Large-Scale Data Mining: Models and Algorithms
ECE 232E Spring 2018
Prof. Vwani Roychowdhury
UCLA, Department of ECE

Project 1
Random Graphs and Random Walks

Due on Monday, April 23, 2018 by 11:59 PM

One can use igraph library[1] to generate different networks and measure various properties of a given network. The library has R and Python implementations. You may choose either language that you prefer. However, for this project, using R is strongly recommended, as some functions might not be implemented for the Python version of the package.

Submission: Upload a zip file containing your report and codes to CCLE. One submission from any member of groups is sufficient.

[1] http://igraph.sourceforge.net/

1 Generating Random Networks

1. Create random networks using Erdös-Rényi (ER) model

   (a) Create an undirected random networks with $n = 1000$ nodes, and the probability p for drawing an edge between two arbitrary vertices 0.003, 0.004, 0.01, 0.05, and 0.1. Plot the degree distributions. What distribution is observed? Explain why. Also, report the mean and variance of the degree distributions and compare them to the theoretical values.

   (b) For each p and $n = 1000$, answer the following questions:
   Are all random realizations of the ER network connected? Numerically estimate the probability that a generated network is connected. For one instance of the networks with that p, find the giant connected component (GCC) if not connected. What is the diameter of the GCC?

   (c) It turns out that the normalized GCC size (i.e., the size of the GCC as a fraction of the total network size) is a highly nonlinear function of p, with interesting properties occurring for values where $p = O(\frac{\ln n}{n})$. For $n = 1000$, sweep over values of p in this region and create 100 random networks for each p. Then scatter plot the normalized GCC sizes vs p. Empirically estimate the value of p where a giant connected component starts to emerge (define your criterion of "emergence")? Do they match with theoretical values mentioned or derived in lectures?

   (d) i. Define the average degree of nodes $c = n \times p = 0.5$. Sweep over number of nodes, n, ranging from 100 to 10000. Plot the expected size of the GCC of ER networks with n nodes and edge-formation probabilities $p = c/n$, as a function of n. What trend is observed?

   ii. Repeat the same for $c = 1$.

   iii. Repeat the same for values of $c = 1.1, 1.2, 1.3$, and show the results for these three values in a single plot.

2. Create networks using preferential attachment model

   (a) Create an undirected network with $n = 1000$ nodes, with preferential attachment model, where each new node attaches to $m = 1$ old nodes. Is such a network always connected?

   (b) Use fast greedy method to find the community structure. Measure modularity.

   (c) Try to generate a larger network with 10000 nodes using the same model. Compute modularity. How is it compared to the smaller network's modularity?

   (d) Plot the degree distribution in a log-log scale for both $n = 1000, 10000$, then estimate the slope of the plot.

   (e) You can randomly pick a node i, and then randomly pick a neighbor j of that node. Plot the degree distribution of nodes j that are picked with this process, in the log-log scale. How does this differ from the node degree distribution?

   (f) Estimate the expected degree of a node that is added at time step i for $1 \leq i \leq 1000$. Show the relationship between the age of nodes and their expected degree through an appropriate plot.

   (g) Repeat the previous parts for $m = 2,$ and $m = 5$. Why was modularity for $m = 1$ high?

   (h) Again, generate a preferential attachment network with $n = 1000$, $m = 1$. Take its degree sequence and create a new network with the same degree sequence, through stub-matching procedure. Plot both networks, mark communities on their plots, and measure their modularity. Compare the two procedures for creating random power-law networks.

3. Create a modified preferential attachment model that penalizes the age of a node

   (a) Each time a new vertex is added, it creates m links to old vertices and the probability that an old vertex is cited depends on its degree (preferential attachment) and age. In particular, the

probability that a newly added vertex connects to an old vertex is proportional to:

$$P[i] \sim (ck_i^{\alpha} + a)(dl_i^{\beta} + b),$$

where k_i is the degree of vertex i in the current time step, and l_i is the age of vertex i. Produce such an undirected network with 1000 nodes and parameters $m = 1$, $\alpha = 1, \beta = -1$, and $a = c = d = 1, b = 0$. Plot the degree distribution. What is the power law exponent?

