Network Mining

Community detection and link prediction

— Session 2 —

Agenda

1. Community detection
    1. Introduction
    2. Taxonomy of approaches
    3. Evaluation
2. Link prediction
    1. Introduction
    2. Proximity measures
    3. Supervised framework for link prediction

1. Community Detection

Community

Community. It is formed by individuals such that those within a group interact with each other more frequently than with those outside the group, a.k.a. **group, cluster, cohesive subgroup, module** in different contexts

Community detection. discovering groups in a network where individuals' group memberships are not explicitly given

Why communities in social media?

- Human beings are social
- Easy-to-use social media allows people to extend their social life in unprecedented ways
- Difficult to meet friends in the physical world, but much easier to find friend online with similar interests
- Interactions between nodes can help determine communities

Community in Social Media

Two types of groups in social media

- **Explicit Groups:** formed by user subscriptions
- **Implicit Groups:** implicitly formed by social interactions

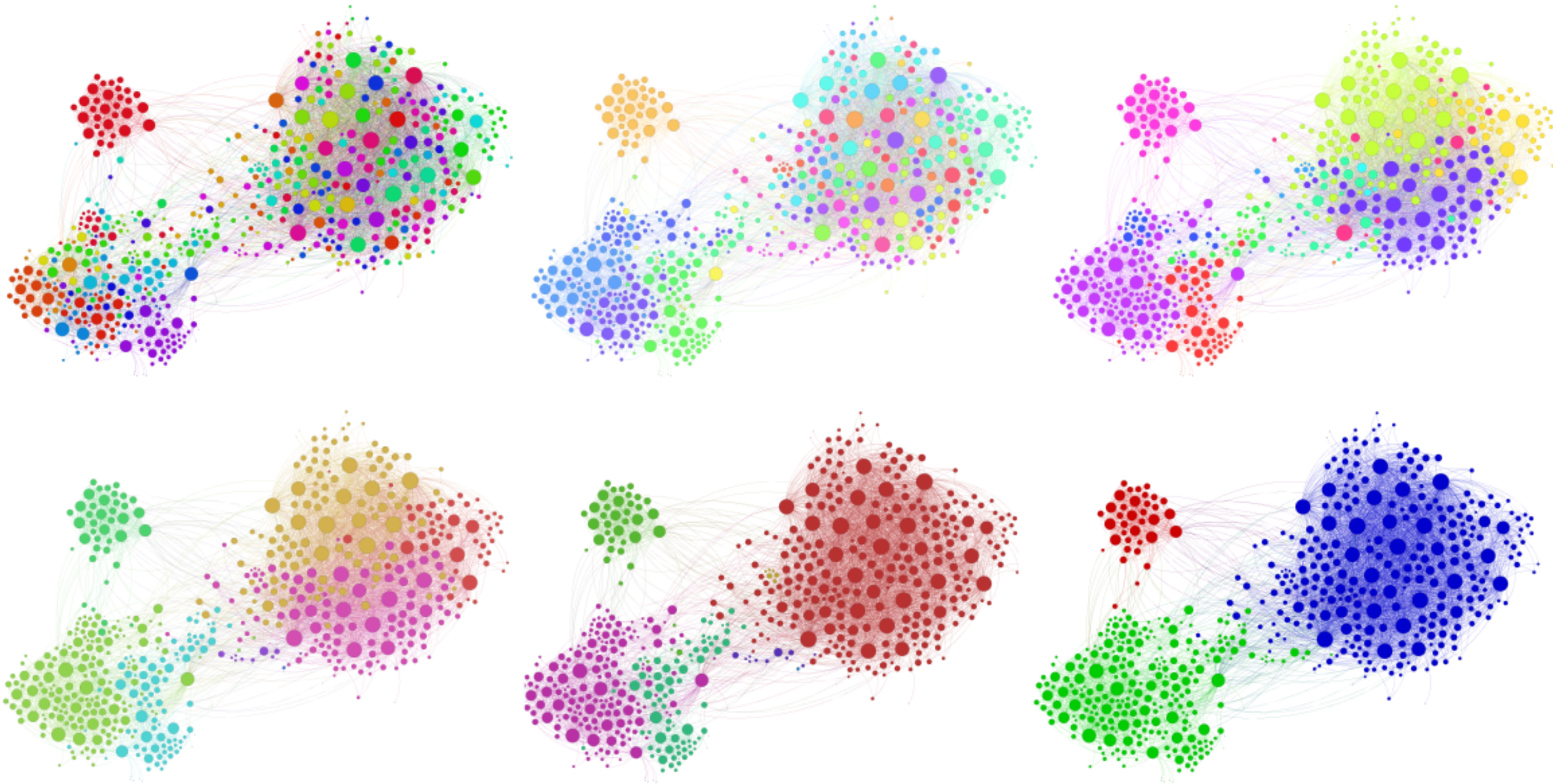
Some social media sites allow people to join groups, is it necessary to extract groups based on network topology?

