the original data. Report the value you find.

(f) [**5 points**] Which method gives the best estimate? Which one the worst?

3 Combination of exponentials for generating power laws [25 points – Bob]

In class we've already seen how the process of preferential attachment can generate power-law distributions. In this question you will explore yet another power-law generating process, called *combination of exponentials.*

Empirical power laws in word-frequency distributions [10 points]

Download the files

- http://www.stanford.edu/class/cs224w/homeworks/hw3/mobydick.txt and
- http://www.stanford.edu/class/cs224w/homeworks/hw3/donquijote.txt,

which contain Herman Melville's *Moby Dick* (in English) and Miguel de Cervantes Saavedra's *Don Quijote* (in Spanish), respectively, with one single word on each line.

(a) [**5 points**] For each of the two files, plot the distribution of word frequencies on a log-log scale. That is, the x-axis should show the number of times a word can appear (i.e., *word frequency*); the y-axis should show the fraction of distinct words with frequency x.

(b) [**5 points**] Using the maximum-likelihood technique of A. Clauset, C.R. Shalizi, and M.E.J. Newman ("Power-law distributions in empirical data," *SIAM Review,* 2009; see lecture notes), fit power-law coefficients α and $x_{\min}$ to both plots (give one pair of coefficients per plot).

What do you conclude about the statistical properties of human language?

Theoretical power laws in monkey novels [15 points]

It might seem surprising that the word-frequency distribution of natural language should follow a power law as strictly as it does, and there has been much scholarly dispute about the processes that make all human writers sample words from exactly the same type of distribution. In this sub-question you will prove that even a randomly typing monkey produces words according to a power-law frequency distribution, and that the power law observed in humans is maybe not that surprising after all.

Assume our monkey hits the space bar with probability q_s and each of m letters with probability $(1 - q_s)/m$. In this setting, we define a *word* as a sequence of letters separated by spaces.

(c) [4 points] How many distinct words of length y are there? Conclude that $p(y)$, the probability of a word having length y, is proportional to e^{ay} (where the word is picked uniformly at random from the set of *distinct* words). Give an expression for a in terms of m.

(d) [5 points] Consider a particular word of length y. Let x be the *relative frequency* of this word (i.e., the probability of the word, followed by a space, in the monkey's output). Derive an expression for x in terms of y, q_s, and m. Conclude that the frequency x of a word is proportional to e^{by}. Give an expression for b in terms of q_s and m.

(e) [6 points] Using parts (c) and (d), derive an expression for $p(x)$, the probability of a word having relative frequency x (where the word is picked uniformly at random from the set of *distinct* words). In particular, show that $p(x)$ is proportional to $x^{-\alpha}$ and give an expression for α in terms of q_s and m. To which value does α converge as m gets large and q_s gets small? Does it converge from above or below? Compare this value to the values you have empirically found in Question 3 (b).

[Hint: Although y can take on only discrete values, you may pretend the function $p(y) = e^{ay}$ you derived in (c) is defined over continuous rather than integral values of y, and that $p(x)$ is also a continuous function in x. This will simplify the analysis, as you can use the fact that $|p(y)\ dy| = |p(x)\ dx|$ if x is an invertible function of y.]

4 Exploring Robustness of Networks [25 Points - Anshul]

In this problem we will examine the relationship between connectivity and robustness in several different networks. Our analysis will be closely related to a Nature publication by Albert, Jeong and Barabasi, *Error and Attack Tolerance of Complex Networks* (2000), which may serve as useful background reading.

While working through this problem, bear in mind that there are a number of ways to formalize the notion of network robustness. Depending on the application, we may be interested

in the mere possibility of reaching other nodes, or the average shortest path length between pairs of nodes, or the worst case shortest path length (i.e., the diameter). For part (a), we will be using the diameter and for part (b), we will be using the fraction of nodes in the largest connected component.

We will use the following three undirected networks for this problem:

- *G(n,m) Random network* - Pick $n = 10670$ nodes and $m = 22002$ edges at random to construct this network.

- *Real World Autonomous System Network* - You can download this network from http://snap.stanford.edu/data/oregon1_010331.txt.gz. It is an undirected network for the internet network with a total of 10670 nodes and 22002 edges.

- *Graph with preferential attachment* - Generate an undirected graph using a variation of the preferential attachment method described in Problem 2. Begin with a complete graph of 40 nodes. At each time step, introduce a new node to the network. Add two edges for the new node in the following fashion - select an edge from the current network uniformly at random and add an edge between the new node and one of the edge's endpoints. Pick the edge's endpoint uniformly at random. Continue doing this till your graph has 10670 nodes. Your graph should ideally have 22040 edges(the actual number may be lesser because of the random number generation but it is expected to be around 22000).

You will be deleting nodes from these networks and computing a given property of the network after each deletion phase. Since it would be computationally too intensive to do this after the deletion of every node, delete nodes in batches of X. First delete X nodes, compute the network property, delete additional X nodes, compute the property again and so on till $Y\%$ of the nodes have been deleted.

You will be considering two node deletion policies in this question. The first, called *Failure*, samples nodes uniformly at random and deletes them from the graph. As the name suggests, this corresponds to a real life scenario where every node is equally probable to break down. The second policy, called *Attack*, corresponds to a scenario where an adversary selectively targetsa specific node. In this case, always delete the nodes in decreasing order of their degree, i.e. highest degree node first. As you will learn from your analysis, the internet network is fairly easy to attack!

(a) [13 Points]

The property you will be computing in this part is the diameter of the graph. Since computing the diameter can be very expensive, you will sample 20 nodes uniformly at random from

the graph, find the shortest path length from these sampled nodes to all other nodes (by doing a BFS for example) and take the maximum value. This logic is already implemented in the *GetBfsFullDiam* function in Snap; you can use that directly.

$$diam = \max_{i \in sampleSet} (\max_{0 \leq j \leq |N|} d(u_i, u_j))$$

For each of the three networks, plot the diameter of the network against the percentage of nodes removed from the network for both the *Failure* and the *Attack* scenario, i.e. your plot should have a total of six lines, two lines each for the three networks, one for the *Failure* scenario and one for *Attack*. Do this for the following two scenarios:

- $X = |N|/100$ and $Y = 50$
- $X = |N|/1000$ and $Y = 2$

where N is the number of nodes in the original network (in this case 10670).

Analyze the *Failure* vs *Attack* scenarios for each of the networks, as well as, comment on the difference (if any) between the behaviour of different networks under the two deletion policies.

(b) [12 Points]

In this part, the property being used is the fraction of nodes which are part of the largest connected component. Plot this fraction versus the percentage of the deleted nodes of all three networks for the two deletion policies in the same plot (again 6 lines in the plot). Use $X = |N|/100$ and $Y = 50$. Comment on the robustness of the networks using the plot.