COMP4222 Machine Learning with Structured Data

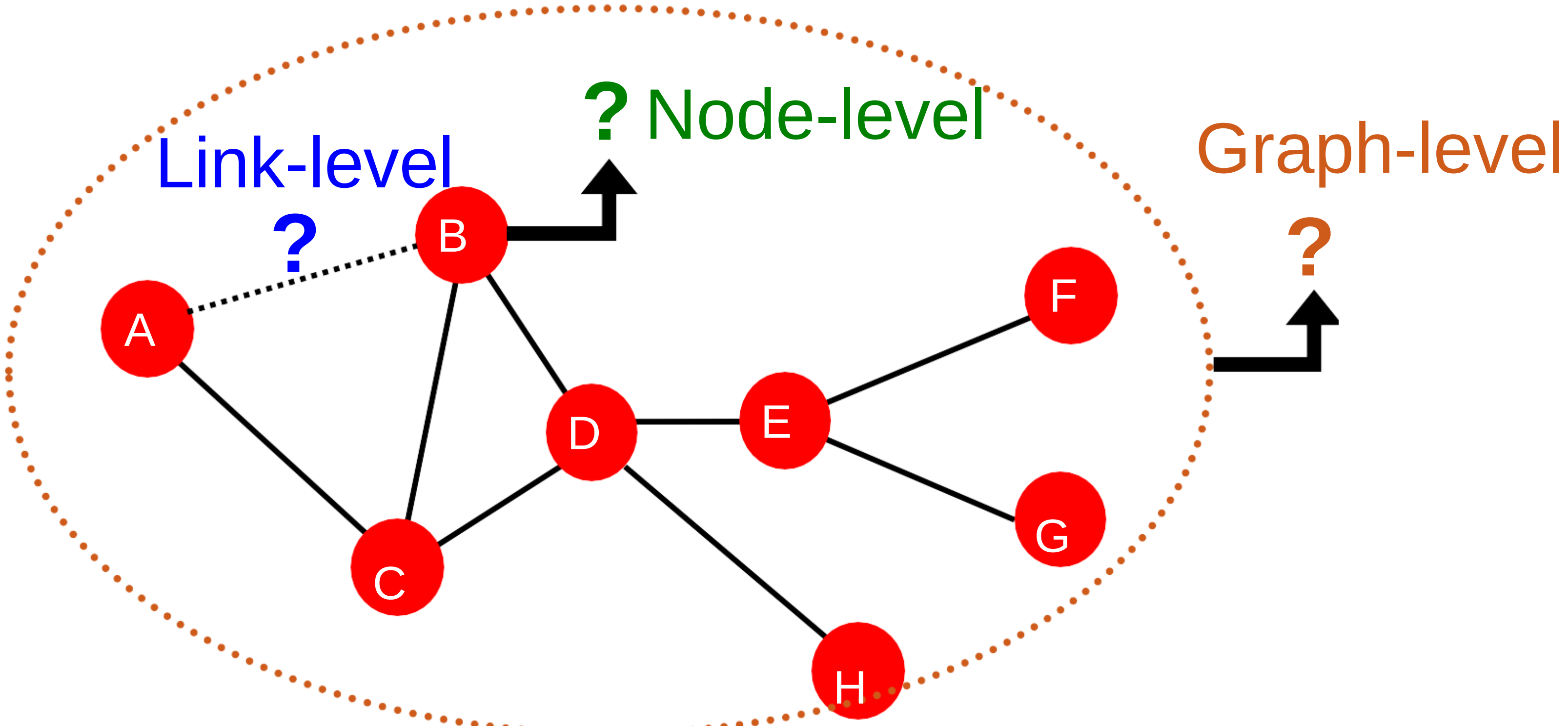
Traditional Machine Learning Methods

Yangqiu Song

Slides credits: Jure Leskovec @Stanford, Lada Adamic @Facebook, and James Moody @Duke

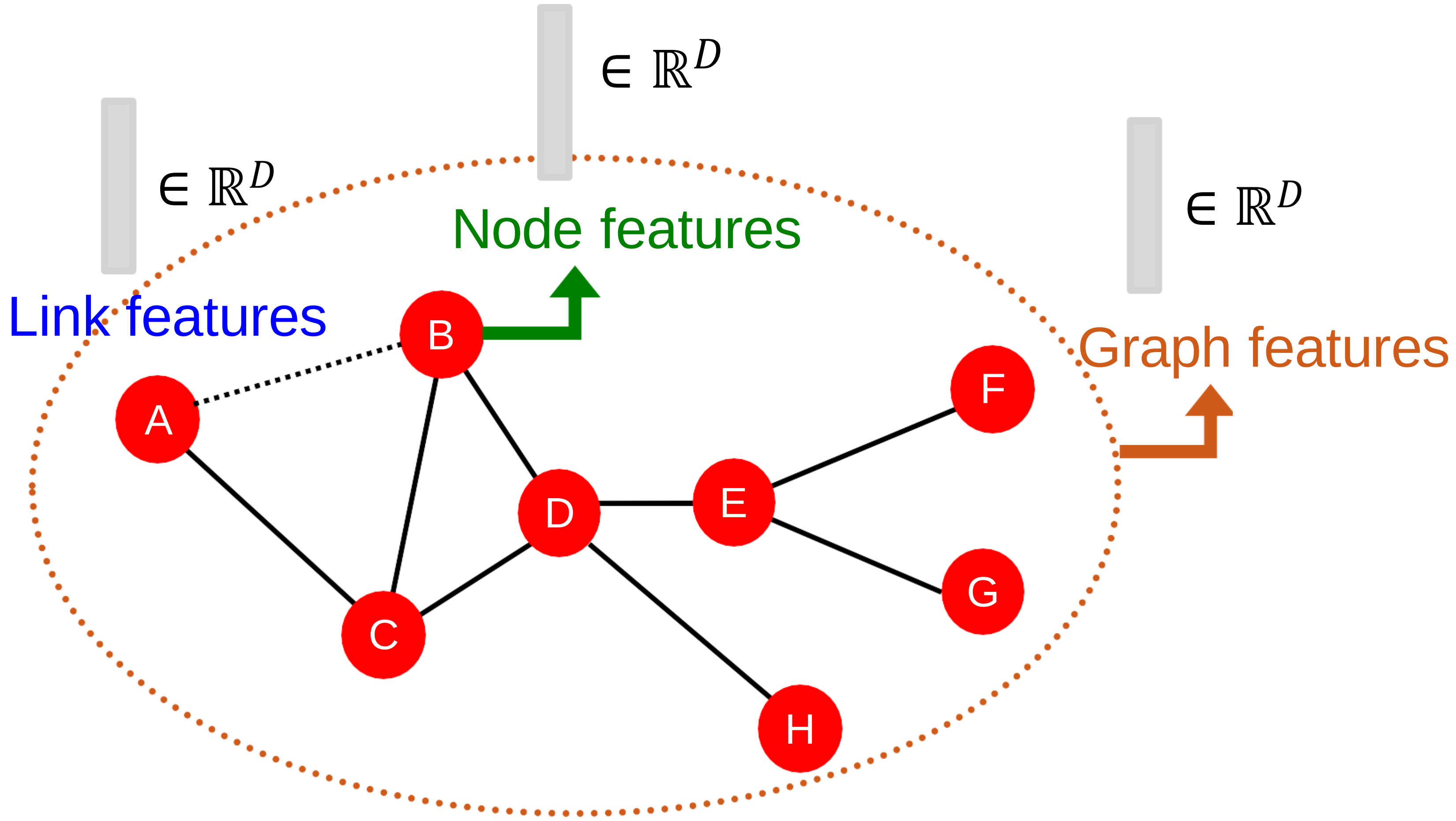
Machine Learning Tasks: Review

- Node-level prediction
- Link-level prediction
- Graph-level prediction



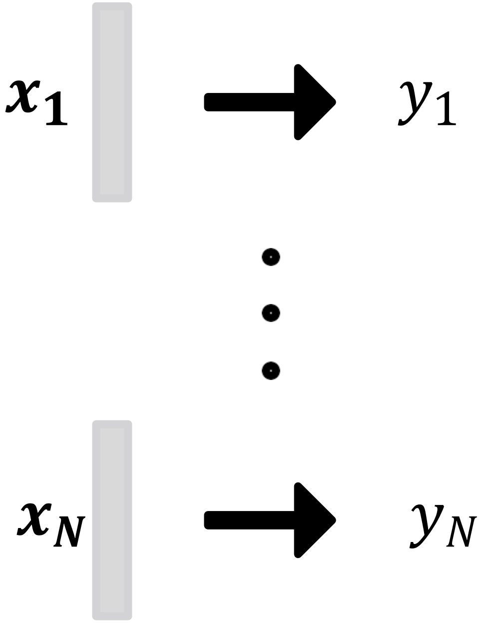
Traditional ML Pipeline

- Design features for nodes/links/graphs
- Obtain features for all training data

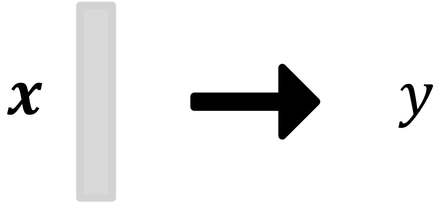


Traditional ML Pipeline

- **Train an ML model:**
  - Random forest
  - SVM
  - Neural network, etc.



- **Apply the model:**
  - Given a new node/link/graph, obtain its features and make a prediction



This Lecture: Feature Design

- Using effective features over graphs is the key to achieving good model performance.
- Traditional ML pipeline uses hand-designed features.
- For simplicity, we focus on undirected graphs.