- Not all sites provide community platform
- Not all people want to make effort to join groups
- Groups can change dynamically

Network interaction provides rich information about the relationship between users

- Can complement other kinds of information
- Help network visualization and navigation
- Provide basic information for other tasks

Subjectivity in Community Detection



Source: https://pegasusdata.com/2013/01/10/facebook-friends-network-mapping-a-gephi-tutorial/

Taxonomy of Community Criteria

Criteria vary depending on the tasks

Community detection methods can be divided into 4 categories (not exclusive):

1 - Node-Centric Community

Each node in a group satisfies certain properties

2 - Group-Centric Community

Consider the connections within a group as a whole. The group has to satisfy certain properties without zooming into node-level

3 - Network-Centric Community

Partition the whole network into several disjoint sets

4 - Hierarchy-Centric Community

Construct a hierarchical structure of communities

Node-Centric Community Detection

Nodes that satisfy different properties

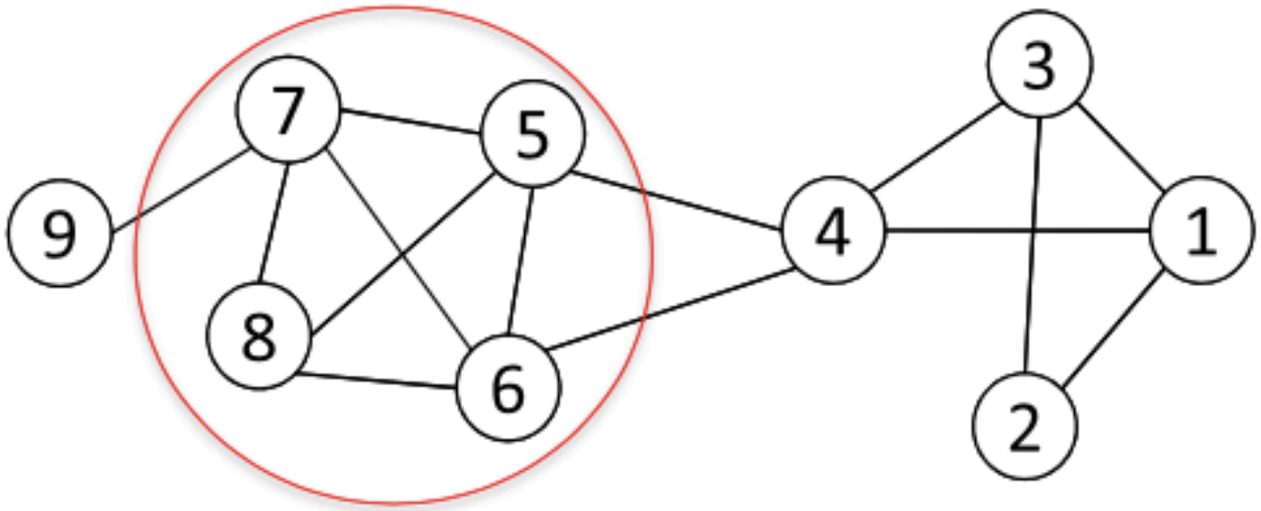
- Complete Mutuality
  - Cliques
- Reachability of members
  - k-clique, k-clan, k-club
- Nodal degrees
  - k-plex, k-core
- Relative frequency of Within-Outside Ties
  - LS sets, Lambda sets

