WRAP-UP FROM LAST LECTURE ...

Data collection → Data processing → Exploratory analysis & Data viz → Analysis, hypothesis testing, & ML → Insight & Policy Decision
THE VALUE OF A VERTEX
IMPORTANCE OF VERTICES

Not all vertices are equally important

Centrality Analysis:
• Find out the most important node(s) in one network
• Used as a feature in classification, for visualization, etc …

Commonly-used Measures
• Degree Centrality
• Closeness Centrality
• Betweenness Centrality
• Eigenvector Centrality
DEGREE CENTRALITY

The importance of a vertex is determined by the number of vertices adjacent to it

- The larger the degree, the more important the vertex is
- Only a small number of vertex have high degrees in many real-life networks

Degree Centrality:

\[ C_D(v_i) = d_i = \sum_j A_{ij} \]

Normalized Degree Centrality:

\[ C'_D(v_i) = d_i / (n - 1) \]

For vertex 1, degree centrality is 3; Normalized degree centrality is \(3/(9-1)=3/8\).
CLOSENESS CENTRALITY

“Central” vertices are important, as they can reach the whole network more quickly than non-central vertices. Importance measured by how close a vertex is to other vertices.

Average Distance:

\[
D_{avg}(v_i) = \frac{1}{n-1} \sum_{j \neq i}^n g(v_i, v_j)
\]

Closeness Centrality:

\[
C_C(v_i) = \left[ \frac{1}{n-1} \sum_{j \neq i}^n g(v_i, v_j) \right]^{-1} = \frac{n-1}{\sum_{j \neq i}^n g(v_i, v_j)}
\]
CLOSENESS CENTRALITY

Vertex 4 is more central than vertex 3
BETWEENNESS CENTRALITY

Vertex *betweenness* counts the number of shortest paths that pass through one vertex.

Vertices with high betweenness are important in communication and information diffusion.

**Betweenness Centrality:**

\[ C_B(v_i) = \sum_{v_s \neq v_i \neq v_t \in V, s < t} \frac{\sigma_{st}(v_i)}{\sigma_{st}} \]

\( \sigma_{st} \) : The number of shortest paths between s and t

\( \sigma_{st}(v_i) \) : The number of shortest paths between s and t that pass \( v_i \)
BETWEENNESS CENTRALITY

\( \sigma_{st} \): The number of shortest paths between \( s \) and \( t \)

\( \sigma_{st}(v_i) \): The number of shortest paths between \( s \) and \( t \) that pass \( v_i \)

\[ C_B(v_i) = \sum_{v_s \neq v_i \neq v_t \in V, s < t} \frac{\sigma_{st}(v_i)}{\sigma_{st}} \]

What is the betweenness centrality for node 4 ????????????
EIGENVECTOR CENTRALITY

A vertex’s importance is determined by the importance of the friends of that vertex. If one has many important friends, he should be important as well.

\[ C_E(v_i) \propto \sum_{v_j \in N_i} A_{ij} C_E(v_j) \]

\[ x \propto Ax \rightarrow Ax = \lambda x. \]

The centrality corresponds to the top eigenvector of the adjacency matrix A.

A variant of this eigenvector centrality is the PageRank score.
NETWORKX: CENTRALITY

Many other centrality measures implemented for you!


Degree, in-degree, out-degree

Closeness

Betweenness

- Applied to both edges and vertices; hard to compute

Load: similar to betweenness

Eigenvector, Katz (provides additional weight to close neighbors)
STRENGTH OF RELATIONSHIPS
WEAK AND STRONG TIES

In practice, connections are not of the same strength. Interpersonal social networks are composed of strong ties (close friends) and weak ties (acquaintances).

Strong ties and weak ties play different roles for community formation and information diffusion.

Strength of Weak Ties [Granovetter 1973]

- Occasional encounters with distant acquaintances can provide important information about new opportunities for job search.
Social media allows users to connect to each other more easily than ever.

- One user might have thousands of friends online
- Who are the most important ones among your 300 Facebook friends?

Imperative to estimate the strengths of ties for advanced analysis
- Analyze network topology
- Learn from User Profiles and Attributes
- Learn from User Activities
**LEARNING FROM NETWORK TOPOLOGY**

**Bridges** connecting two different communities are weak ties. An edge is a bridge if its removal results in disconnection of its terminal vertices.

Bridge edge(s) ?????