Machine Learning with Graphs

- Goal: Make predictions for a set ofobjects

- Design choices:
  - **Features:** *d*-dimensional vectors
  - **Objects:** Nodes, edges, sets of nodes, entire graphs
  - **Objective function:**
    - What task are we aiming to solve?

Machine Learning with Graphs

- Example: Node-level prediction

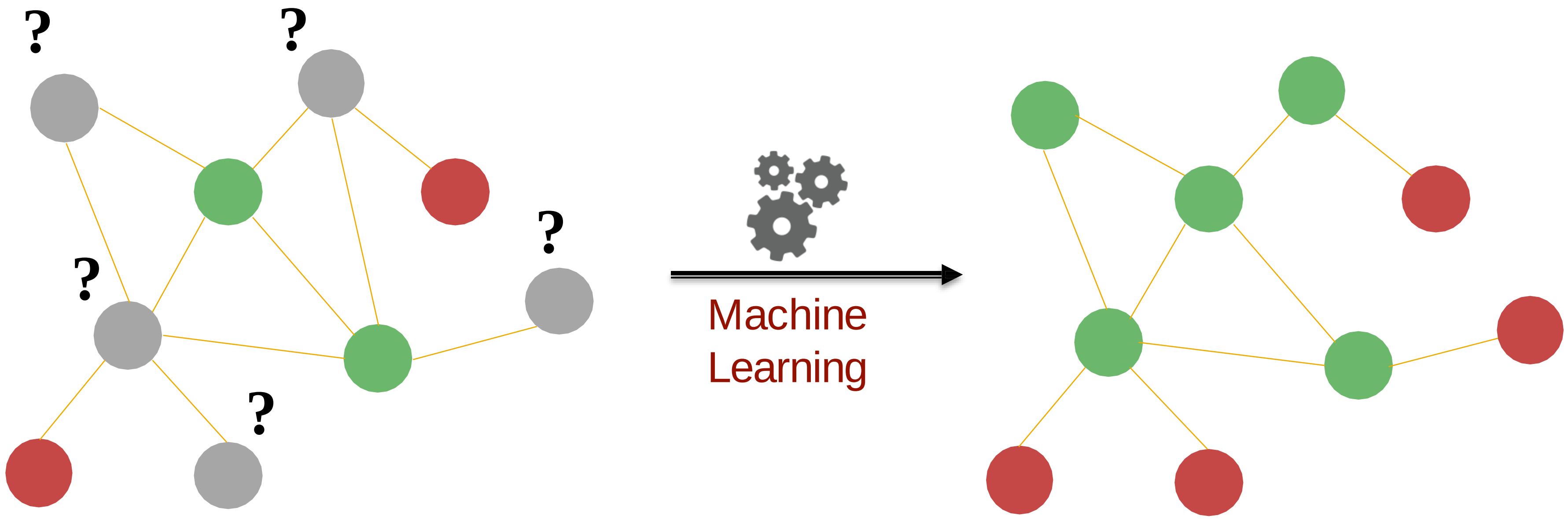
- Given: $G = (V, E)$

- Learn a function: $f : V \to \mathbb{R}$

- How do we learn the function?

Node Level Tasks and Features

Node Level Tasks



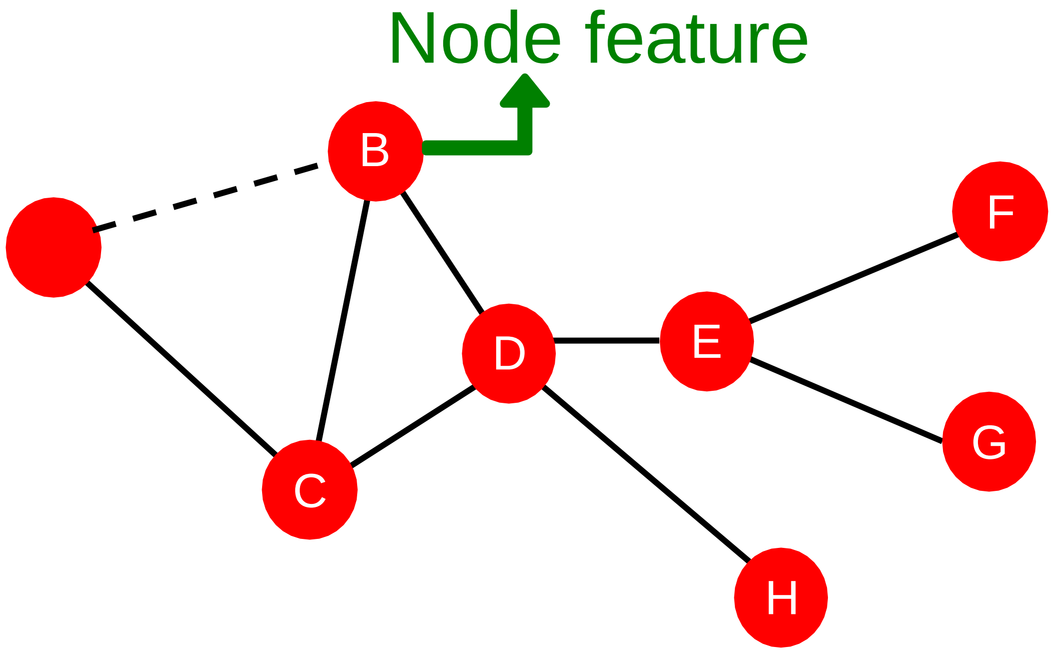
Node classification

ML needs features

(Label propagation will be introduced later)

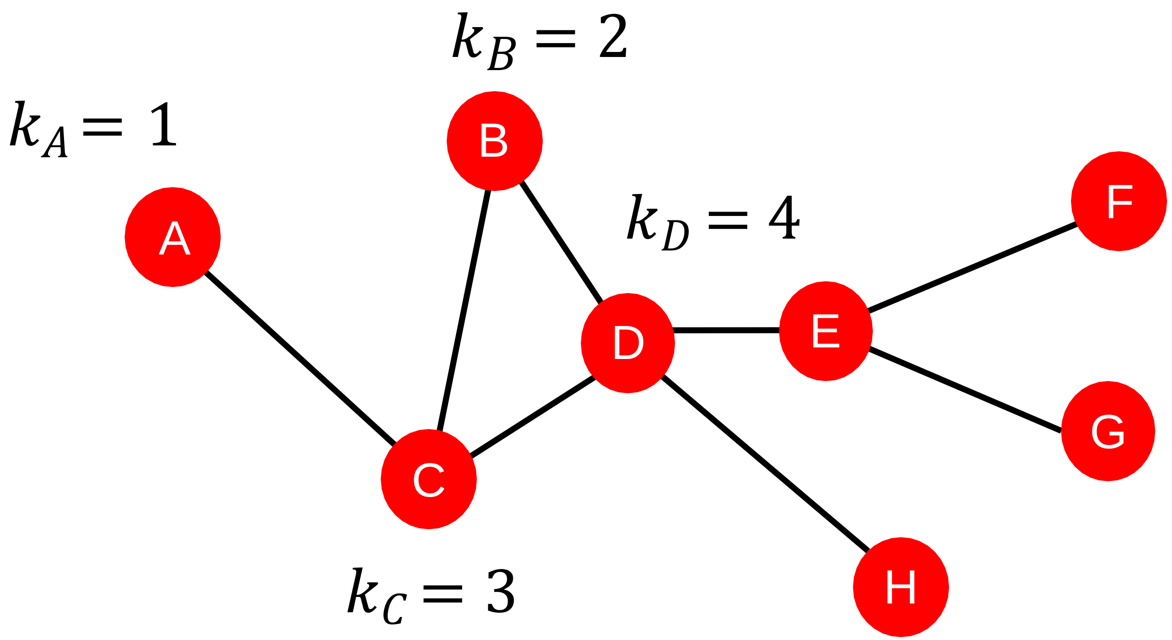
Node Level Features: Overview

- Goal: Characterize the structure and position of a node in the network:
  - Node degree
  - Node centrality
  - Graphlets



Node Features: Node Degree

- The degree k_v of node v is the number of edges (neighboring nodes) the node has.
- Treats all neighboring nodes equally.



Node Features: Node Centrality

- Node degree counts the neighboring nodes without capturing their importance.