Commonly used in traditional social network analysis

Note: only some representative ones are discussed here

Complete Mutuality: Cliques

A clique is a complete maximal subgraph



Nodes 5, 6, 7 and 8 form a clique

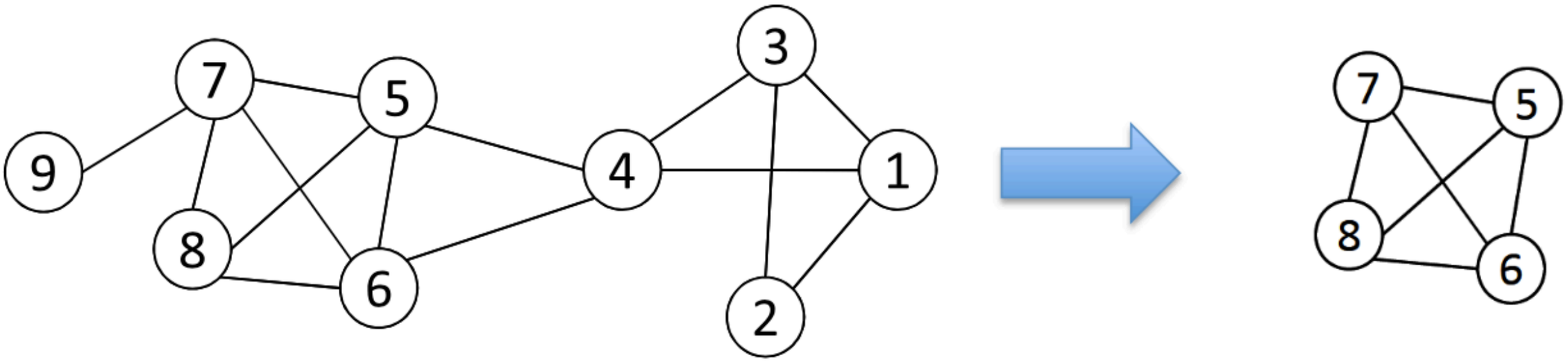
- NP-hard to find the maximum clique in a network
- Straightforward implementation to find cliques is very expensive in time complexity

Finding the Maximum Clique

- In a clique of size k, each node has a degree ≥ k-1
- Nodes with degree < k-1 will not be included in the maximum clique
- Recursive pruning procedure :
  - Sample a sub-network from the given network, and find a clique in the sub-network, say, by a greedy approach
  - Suppose the clique above is size k, in order to find out a *larger* clique, all nodes with degree ≤ k-1 should be removed
- Repeat until the network is small enough
- In social media, many nodes are removed as social networks follow a power law distribution for node degrees

Maximum Clique

Example



- Suppose we sample a sub-network with nodes 1 to 5 and find a 3-clique {1,2,3}
- In order to find a clique > 3, remove nodes with degree ≤ 2

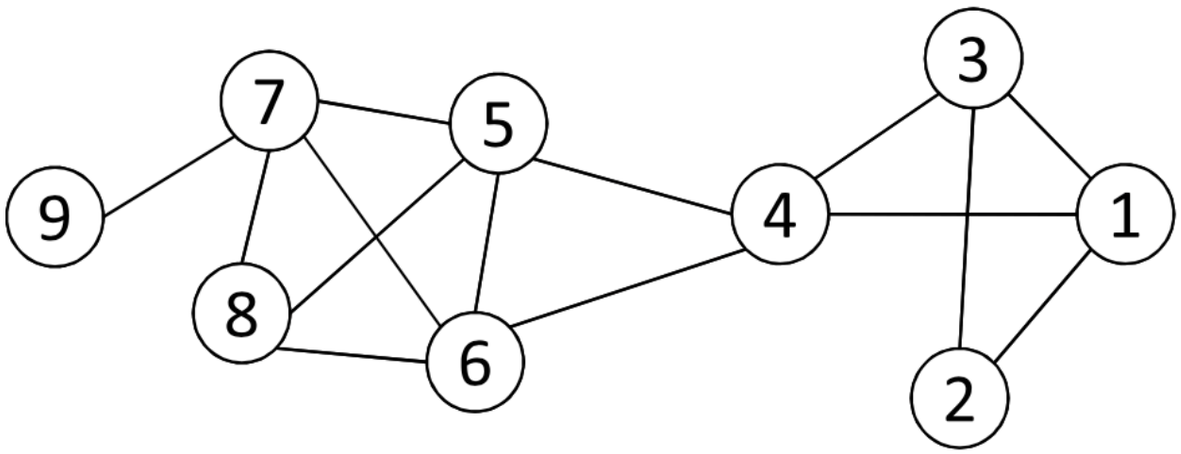
Clique Percolation Method (CPM)