(b) Use fast greedy method to find the community structure. What is the modularity?

2 Random Walk on Networks

1. Random walk on Erdös-Rényi networks

   (a) Create an undirected random network with 1000 nodes, and the probability p for drawing an edge between any pair of nodes equal to 0.01.

   (b) Let a random walker start from a randomly selected node (no teleportation). We use t to denote the number of steps that the walker has taken. Measure the average distance (defined as the shortest path length) $\langle s(t) \rangle$ of the walker from his starting point at step t. Also, measure the standard deviation $\sigma^2(t) = \langle (s(t) - \langle s(t) \rangle)^2 \rangle$ of this distance. Plot $\langle s(t) \rangle$ v.s. t and $\sigma^2(t)$ v.s. t. Here, the average $\langle \cdot \rangle$ is over random choices of the starting nodes.

   (c) Measure the degree distribution of the nodes reached at the end of the random walk. How does it compare to the degree distribution of graph?

   (d) Repeat (b) for undirected random networks with 100 and 10000 nodes. Compare the results and explain qualitatively. Does the diameter of the network play a role?

2. Random walk on networks with fat-tailed degree distribution

   (a) Generate an undirected preferential attachment network with 1000 nodes, where each new node attaches to $m = 1$ old nodes.

   (b) Let a random walker start from a randomly selected node. Measure and plot $\langle s(t) \rangle$ v.s. t and $\sigma^2(t)$ v.s. t.

   (c) Measure the degree distribution of the nodes reached at the end of the random walk on this network. How does it compare with the degree distribution of the graph?

   (d) Repeat (b) for preferential attachment networks with 100 and 10000 nodes, and $m = 1$. Compare the results and explain qualitatively. Does the diameter of the network play a role?

3. PageRank

   The PageRank algorithm, as used by the Google search engine, exploits the linkage structure of the web to compute global "importance" scores that can be used to influence the ranking of search results. Here, we use random walk to simulate PageRank.

   (a) Create a directed random network with 1000 nodes, using the preferential attachment model, where $m = 4$. Note that in this directed model, the out-degree of every node is m, while the in-degrees follow a power law distribution. Measure the probability that the walker visits each node. Is this probability related to the degree of the nodes?

   (b) In all previous questions, we didn't have any teleportation. Now, we use a teleportation probability of $\alpha = 0.15$. By performing random walks on the network created in 3(a), measure the probability that the walker visits each node. Is this probability related to the degree of the node?

4. Personalized PageRank

   While the use of PageRank has proven very effective, the web's rapid growth in size and diversity drives an increasing demand for greater flexibility in ranking. Ideally, each user should be able to define their own notion of importance for each individual query.

   (a) Suppose you have your own notion of importance. Your interest in a node is proportional to the node's PageRank, because you totally rely upon Google to decide which website to visit (assume that these nodes represent websites). Again, use random walk on network generated in part 3 to simulate this personalized PageRank. Here the teleportation probability to each node is proportional to its PageRank (as opposed to the regular PageRank, where at teleportation, the chance of visiting all nodes are the same and equal to $\frac{1}{N}$). Again, let the teleportation probability be equal to $\alpha = 0.15$. Compare the results with 3(b).

(b) Find two nodes in the network with median PageRanks. Repeat part (a) if teleportations land only on those two nodes (with probabilities 1/2, 1/2). How are the PageRank values affected?

(c) More or less, (c) is what happens in the real world, in that a user browsing the web only teleports to a set of trusted web pages. However, this is against the different assumption of normal PageRank, where we assume that people's interest in all nodes are the same. Can you take into account the effect of this self-reinforcement and adjust the PageRank equation?

Final Remarks

The following functions from igraph library are useful for this project:

- `degree, degree.distribution, diameter, vcount, ecount`
- `random.graph.game, barabasi.game, aging.prefatt.game, degree.sequence.game`
- `page_rank`

For part 2 of the project, you can start off with the Jupyter notebook provided to you.