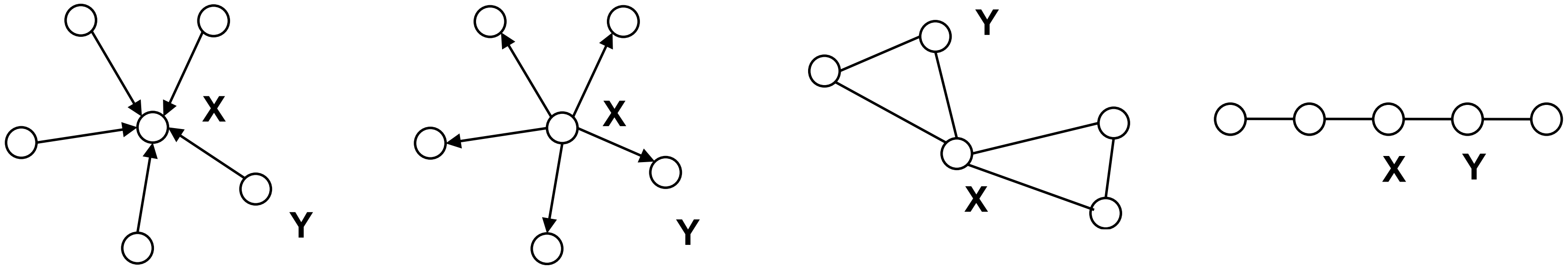
- Node centrality c_v takes the node importance in a graph into account

Centrality

- Which nodes are most ‘central’?
- Definition of ‘central’ varies by context/purpose
- Relative to rest of network:
  - closeness, betweenness, eigenvector (Bonacich power centrality), Katz, PageRank, …

Centrality: Who’s Important based on Their Network Position

In each of the following networks, X has higher centrality than Y according to a particular measure



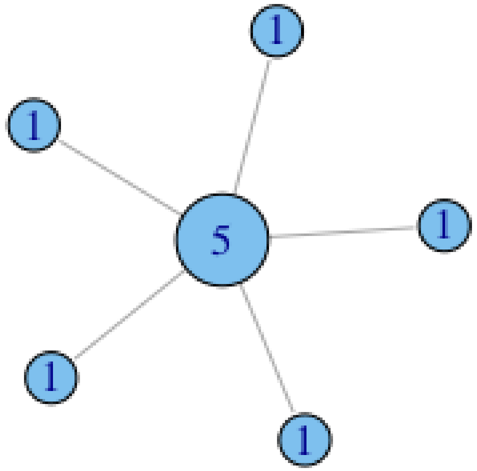
in-degree out-degree betweenness closeness

Centrality Outline

- Degree centrality
- Betweenness centrality
- Closeness centrality

Degree Centrality (Undirected)

He who has many friends is most important.



When is the number of connections the best centrality measure?

- people who will do favors for you
- people you can talk to (influence set, information access, …)
- influence of an article in terms of citations (using in-degree)

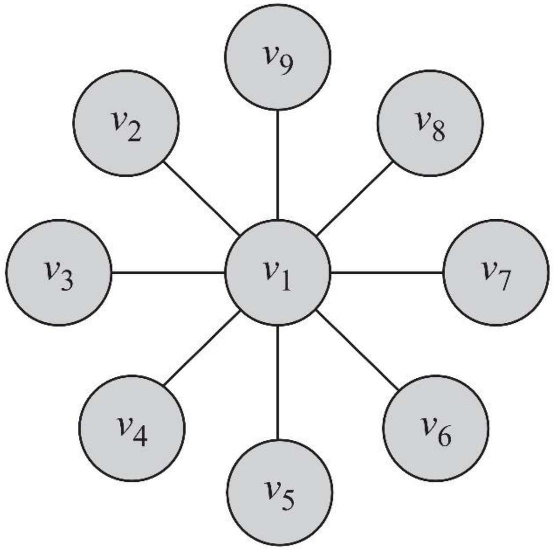
Degree Centrality

- **Degree centrality**: ranks nodes with more connections higher in terms of centrality

$$C_d(v_i) = d_i$$

- d_i is the degree (number of friends) for node v_i

In this graph, degree centrality for node v_1 is d_1=8 and for all others is $d_j = 1, j \neq 1$



Normalized Degree Centrality

- Normalized by the maximum possible degree

$$C_d^{\mathrm{norm}}(v_i) = \frac{d_i}{n-1}$$

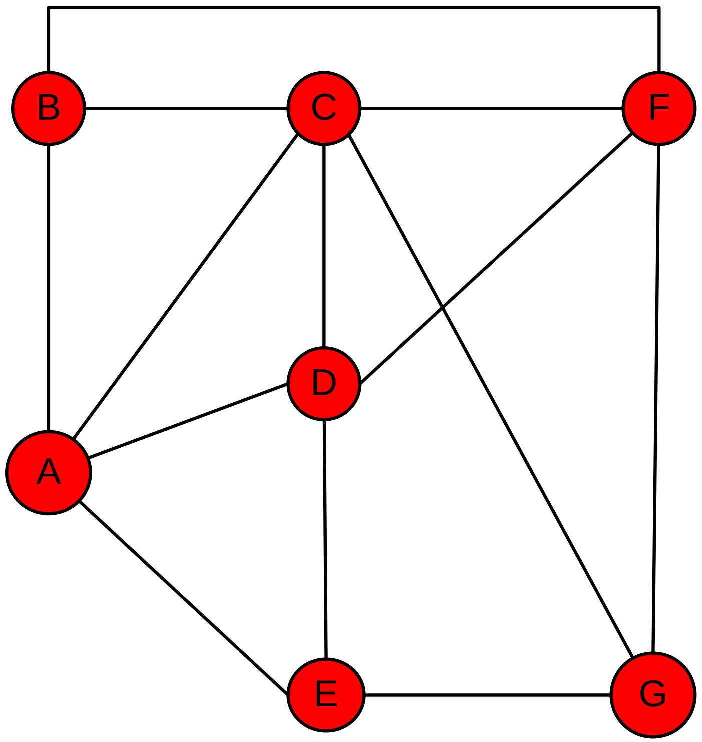
- Normalized by the maximum degree

$$C_d^{\max}(v_i) = \frac{d_i}{\max_j d_j}$$

- Normalized by the degree sum

$$C_d^{\mathrm{sum}}(v_i) = \frac{d_i}{\sum_j d_j} = \frac{d_i}{2|E|} = \frac{d_i}{2m}$$

Degree Centrality (Undirected Graph) Example



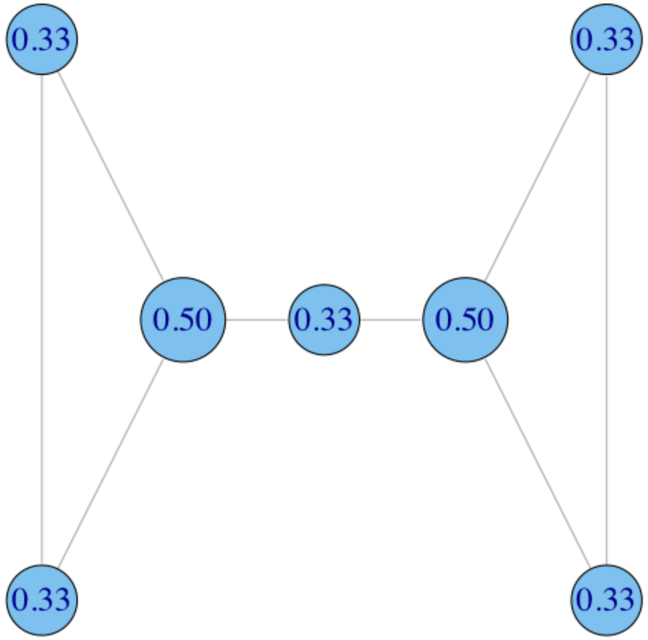
Node	Degree	Centrality	Rank
A	4	2/3	**2**
B	3	1/2	**5**
C	5	5/6	**1**
D	4	2/3	**2**
E	3	1/2	**5**
F	4	2/3	**2**
G	3	1/2	**5**

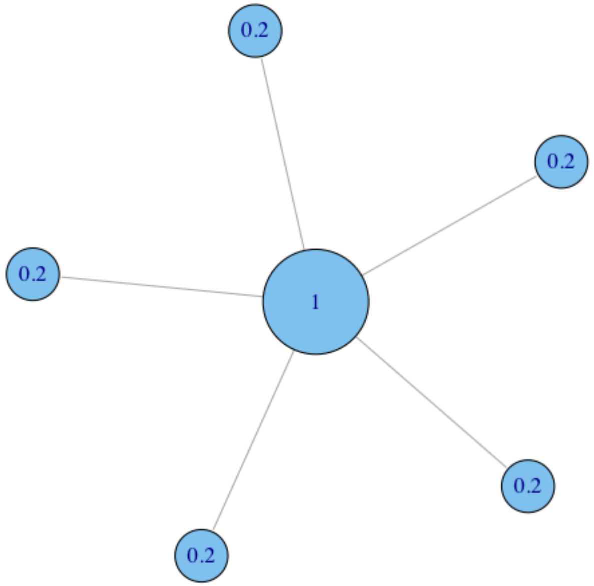
Normalized by the maximum possible degree

$$C_d^{\mathrm{norm}}(v_i) = \frac{d_i}{n-1}$$

Degree: Normalized Degree Centrality

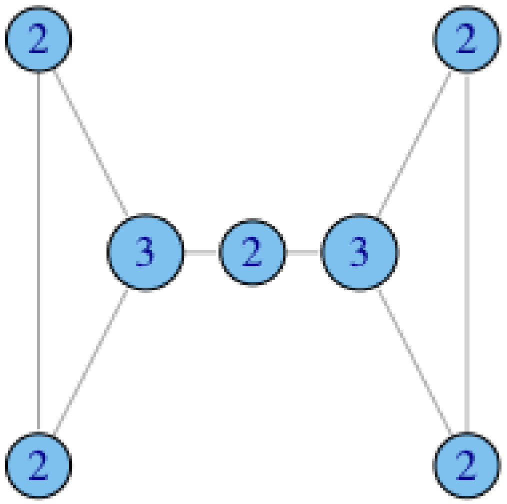
Divide by the max. possible, i.e. (N-1)





When Degree isn't Everything

In what ways does degree fail to capture centrality in the following graphs?