- Clique is a very strict definition, unstable
- Normally use cliques as a core or a seed to find larger communities

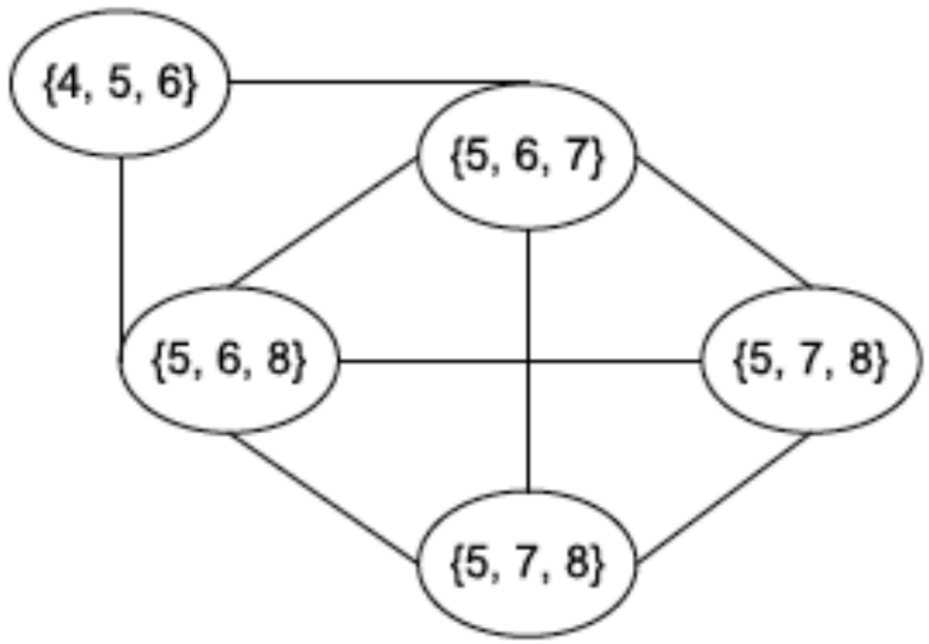
- CPM is such a method to find overlapping communities
  - **Input**
    - A parameter k, and a network
  - **Procedure**
    - Find out all cliques of size k in a given network
    - Construct a clique graph. Two cliques are adjacent if they share k-1 nodes
    - Each connected components in the clique graph form a community

CPM

Example



Cliques of size 3:
{1, 2, 3}, {1, 3, 4}, {4, 5, 6},
{5, 6, 7}, {5, 6, 8}, {5, 7, 8},
{6, 7, 8}



Communities:
{1, 2, 3, 4}
{4, 5, 6, 7, 8}

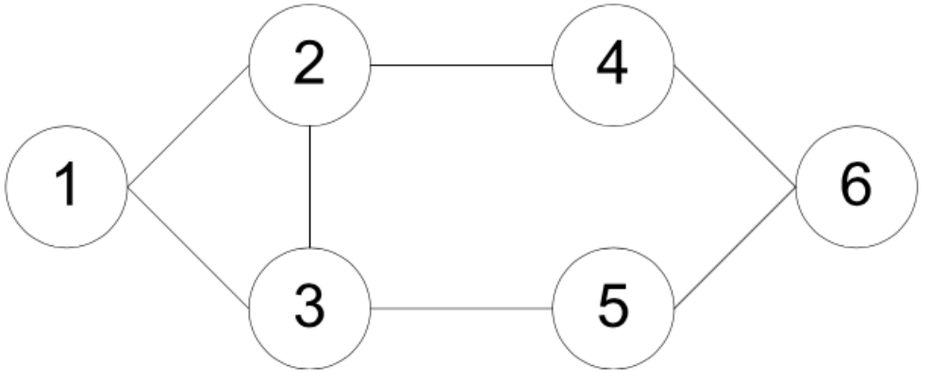
Reachability

k-clique and k-club

k-clique: a subgraph in which the largest geodesic distance between any nodes ≤ k

