Graphs

- Powerful mathematical abstraction
- Structure consisting of nodes and connections between the nodes (edges)
- $G = (V, E)$
- $|V|$ number of nodes, *order* of the graph
- $|E|$ number of edges, *size* of the graph
- *degree* number of incoming edges to a node

Types of Graphs

- Undirected: each edge relationship is bidirectional
- Directed: Edge points in one direction, $v_1 \rightarrow v_2$
- Weighted, Labeled, coloring: Assignment of values to the edges or nodes of a graph
- *Multi-*: Allows possibility of multiple edges between the same nodes

Representations of Graphs (Data Structures)

- Adjacency Matrix: $|V| \times |V|$ matrix where rows and columns correspond to pairs of vertices, 1 in the (i, j) position of the edge (v_i, v_j) is in the graph, 0 otherwise
- List of the edges: Subset of cartesian product space $V \times V$
- Adjacency list: list-of-lists, $v_i \rightarrow$ (nodes connected to v_i)

Parent and Child Nodes

- Leaf node: node with only one incoming connection
- Child: Node connected to a starting node, starting node is the *parent*
- Child nodes of the same parent are siblings

Searching Graphs

- We might have some reason to *search* through a graph, find a particular node that satisfies some property, find the shortest path to a given node
- Uninformed, systemically traverse nodes until goal is found
- Two basic types of search, *depth first* and *breadth first*
- DFS: start at some node, mark as visited, for each unvisited neighbor apply recursively
- BFS: first visit all of the immediate neighbors before moving on

Applications and Types of Graphs

- Social Networks: Nodes are people and edges are some kind of relationship (friend, contact, etc.)
  - Online or Real World
- Different types of Networks: Electrical Circuits, Gene Regulatory Networks, Hyperlink Relationships of Web Pages
- Theory of Computing: Deterministic and Non-Deterministic Finite Automata

Problems

- Abstraction of the problem of searching spaces, pathfinding
- Spread of a disease, contagion, new idea, innovation, etc.
- *Example Question*: Traverse graph by choosing randomly among neighbors at your present location. On average how often do we visit each node?

Markov Model

- Markov Chain is a probabilistic finite state machine
- The *Markov* property, in general is a kind of *memoryless*
  - Where you are going depends only on where you are now
- **Data**
- Finite number of states
- Probabilities to transition from one state to another, transition matrix $\mathbf{K} = k_{i,j}$
- Like a dynamical system with a random update

Properties of Markov Models

- Irreducibility: any state is reachable from any other state
- Aperiodic: system returns to a given state in irregular intervals
- Homogeneous: Transitions don't depend on step, same for all time
- Existence of stationary distributions
- Think of π as a vector of probabilities on the state space

$$\pi = \mathbf{K}\pi \tag{1}$$

$$\sum_i \pi(i) = 1 \tag{2}$$

Steady State

- If the transition probabilities give an *irreducible* and *aperiodic* Markov chain then the system is guaranteed to converge to its invariant distribution
- Steady State behavior of a Markov chain can be found by analyzing *balance* equations

$$\pi(x^i)K(x^{i-1}|x^i) = \pi(x^{i-1})K(x^i|x^{i-1}) \tag{3}$$

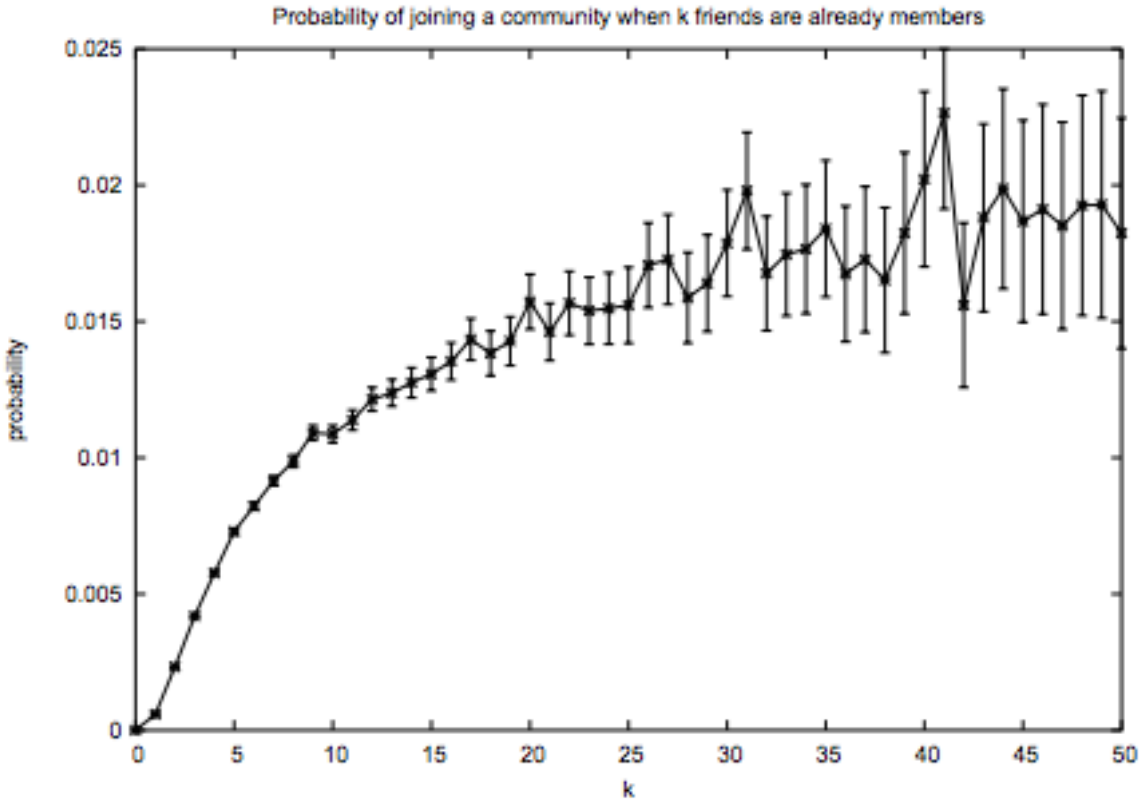
Social Networks

- Graph representing social relationships typically is not static
- Network changes over time: form new friendships, follow or unfollow each other
- Obviously many businesses and government organizations have a vested interest in how this process occurs

Contagion Model

- Much of the essence of these processes can be captured by very simple models
- Probability that someone adopts a behavior is proportional to the number of friends that have already adopted the behavior
- Underlying topology will influence how things diffuse through the network

Spreading of Behavior in Online Communities



- Influence of friends on an individual's behavior

Assignment

- Once you have selected your team members, you should collectively browse through the data sources listed above and choose a topic to study
- Be aware that if you choose to collect your own dataset from an api your team will also have to write a scraper application (be aware of rate limitations etc.)
- Also, depending on the dataset you choose you may be able to simply import a single csv file using (for example) `pandas` or you may need to have a strategy to deal with data in JSON or XML format

networkx package

- Python package for

networkx package in python

- networks is a package for general graph computations

```
import networkx as nx
import matplotlib.pyplot as plt

G = nx.generators.krackhardt_kite_graph()
G.number_of_edges()

nx.draw(G)
plt.show()
```

In-Class Exercise

- Use the `networkx` package commands nx.traversal.dfs_edges() and nx.traversal.bfs_edges() to find the depth first and breadth first traversals of the sample graph from the previous slide
- Plot the resulting tree with order visited