- Ability to broker between groups
- Likelihood that information originating anywhere in the network reaches you…

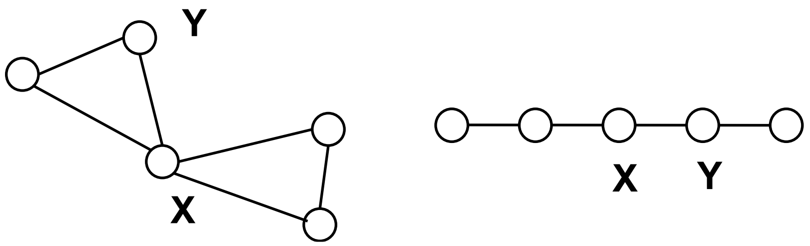
Centrality Outline

- Degree centrality
  - Centralization
- Betweenness centrality
- Closeness centrality

Betweenness: another centrality measure

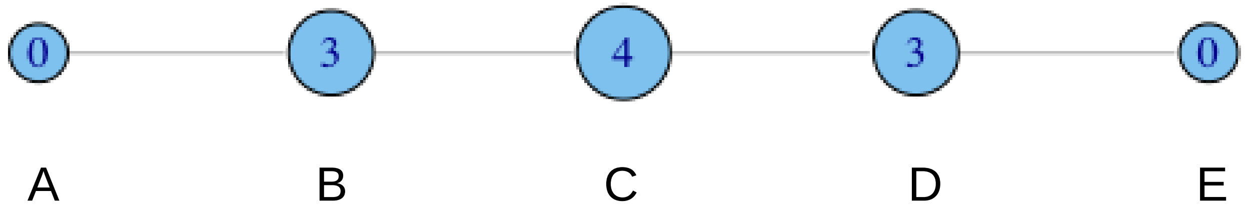
- **Intuition**: how many pairs of individuals would have to go through you in order to reach one another in the minimum number of hops?

- Who has higher betweenness, X or Y?



Betweenness on Toy Networks

- Non-normalized version:



- A lies between no two other vertices
- B lies between A and 3 other vertices: C, D, and E
- C lies between 4 pairs of vertices (A,D),(A,E),(B,D),(B,E)

- Note that there are no alternate paths for these pairs to take, so C gets full credit

Betweenness Centrality: Definition

betweenness of vertex i

paths between j and k that pass through i

$$C_B(i) = \sum_{j<k} g_{jk}(i) / g_{jk}$$

all paths between j and k

Where g_{jk} = the number of geodesics connecting *j-k*, and
g_{jk} (*i*)= the number that actor *i* is on.

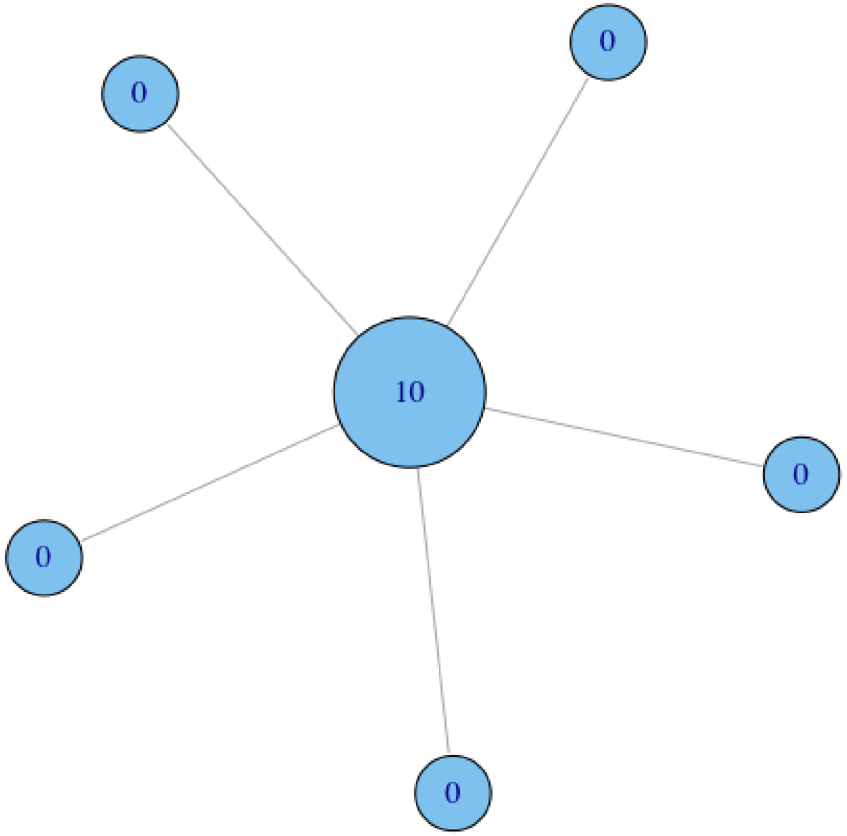
Usually further normalized by:

$$C'_B(i) = C_B(i) / \boxed{[(n-1)(n-2)/2]}$$

number of pairs of vertices excluding the vertex itself

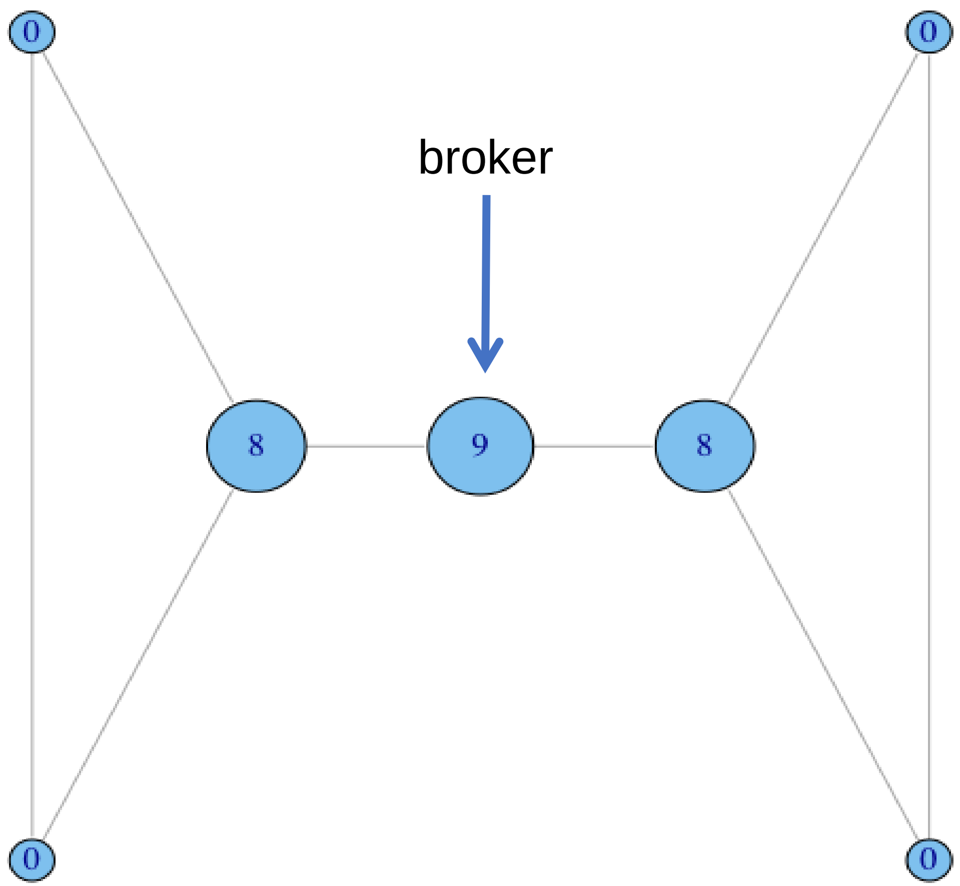
Betweenness on Toy Networks

- Non-normalized version:

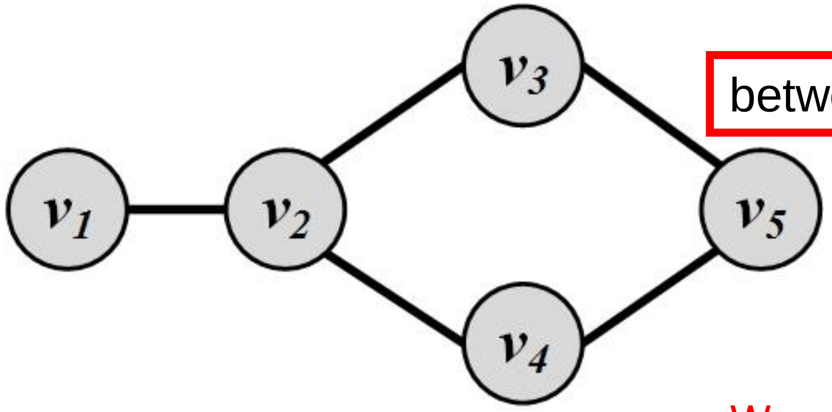


Betweenness on Toy Networks

- Non-normalized version:



