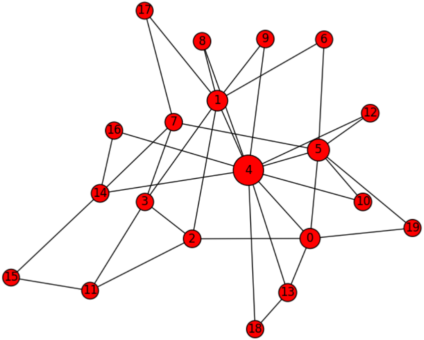
Class Management

- Final exam scheduled ?
- Two remaining class periods
  - Wednesday: Overall summary of the topic, important directions in the field, etc.
  - Friday: Review for the final examination, cover topics since the beginning of the semester, give opportunity to ask questions
- The final report for your project is due this Friday
  - Official due date
  - Will leave the submission box open and will look at latest submission (without penalty)

Visualizing Centrality



- An example where we weight the nodes visually with a value corresponding to their degree

Graph Centrality

- The *centrality* of a node in a graph indicates the importance of a node within its given graph or network
- Multiple centrality measures
- Numerical (scalar) value that gives us some information about how well the node connects the network

Communities

- Clearly, social networks contain different communities
- Groups of friends, groups of people with common interests
- What properties should the network graph have if communities are present?
- How can we identify communities from the network?
- Challenge: individual may belong to multiple communities

Triangle Inequality

$$||\mathbf{x} + \mathbf{y}|| \leq ||\mathbf{x}|| + ||\mathbf{x}|| \tag{1}$$

- Triangle Inequality is a basic notion about how distances work
- Sum of the lengths of two sides should be greater than the length of the remaining side

Top-down and Bottom-up Clustering

- There are two types of hierarchical clustering algorithms: *agglomerative* and *divisive*
- Agglomerative: Begin with all data instances in separate clusters and repeatedly merge clusters until we have a single group (merge two 'nearest' clusters)
- Divisive: Opposite of agglomerative, begin with single large group and split until all clusters are singleton
- We will look agglomerative algorithms today
- From the definition we need a way to measure distances between two clusters (not only between data instances): this is known as the *linkage*

Linkages

- Given two groups of data instances $A = \{x_1, x_2, x_3\}$ and $B = \{x_4, x_5, x_6\}$
- Compare A and B using the distances their elements
- Distance here can be any metric we choose including the euclidean distance
- Goal: some function d that takes clusters (groups) as arguments and returns a number representing the dissimilarity of those clusters, $d(A, B)$

Distances/Clustering in a Graph

- Suppose we want to cluster the nodes in a graph
- To apply clustering we first need to define a distance measure
- Obviously, distance between two nodes should in some way depend upon the presence or absence of edges
- First distance measure: $d(u, v) = 0$ when edge (u, v) exists 1 otherwise
- Second distance measure: $d(u, v) = 0$ when edge (u, v) exists ∞ otherwise
- Do either of these distance measures satisfy the triangle inequality?

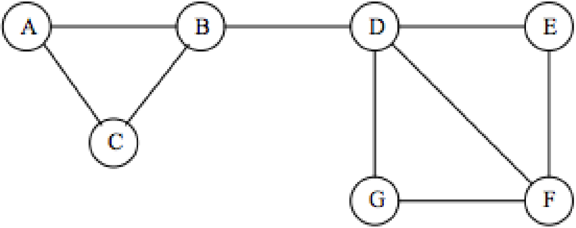
Betweenness of an Edge

- Recall that we investigated a *betweenness centrality* that counted how many shortest paths a node participated in
- We can do the same for *edges*
- Edge Betweenness: Look at the fraction of shortest paths that an edge participates in
- Relevance to community detection: Removing edges with high betweenness, remaining connected components should be communities

Triangles in a Social Network Graph

- A *triangle* is a set of three nodes that are all mutually connected
- Why might we expect there to be more triangles in a social network graph than in a random graph?
- Counting triangles we can guess whether a graph is a social network
- Density of triangles is related to the age of a community

Example Graph



- A sample graph to experiment with.
- Where would you say are the natural communities in this graph?

In-Class Exercise

- Code up the preceding graph in `networkx` and attempt to detect its communities
- Try the following techniques:
- Hierarchical Clustering, (define your own distance matrix)
- Removing edges with high betweenness