k-club: a substructure of diameter ≤ k

Note that any k-club in G is also a k-clique, however the converse is not true, since all shortest paths between a pair of vertices in a k-clique C can contain vertices outside C



Cliques: {1, 2, 3}
2-cliques: {1, 2, 3, 4, 5}, {2, 3, 4, 5, 6}
2-clubs: {1,2,3,4}, {1, 2, 3, 5}, {2, 3, 4, 5, 6}

{1,2,3,4,5} is not a 2-club because length(4,5) in the subgraph induced by {1,2,3,4,5} is greater than 2

- Commonly used in traditional SNA
- Often involves combinatorial optimization to find k-cliques or k-clubs with high values for k (k > maximum degree)

Group-Centric Community Detection

Density-Based Groups

The group-centric criterion requires the whole group to satisfy a certain condition

E.g., the group density >= a given threshold

A subgraph $G_s(V_s, E_s)$ is a γ-dense quasi-clique if:

$$\frac{|E_s|}{|V_s|(|V_s|-1)/2} \geq \gamma$$

A similar strategy to that of cliques can be used

- Sample a subgraph, and find a maximal γ-dense quasi-clique (say, of size k)
- Remove nodes with degree < $k\gamma$

Network-Centric Community Detection

Network-centric criterion needs to consider the connections within a network globally

Goal: partition nodes of a network into disjoint sets

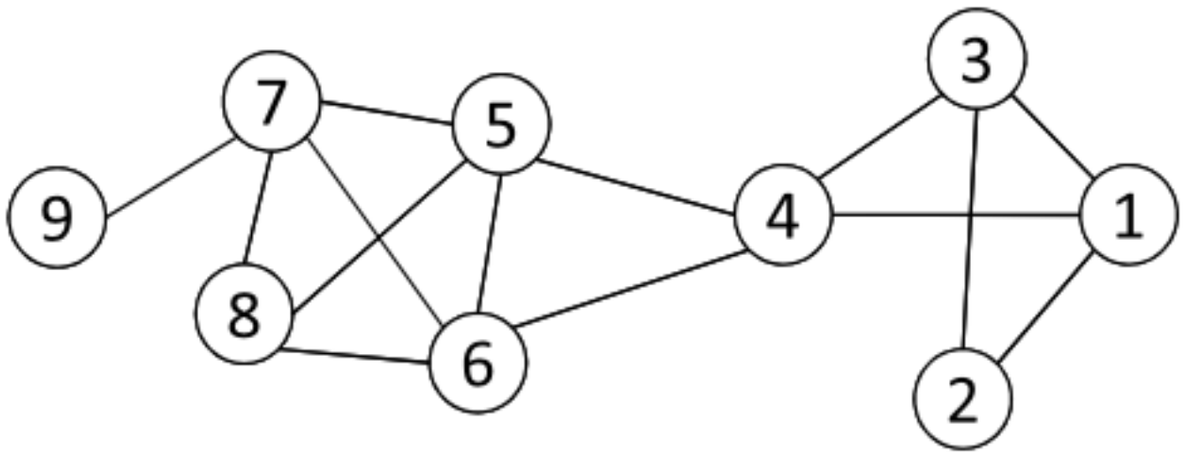
Sub-categories

- Clustering based on vertex similarity
- Latent space models
- Block model approximation
- Spectral clustering
- Modularity maximization
- Label propagation

Clustering Based on Vertex Similarity

- Apply k-means or similarity-based clustering to nodes
- Vertex similarity is defined in terms of **the similarity of their neighborhood**
- **Structural equivalence:** two nodes are structurally equivalent iff they are connecting to the same set of actors

Nodes 1 and 3 are structurally equivalent; So are nodes 5 and 7.