Betweenness Centrality: Example



betweenness of vertex i

paths between j and k that pass through i

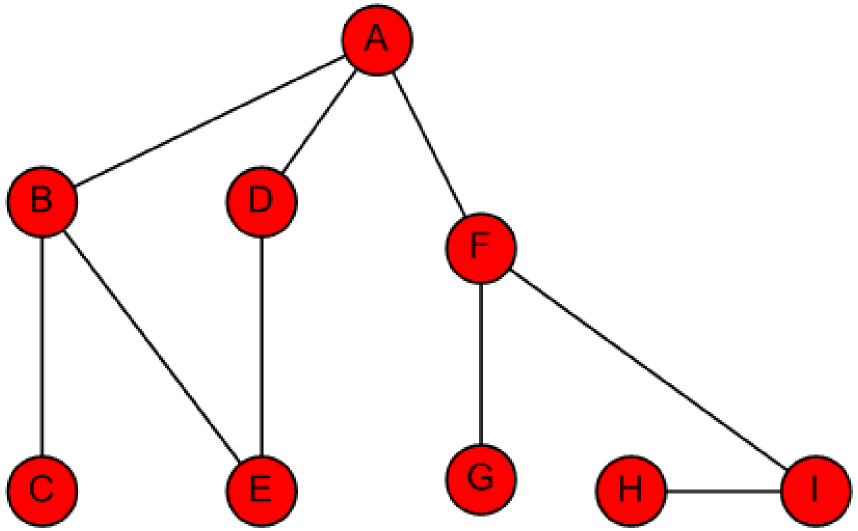
$$C_B(i) = \sum_{j<k} g_{jk}(i) / g_{jk}$$

all paths between j and k

We multiple 2 here when considering a path from j to k is different from a path from k to j

$$C_b(v_2) = 2 \times (\underbrace{(1/1)}_{s=v_1,t=v_3} + \underbrace{(1/1)}_{s=v_1,t=v_4} + \underbrace{(2/2)}_{s=v_1,t=v_5} + \underbrace{(1/2)}_{s=v_3,t=v_4} + \underbrace{0}_{s=v_3,t=v_5} + \underbrace{0}_{s=v_4,t=v_5})$$
$$= 2 \times 3.5 = 7,$$
$$C_b(v_3) = 2 \times (\underbrace{0}_{s=v_1,t=v_2} + \underbrace{0}_{s=v_1,t=v_4} + \underbrace{(1/2)}_{s=v_1,t=v_5} + \underbrace{0}_{s=v_2,t=v_4} + \underbrace{(1/2)}_{s=v_2,t=v_5} + \underbrace{0}_{s=v_4,t=v_5})$$
$$= 2 \times 1.0 = 2.$$
$$C_b(v_4) = C_b(v_3) = 2 \times 1.0 = 2,$$
$$C_b(v_5) = 2 \times (\underbrace{0}_{s=v_1,t=v_2} + \underbrace{0}_{s=v_1,t=v_3} + \underbrace{0}_{s=v_1,t=v_4} + \underbrace{0}_{s=v_2,t=v_3} + \underbrace{0}_{s=v_2,t=v_4} + \underbrace{(1/2)}_{s=v_3,t=v_4})$$
$$= 2 \times 0.5 = 1,$$

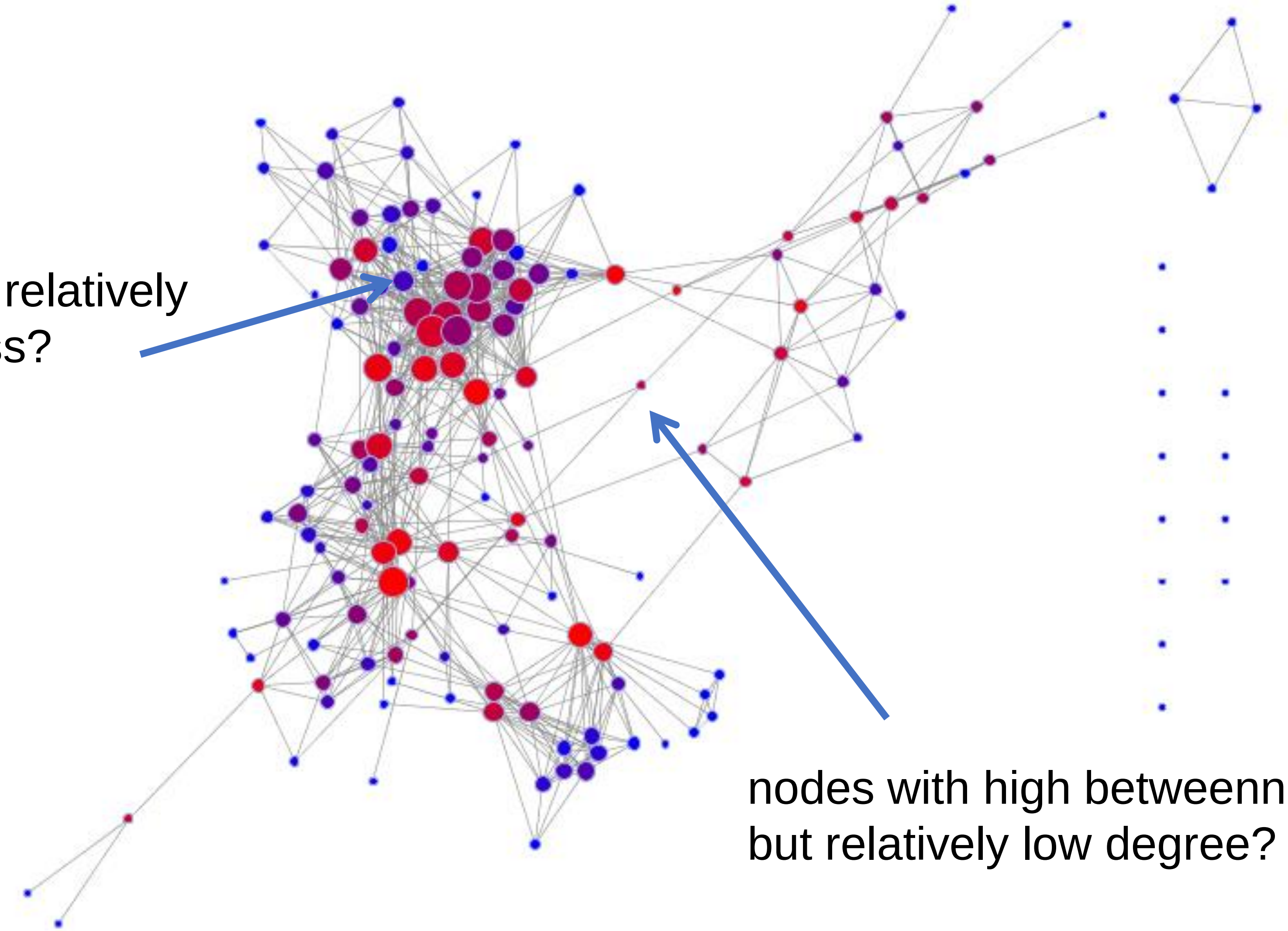
Betweenness Centrality: Example



Node	Betweenness Centrality	Rank
A	16 + 1/2 + 1/2	**1**
B	7+5/2	**3**
C	0	**7**
D	5/2	**5**
E	1/2 + 1/2	**6**
F	15 + 2	**1**
G	0	**7**
H	0	**7**
I	7	**4**

Example

Nodes are sized by degree, and colored by betweenness.



high degree but relatively low betweenness?

nodes with high betweenness but relatively low degree?

Centrality Outline

- Degree centrality
  - Centralization
- Betweenness centrality
- Closeness centrality

Closeness: Another Centrality Measure

- What if it’s not so important to have many direct friends?
- Or be “between” others
- But one still wants to be in the “middle” of things,
    - not too far from the center

Closeness Centrality: Definition

Closeness is based on the length of the average shortest path between a vertex and all vertices in the graph

Closeness Centrality:

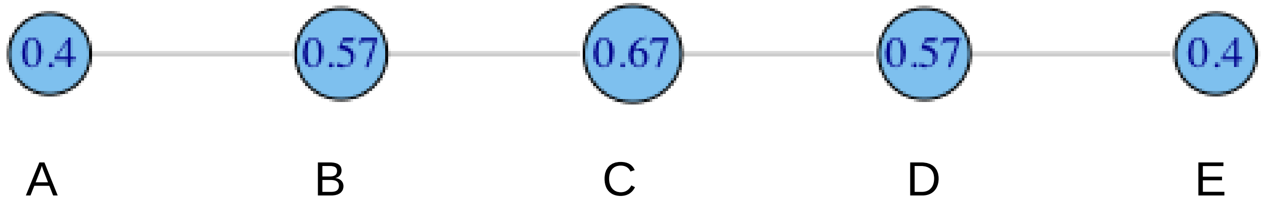
$$C_c(i) = \left[\sum_{j=1}^{N} d(i,j) \right]^{-1}$$

