Lecture 10: Networks

Nicholas Houchois

April 7, 2017

1 History of Networks

1. 1930s - Relationships are first portrayed as a network. These initial networks had to be manually constructed by researchers. Due to the difficulties of collecting data, the datasets and resulting networks were very small.
2. 1960s - Random graph theory develops. Networks can be randomly generated by adding nodes and connecting them with a given probability $p > \frac{((1+\varepsilon)\ln n)}{n}$
3. 1970s - Important research on social networks conducted by Mark Granovetter. He discovers that friends of friends almost always know each other. This results in triangles being formed far more frequently in real social networks than in randomly generated networks. He also studied the relationship between strong and weak ties. Strong ties tend to form triangles whereas weak ties tend to bridge various separate triangles. All of these observations are compared to the base case of a randomly generate graph. Triadic closure makes it hard to mathematically study graphs that represent reality.
4. 1970s - Derek de Solla Price publishes an influential paper on the cumulative advantage effect. According to his research, popularity begets popularity, the rich get richer and this inequality increases over time.
5. 1990s - Duncan Watts and Steven Strogatz develop tshe Watts and Strogatz model to bridge differences between random graphs and real networks. The small world model generates a regular graph and then rewires nodes to other random nodes with a probability of p. This model includes many real-world characteristics such as triadic closure while still allowing researchers to make mathematical statements about the network.
6. 1990s - Researchers begin modeling the internet. Unlike social networks and the small world model, the models of the internet show lots of singular attention.
7. 2000s - Political blogs are highly clustered according to political stance with very few interconnected links. This contributes to increased political polarization.

2 Types of Networks

- Social Networks
  - People saying that they're friends and therefore connected
  - Explicit network
  - E.g. Facebook
- Information Networks
  - Pointing towards information
  - Network of citations
  - Explicit network
  - E.g. The internet

- Activity Networks
  - Shows a connection between two people depending on an activity
  - For example, two people have emailed each other or talked on the phone. It demonstrates a connection, but it is hard to define
  - E.g. Email
- Biological Networks
  - Represent physical processes that are governed by very different forces
  - E.g. Protein interactions
- Geographical Networks
  - Represents physical areas including maps and routing
  - E.g. Road networks

Networks may be a combination of the above types or contain aspects of multiple types. For example, Facebook is a social network but has evolved to include aspects of an information network. By contrast, Twitter is a more pure information network. All of these very different data sources can all be abstracted to the same general thing.

3 Representations of Networks

There are many different levels of abstraction for representing networks and the attributes that you choose to include can drastically change the structure and subsequent interpretation of the network.

- Directed vs undirected graphs. For example, being friends with someone vs following a page on Facebook
- Weighted edges
- Metadata such as node attributes or edge attributes

4 Network Data Structures

- Array of tuples

$$\{[0,1],[0,6],[0,8],[1,4],[1,6],[1,9],[2,4],[2,6],[3,4],[3,5],[3,8],[4,5],[4,9],[7,8],[7,9]\}$$

  - Simple to store.
  - Difficult to compute with.
  - For example, a query such as "Who are the neighbors of node 4" will take $O(E)$ time because there is no index and, consequently, the entire list must be scanned.
- Adjacency matrix
  - Each bit represents a connection between two nodes.
  - Easy to compute linear algebra operations or check edges.
  - Takes $O(C)$ time to lookup a specific connection between nodes.
  - Takes $O(n)$ to check neighbors of a specific node, because you still have to scan down the entire column / row of the node.
  - Symmetric about the diagonal if the graph is undirected.
  - Often stored as a sparse matrix.

	0	1	2	3	4	5	6	7	8	9
0	0	1	0	0	0	0	1	0	1	0
1	1	0	0	0	1	0	1	0	0	1
2	0	0	0	0	1	0	1	0	0	0
3	0	0	0	0	1	1	0	0	1	0
4	0	1	1	1	0	1	0	0	0	1
5	0	0	0	1	1	0	0	0	0	0
6	1	1	1	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	1	1
8	1	0	0	1	0	0	0	1	0	0
9	0	1	0	0	1	0	0	1	0	0

- Adjacency list

$$
\begin{aligned}
0 &\mapsto 1, 6, 8 \\
1 &\mapsto 0, 4, 6, 9 \\
2 &\mapsto 4, 6 \\
3 &\mapsto 4, 5, 8 \\
4 &\mapsto 1, 2, 3, 5, 9 \\
5 &\mapsto 3, 4 \\
6 &\mapsto 0, 1, 2 \\
7 &\mapsto 8, 9 \\
8 &\mapsto 0, 3, 7 \\
9 &\mapsto 1, 4, 7
\end{aligned}
$$

  - Neighbors of a node are stored together.
  - Good for graph traversal.
  - Similar to a sparse matrix

5 Descriptive Statistics

- Degree: how many connections does a node have?
  - Degree distributions
- Path length: how long is the shortest path between two nodes?
  - Breadth first search
- Clustering: How many friends of friends are also friends?
  - Triangle counting
- Components: How many disconnected parts does the network have?
  - Connected components

6 Working with Networks in R

NetworkX in Python is another good library for working with networks

- Calculating Degree:

  - Group by source node
  - Count # destination nodes $\rightarrow$ (source, edge)
  - Group by degree
  - Count # source nodes

- Calculating Path Length:
  - Path.length.hist will plot the path length
  - Easiest to pull out data from here
  - Note: calculating the shortest path for all pairs is computationally expensive and slow