Lecture 11: Networks / Counting on graphs
Modeling Social Data, Spring 2017
Columbia University

Suer Hu

April 13, 2017

1 Network Analysis Briefing

1.1 History

Network analysis has a history starting from 1930. The hot topic today is Contagious effect through social networks.

1.2 Types of Networks

Determining which kind of network it is is often the first step towards a social network analysis, since each network type has its unique properties.

1. Social Networks e.g. Facebook, Twitter networks
2. Information Networks e.g. political blogs (instead of relationship of friends, the interactions may only be referring)
3. Activity Networks e.g. emails
4. Biological Networks e.g. protein interaction clusters
5. Geographical Networks e.g. city roads

1.3 Levels of Networks

1. direction
2. weighted e.g. cost of edge movement, cost of communication
3. metadata e.g. every other data associated

1.4 Which Network

Networks can be very different in terms of what you want to convey with it. We can define how strict we are about the what counts as an edge.
For example, whether to include one-way communication or reciprocated communication in an analysis of facebook relationships; In an organization, whether we care more about the hierarchy or the collaboration of the individuals.

1.5 Data Structure

1. Simple Storage.
   Pro: Simple for storage;
   Con: Complex checking

2. Adjacency Matrix.
   Pro: Easy checking, constant times; When network is directed, matrix is symmetric.
   Con: Matrix is often sparse and hard for storage.

3. Adjacency List.
   Pro: Good for graphing;
   Con: Storage.

1.6 Descriptive Statistics

1. degree

2. path length

3. clustering

4. component

2 Application in R

package: igraph

3 Algorithms for Counting on Networks

3.1 Degree Distribution

group by source nodes
count the number of destination nodes to get source degree
group by degree
count the number of source nodes

3.2 Shortest Path Length

Basic idea: start with source nodes and explore the neighbors.

Algorithm:
init nodes at infinity
    source dist.0
    current boundary
    new boundary is empty
explore non-empty boundaries
loop over all nodes in current boundary
explore each undiscovered neighbors
iterate until no undiscovered neighbors

3.3 Connected Components

init nodes at infinity
pick random unreached nodes
run BFS from that node
    label everything reached as one component
repeat until there is no reachable nodes

3.4 Mutual Friends

Application: Friend recommendation on social media

Algorithm: (need adjacency list)
for each node:
    run over each pairs of its neighbours, increment by 1

This algorithm can get expensive when the adjacency list is long, for instance, celebrity nodes.

3.5 Counting Triangles

(keep adjacency matrix and adjacency list)
for each node:
    run over each pairs of its neighbors, increment by 1
    (check adjacency matrix) if neighbors connected