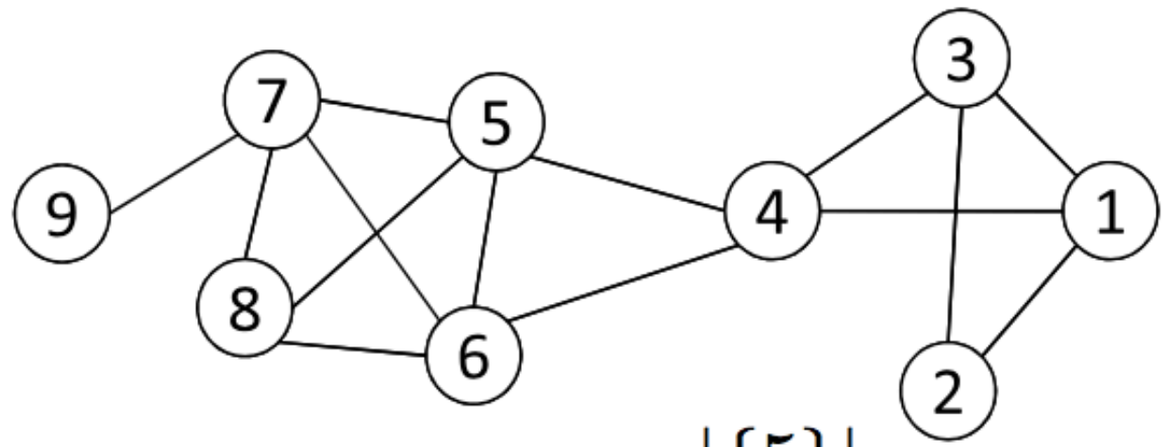


Structural equivalence is too restrictive for practical use

Vertex Similarity

Jaccard Similarity $Jaccard(v_i, v_j) = \frac{|N_i \cup N_j|}{|N_i \cap N_j|}$

Cosine similarity $cosine(v_i, v_j) = \frac{\sum_k A_{ik} A_{jk}}{\sqrt{\sum_k A_{ik}^2}\sqrt{\sum_k A_{jk}^2}}$