depends on inverse distance to other vertices

Normalized Closeness Centrality

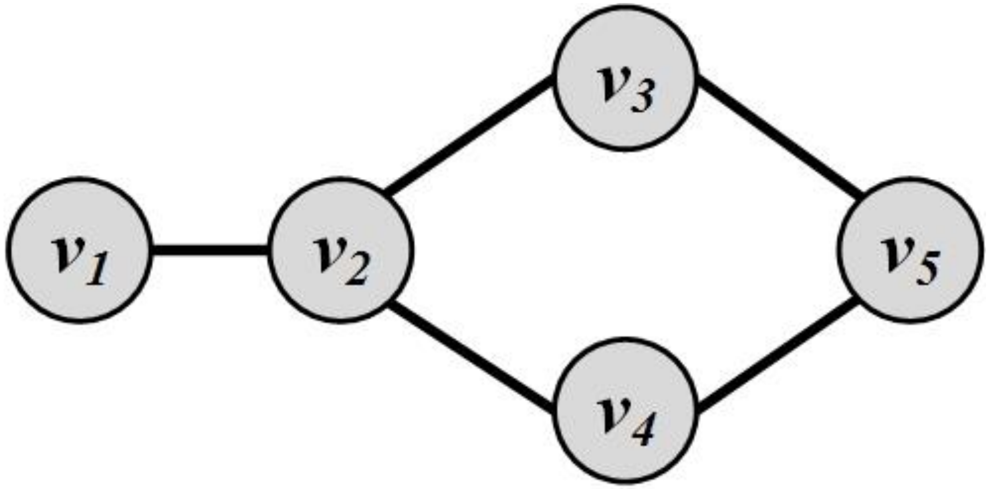
$$C_C^{'}(i) = (C_C(i)).(N-1)$$

Closeness Centrality: Toy Example



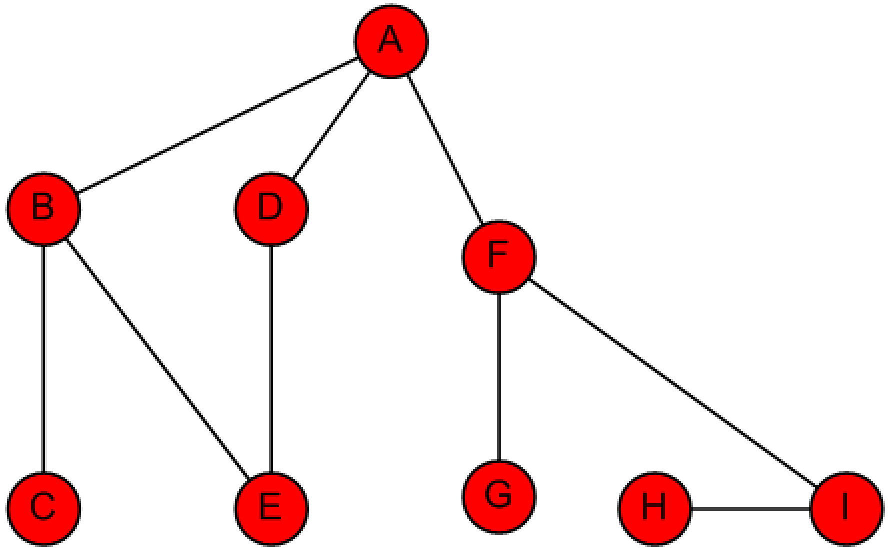
$$C_c'(A) = \left[\frac{\sum_{j=1}^{N} d(A,j)}{N-1} \right]^{-1} = \left[\frac{1+2+3+4}{4} \right]^{-1} = \left[\frac{10}{4} \right]^{-1} = 0.4$$

Closeness Centrality: Example



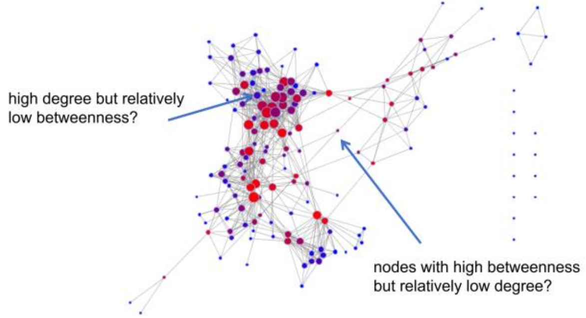
$$
\begin{aligned}
C_c(v_1) &= 1 \;/\; (\; (1+2+2+3)/4 \;) = 0.5, \\
C_c(v_2) &= 1 \;/\; (\; (1+1+1+2)/4 \;) = 0.8, \\
C_c(v_3) = \quad C_c(v_4) &= 1 \;/\; (\; (1+1+2+2)/4 \;) = 0.66, \\
C_c(v_5) &= 1 \;/\; (\; (1+1+2+3)/4 \;) = 0.57.
\end{aligned}
$$

Closeness Centrality: Example (Undirected)



Node	A	B	C	D	E	F	G	H	I	Distance_Avg	Closeness Centrality	Rank
A	0	1	2	1	2	1	2	3	2	1.750	0.571	**1**
B	1	0	1	2	1	2	3	4	3	2.125	0.471	**3**
C	2	1	0	3	2	3	4	5	4	3.000	0.333	**8**
D	1	2	3	0	1	2	3	4	3	2.375	0.421	**4**
E	2	1	2	1	0	3	4	5	4	2.750	0.364	**7**
F	1	2	3	2	3	0	1	2	1	1.875	0.533	**2**
G	2	3	4	3	4	1	0	3	2	2.750	0.364	**7**
H	3	4	5	4	5	2	3	0	1	3.375	0.296	**9**
I	2	3	4	3	4	1	2	1	0	2.500	0.400	**5**

Centrality Comparison



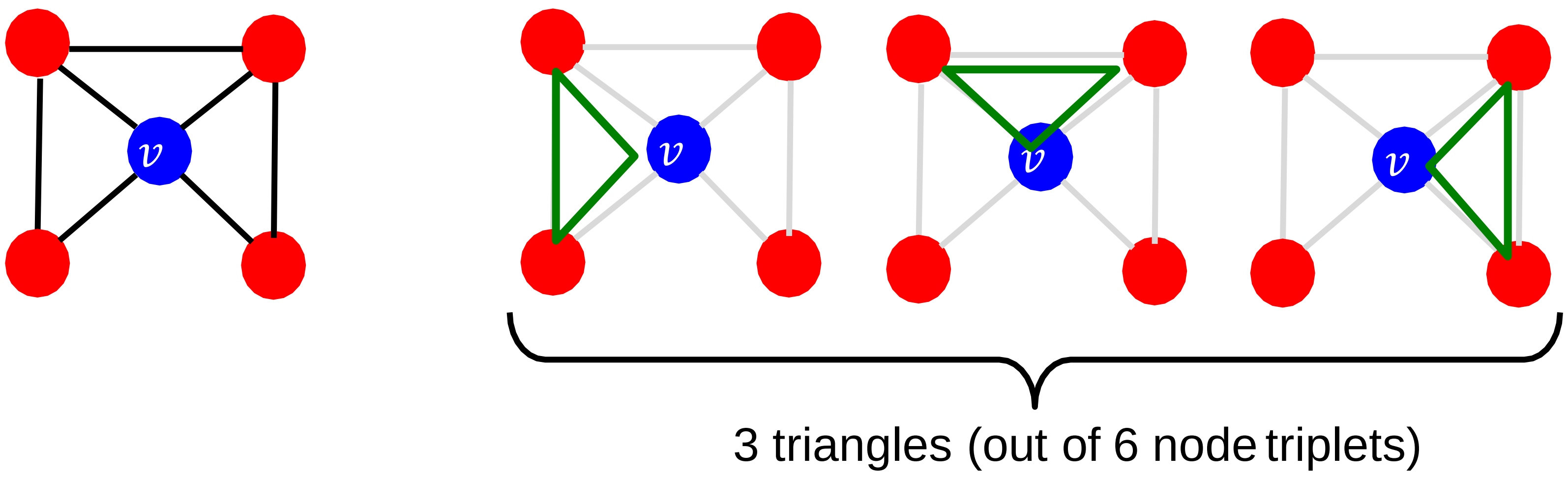
Comparing three centrality values

- Generally, the 3 centrality types will be positively correlated
- When they are not (or low correlation), it usually reveals interesting information

	Low Degree	**Low Closeness**	**Low Betweenness**
High Degree		*Node is embedded in a community that is far from the rest of the network*	*Node's connections are redundant - communication bypasses the node*
High Closeness	*Key node connected to important/active alters*		*Probably multiple paths in the network, node is near many people, but so are many others*
High Betweenness	*Node's few ties are crucial for network flow*	*Very rare! Node monopolizes the ties from a small number of people to many others.*	

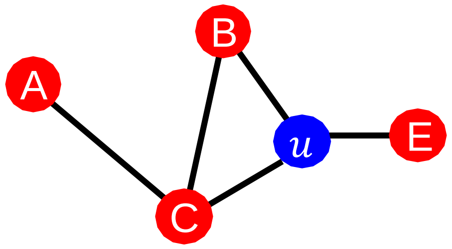
Node Features: Graphlets

- **Observation:** We can count the #(triangles) in the ego-network
- We can generalize the above by counting #(pre-specified subgraphs, i.e., **graphlets**).



Node Features: Graphlets

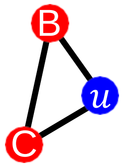
- **Goal:** Describe network structure around node u
  - **Graphlets** are small subgraphs that describe the structure of node u's network neighborhood



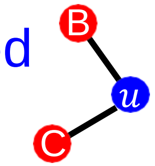
Key Concept 1: Induced Subgraph

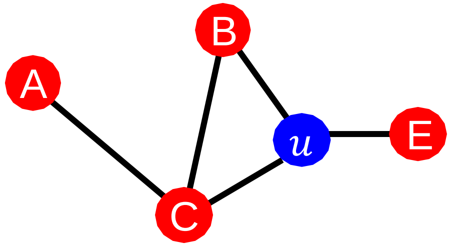
- **Def: Induced subgraph** is another graph, formed from a subset of vertices and *all* of the edges connecting the vertices in that subset.