Bridge edge(s) ?????
“SHORTCUT” BRIDGE

Bridges are rare in real-life networks

Idea: relax the definition by checking if the distance between two terminal vertices increases if the edge is removed

• The larger the distance, the weaker the tie is

Example:

• \( d(2,5) = 4 \) if \((2,5)\) is removed
• \( d(5,6) = 2 \) if \((5,6)\) is removed
• \((5,6)\) is a stronger tie than \((2,5)\)
NEIGHBORHOOD OVERLAP

Tie strength can be measured based on neighborhood overlap; the larger the overlap, the stronger the tie is.

\[
\text{overlap}(v_i, v_j) = \frac{\text{number of shared friends of both } v_i \text{ and } v_j}{\text{number of friends who are adjacent to at least } v_i \text{ or } v_j} = \frac{|N_i \cap N_j|}{|N_i \cup N_j|} - 2.
\]

Example:

\[
\text{overlap}(2, 5) = 0,
\]

\[
\text{overlap}(5, 6) = \frac{|\{4\}|}{|\{2, 4, 5, 6, 7, 10\}| - 2} = 1/4.
\]
LEARNING FROM PROFILES AND INTERACTIONS

Twitter: one can follow others without followee’s confirmation
• The real friendship network is determined by the frequency two users talk to each other, rather than the follower-followee network
• The real friendship network is more influential in driving Twitter usage

Strengths of ties can be predicted accurately based on various information from Facebook
• Friend-initiated posts, message exchanged in wall post, number of mutual friends, etc.

Learning numeric link strength by maximum likelihood estimation
• User profile similarity determines the strength
• Link strength in turn determines user interaction
• Maximize the likelihood based on observed profiles and interactions
COMMUNITY DETECTION

A co-authorship network of physicists and mathematicians (Courtesy: Easley & Kleinberg)
WHAT IS A COMMUNITY?

Informally: “tightly-knit region” of the network.
• How do we identify this region?
• How do we separate tightly-knit regions from each other?

It depends on the definition of **tightly knit**.
• Regions can be nested
• Examples ?????????
• How do bridges fit into this ?????????
WHAT IS A COMMUNITY?

An example of a nested structure of the communities

(Courtesy: Easley & Kleinberg)

Removal of a bridge separates the graph into disjoint components
COMMUNITY DETECTION

Girvan-Newman Method

• Remove the edges of highest betweenness first.
• Repeat the same step with the remainder graph.
• Continue this until the graph breaks down into individual nodes.

As the graph breaks down into pieces, the tightly knit community structure is exposed.
Results in a hierarchical partitioning of the graph
GIRVAN-NEWMAN METHOD

Betweenness(7-8) = 7*7 = 49
Betweenness(1-3) = 1*12 = 12

Betweenness(3-7) = Betweenness(6-7) =
Betweenness(8-9) = Betweenness(8-12) = 3*11 = 33
Betweenness(1-3) = 1*5 = 5
Betweenness(3-7) = Betweenness(6-7) =
Betweenness(8-9) = Betweenness(8-12) = 3*4 = 12

(a) Step 1
GIRVAN-NEWMAN METHOD

(b) Step 2

Betweenness of every edge = 1
G=nx.Graph()

# Returns an iterator over partitions at
# different hierarchy levels
nx.girvan_newman(G)
NETWORKX: VIZ

Can render via Matplotlib or GraphViz

```python
import matplotlib.pyplot as plt

G = nx.Graph()
nx.draw(G, with_labels=True)

# Save to a PDF
plt.savefig("my_filename.pdf")
```

- [https://networkx.github.io/documentation/development/gallery.html](https://networkx.github.io/documentation/development/gallery.html)
# Cycle with 24 vertices
G=nx.cycle_graph(24)

# Compute force-based layout
pos=nx.spring_layout(G, iterations=200)

# Draw the graph
nx.draw(G,pos,
       node_color=range(24),
       node_size=800,
       cmap=plt.cm.Blues)

# Save as PNG, then display
plt.savefig("graph.png")
plt.show()
# Branch factor 3, depth 5
G = nx.balanced_tree(3, 5)

# Circular layout
pos = graphviz_layout(G,
    prog='twopi', args='')

# Draw 8x8 figure
plt.figure(figsize=(8, 8))
nx.draw(G, pos,
    node_size=20,
    alpha=0.5,
    node_color="blue",
    with_labels=False)

plt.axis('equal')
plt.show()