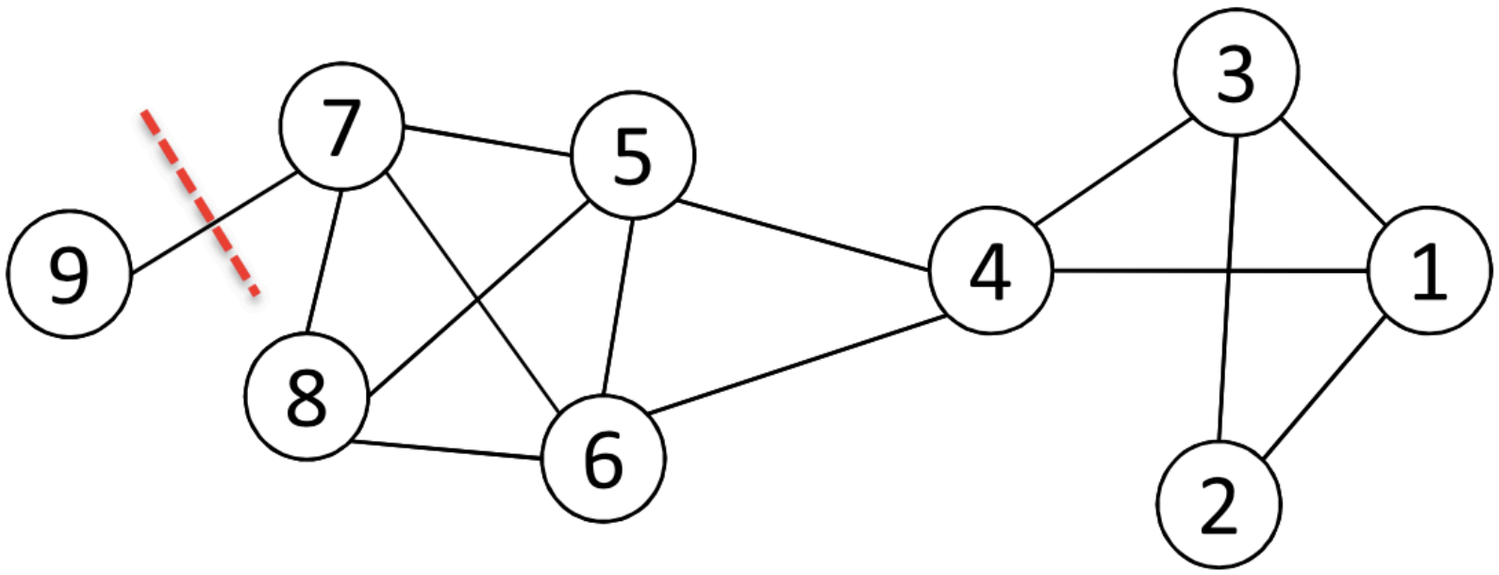


$$Jaccard(4,6) = \frac{|\{5\}|}{|\{1,3,4,5,6,7,8\}|} = \frac{1}{7}$$

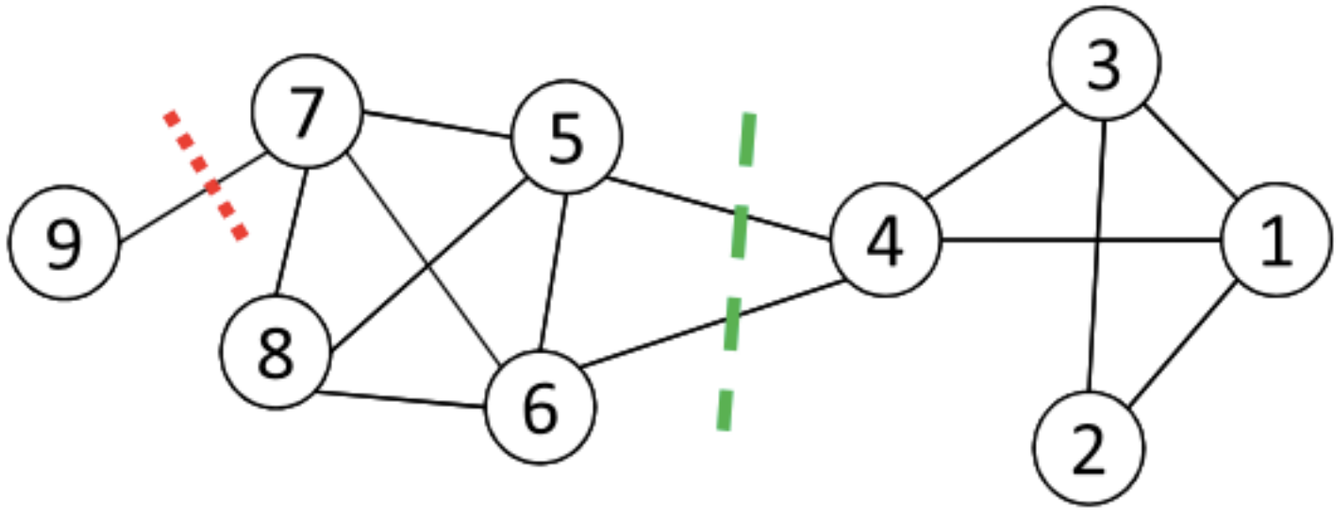
$$cosine(4,6) = \frac{1}{\sqrt{4}\sqrt{4}} = \frac{1}{4}$$

Minimum Cut

- Most interactions are within group whereas interactions between groups are few
- **Community detection: minimum cut problem**
- **Cut:** A partition of vertices of a graph into two disjoint sets
- **Minimum cut problem:** find a graph partition such that the number of edges between the two sets (the cut value) is minimized



Ratio Cut and Normalized Cut



- Minimum cut often returns an unbalanced partition, with one set being a singleton
- **Change the objective function to take the size of the communities into account**

$$\text{Ratio Cut}(\pi) = \frac{1}{k}\sum_{i=1}^{k}\frac{cut(C_i, \bar{C}_i)}{|C_i|},$$

$$\text{Normalized Cut}(\pi) = \frac{1}{k}\sum_{i=1}^{k}\frac{cut(C_i, \bar{C}_i)}{vol(C_i)}$$