Induced subgraph:



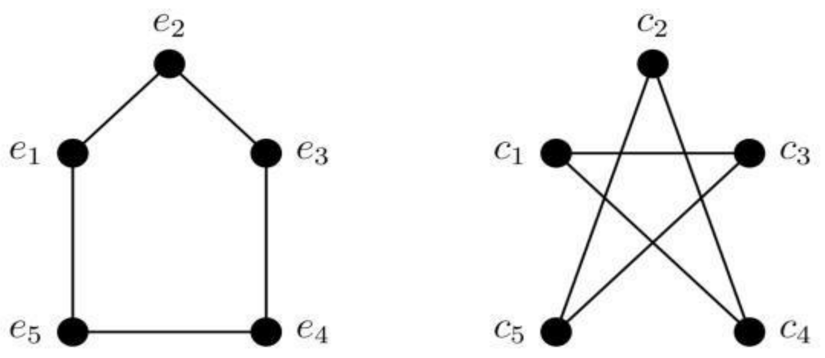
Not induced subgraph:





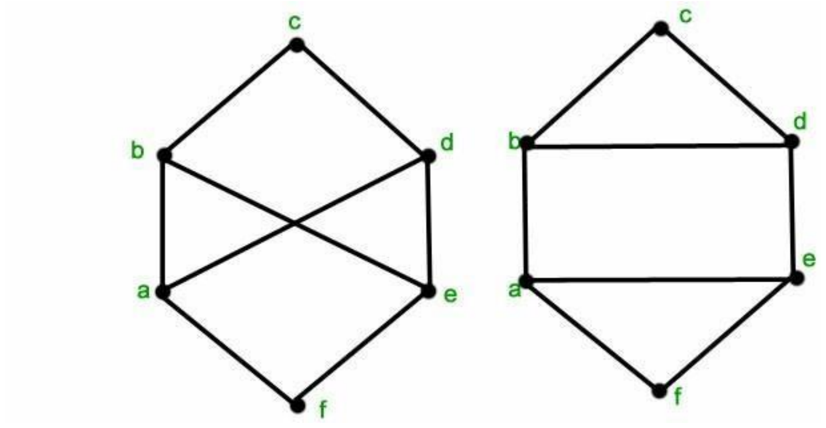
Key Concept 2: Isomorphism

- **Def: Graph Isomorphism**
  - Two graphs which contain the same number of nodes connected in the same way are said to bei somorphic



Isomorphic

Node mapping: (e2,c2), (e1,c5), (e3,c4), (e5,c3), (e4,c1)

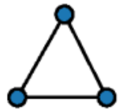
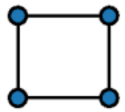
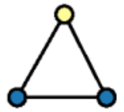
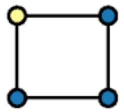
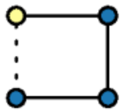
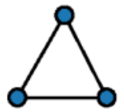
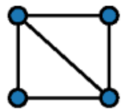
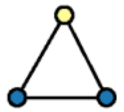
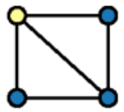
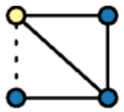
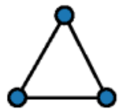
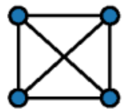
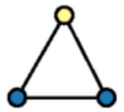
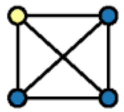
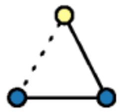
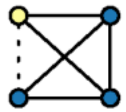


Non-Isomorphic

The right graph has cycles of length 3 but he left graph does not, so the graphs cannot be isomorphic.

Source: Mathoverflow

Subgraph Isomorphism Counting Example

Homogeneous			Heterogenous					
Pattern	Graph	Count	Pattern	Graph	Count	Pattern	Graph	Count
		0			0			0
		12			4			1
		24			6			2

Xin Liu, Haojie Pan, Mutian He, Yangqiu Song, Xin Jiang, Lifeng Shang. Neural Subgraph Isomorphism Counting.
In ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), 2020.

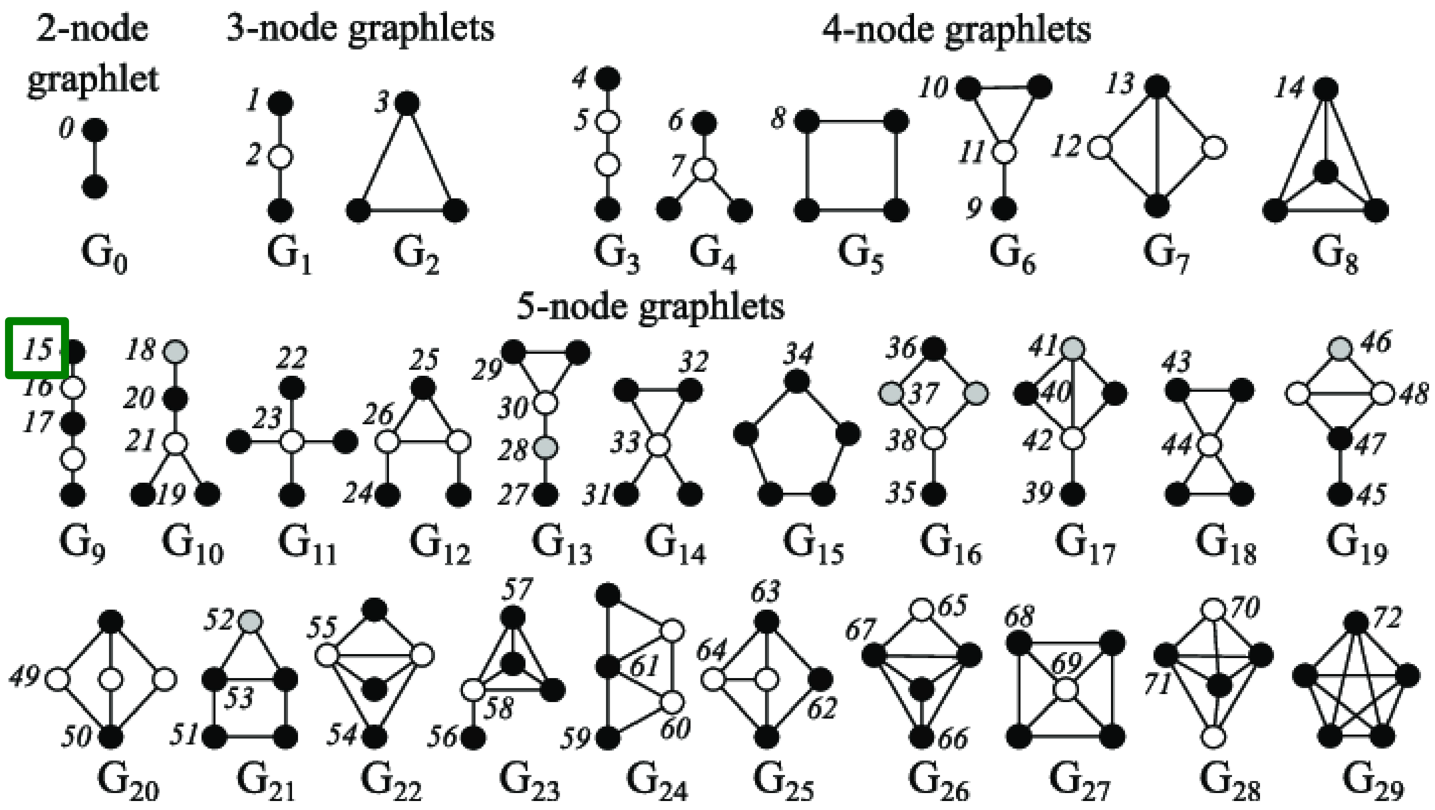
Graphlets

- Graphlets: Rooted connected induced non-isomorphic subgraphs:

There are 73 different graphlets on up to 5 nodes

Take some nodes and all the edges between them.

Graphlet id (Root / "position" of node u)

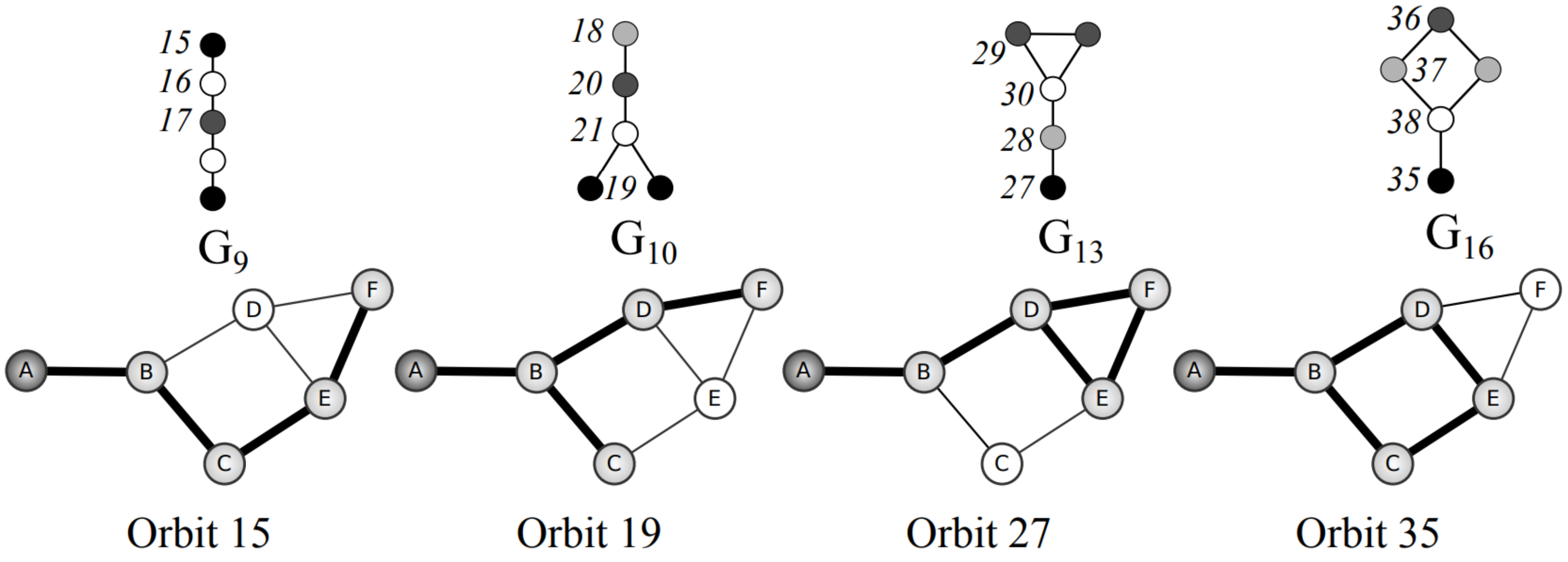


Note: Here is still on homogeneous graphs. Different colours distinguish different orbits and positions.

Nataša Pržulj. Biological network comparison using graphlet degree distribution. Bioinformatics, Volume 23, Issue 2, 15 January 2007, Pages e177–e183.

Graphlet Degree Vector

- Computation of the graphlet degree vector (GDV) of node A in the friendship network
  - GDV provides a measureof a **node's local network topology**
  - Comparing vectors of two nodes provides a more detailed measure of local topological similarity than node degrees



Orbit	0	1	2...3	4	5	6	7...14	15	16...18	19	20...26	27	28...34	35	36...72
GDV(A)	1	2	0...0	3	0	1	0...0	1	0...0	1	0...0	1	0...0	1	0...0

http://www0.cs.ucl.ac.uk/staff/natasa/ergm-graphlets.pdf

Node Level Features: Summary

- **We have introduced different ways to obtain node features.**
- **They can be categorized as:**
  - Importance-based features:
    - Node degree
    - Different node centrality measures
  - Structure-based features:
    - Node degree
    - Graphlet count vector

Node Level Features: Summary

- **Importance-based features**: capture the importance of a node in a graph
  - Node degree:
    - Simply counts the number of neighboring nodes
  - Node centrality:
    - Models importance of neighboring nodes in a graph
    - Different modeling choices: eigenvector centrality, betweenness centrality, closeness centrality
- Useful for predicting influential nodes in a graph
  - **Example:** predicting celebrity users in a social network

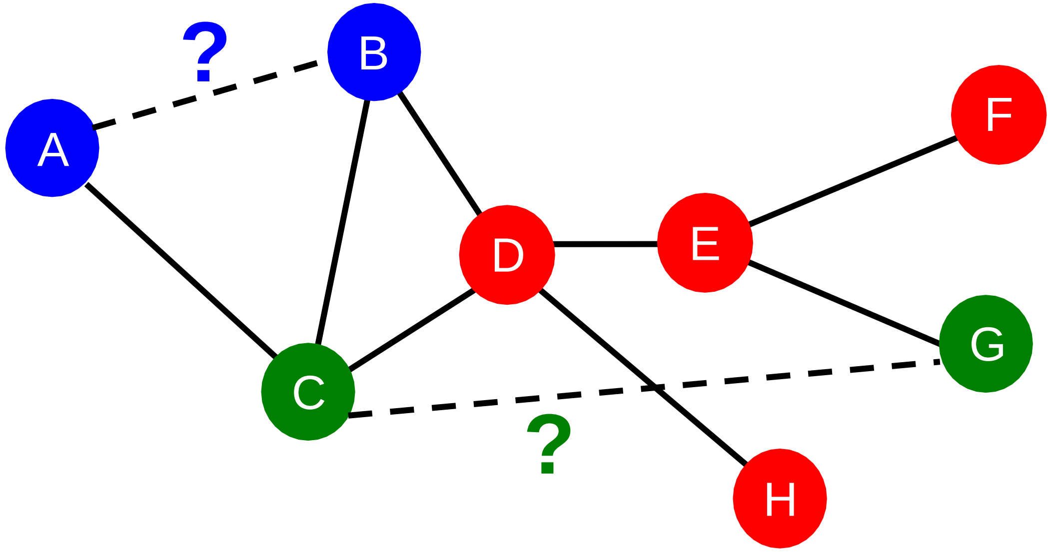
Node Level Features: Summary

- **Structure-based features**: Capture topological properties of local neighborhood around a node.
  - **Node degree:**
    - Counts the number of neighboring nodes
  - **Graphlet degree vector:**
    - Counts the occurrences of different graphlets
- **Useful for predicting a particular role a node plays in a graph:**
  - **Example:** Predicting protein functionality in a protein-protein interaction network.

Link Level Tasks and Features

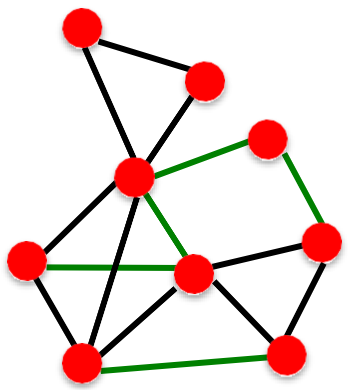
Link Level Prediction Task: Recap

- The task is to predict **new links** based on the existing links.
- At test time, node pairs (with no existing links) are ranked, and top K node pairs are predicted.
- The key is to design features for **a pair of nodes**.



Link Prediction as a Task

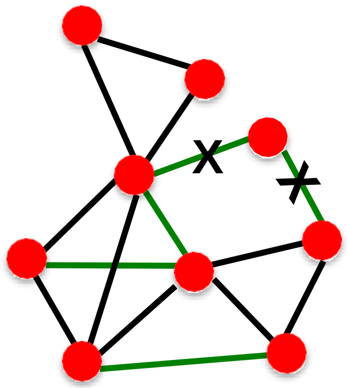
- Two formulations of the link prediction task:
  - **1) Links missing at random:**
    - Remove a random set of links and then aim to predict them
  - **2) Links over time:**
    - Given $G[t_0, t']$ a graph defined by edges up to time t', **output a ranked list L** of edges (not in $G[t_0, t']$) that are predicted to appear in time $G[t_1, t_1]$
      - Training: Facebook graph in 2021
      - Testing: Facebook graph in 2022
    - **Evaluation:**
      - $n = |E_{new}|$: # new edges that appear during the test period $[t_1, t']$
      - Take top n elements of L and count correct edges



$G[t_0\, t'_0]$
$G[t_1, t'_1]$

Link Prediction via Proximity

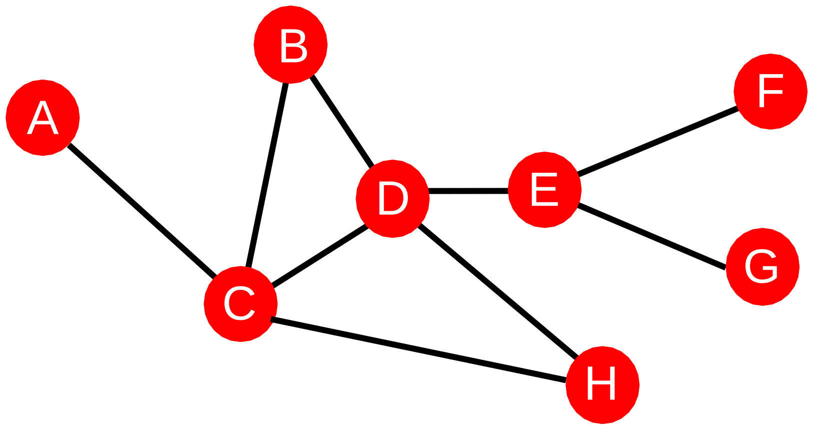
- **Methodology:**
  - For each pair of nodes *(x,y)* compute score *c(x,y)*
    - For example, *c(x,y)* could be the # of common neighbors of *x* and *y*
  - Sort pairs *(x,y)* by the decreasing score *c(x,y)*
  - **Predict top n pairs as new links**
- **See which of these links actually appear in $G[t, t']$**



- Distance-based feature
- Local neighborhood overlap
- Global neighborhood doverlap

Distance Based Features

- Shortest-path distance between two nodes
- Example:



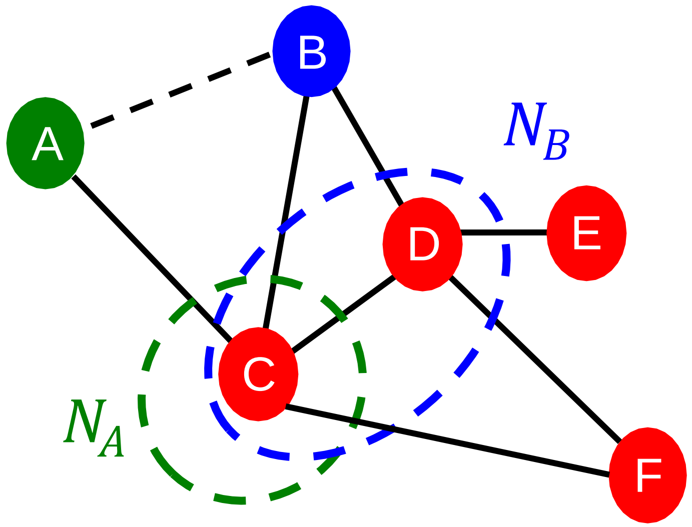
$$S_{BH} = S_{BE} = S_{AB} = 2$$
$$S_{BG} = S_{BF} = 3$$

- However, this does not capture the degree of neighborhood overlap:
  - Node pair *(B, H)* has 2 shared neighboring nodes, while pairs *(B, E)* and *(A, B)* only have 1 such node.

Local Neighborhood Overlap

- Captures # neighboring nodes shared between two nodes $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$:
  - Example: $|N(A) \cap N(B)| = |\{C\}| = 1$

- Jaccard: $\frac{|N(A) \cap N(B)|}{|N(A) \cup N(B)|} = \frac{|\{C\}|}{|\{C,D\}|} = \frac{1}{2}$



Link Level Features: Summary

- **Distance-based features:**
  - Uses the shortest path length between two nodes but does not capture how neighborhood overlaps.
- **Local neighborhood overlap:**
  - Captures how many neighboring nodes are shared by two nodes.
  - Becomes zero when no neighbor nodes are shared.
- **Global neighborhood overlap:**
  - Ommitted; will be illustrated in Personalized PageRank / Label Propagation