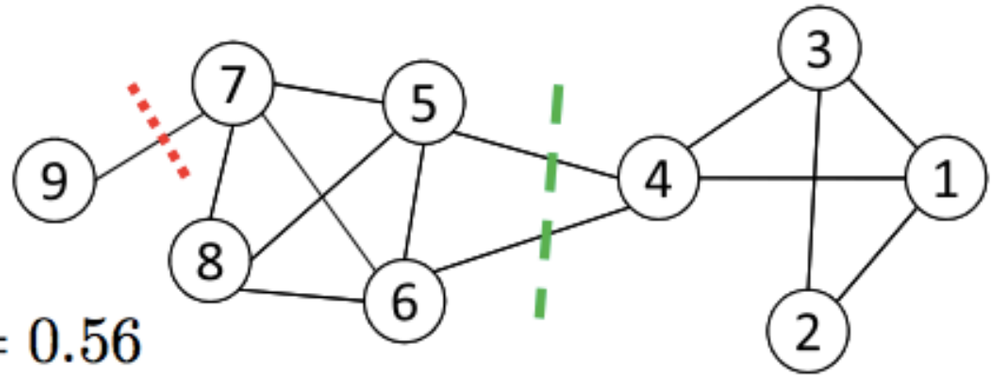
C_i is a community

$cut(C_i, \overline{C}_i)$ is the number of edges between C_i and $\overline{C}_i$

$vol(C_i)$ is a the sum of degrees of nodes in C_i

Ratio Cut and Normalized Cut

Example



For partition in red: π_1

$$\text{Ratio Cut}(\pi_1) = \frac{1}{2}\left(\frac{1}{1} + \frac{1}{8}\right) = 9/16 = 0.56$$

$$\text{Normalized Cut}(\pi_1) = \frac{1}{2}\left(\frac{1}{1} + \frac{1}{27}\right) = 14/27 = 0.52$$

For partition in green: π_2

$$\text{Ratio Cut}(\pi_2) = \frac{1}{2}\left(\frac{2}{4} + \frac{2}{5}\right) = 9/20 = 0.45 < \text{Ratio Cut}(\pi_1)$$

$$\text{Normalized Cut}(\pi_2) = \frac{1}{2}\left(\frac{2}{12} + \frac{2}{16}\right) = 7/48 = 0.15 < \text{Normalized Cut}(\pi_1)$$

Both ratio cut and normalized cut prefer a balanced partition

Modularity

- A metric to assess the quality of a partition (clustering)
- Networks with high modularity have dense connections between the nodes within modules but sparse connections between nodes in different modules

Sum of weights of the adjacent edges

Kronecker's Delta

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i * k_j}{2m} \right] \delta(c_i, c_j)$$

Number of edges

The class associated with node I

Modularity Maximization

- Modularity is NP-hard to optimize (Brandes, 2007)
- Greedy Heuristic (Newman, 2003)
  - C = trivial clustering with each node in its own cluster
  - Repeat:
    - Merge the two clusters that will increase the modularity by the largest amount
    - Stop when all merges would reduce the modularity.
- Louvain (Blondel et al., 2008)
  - Very efficient
  - See next slide for intuition about of the algorithm
- Other approximate methods exist to solve this problem, e.g., spectral technique

Modularity Maximization

Louvain algorithm

- Greed optimization algorithm in two steps:

    1. Looks for small communities (local modularity optimization)
    2. Aggregates nodes in the same community and builds network whose nodes are communities

- Steps repeated until a maximum of modularity is attained
- Exact computational complexity not known
- In practice O(n log n)

Hierarchy-Centric Community Detection

Goal: build a hierarchical structure of communities based on network topology

Allow the analysis of a network at different resolutions

Representative approaches:

- Divisive Hierarchical Clustering
- Agglomerative Hierarchical clustering

Divisive Hierarchical Clustering

Divisive clustering

- Partition nodes into several sets
- Each set is further divided into smaller ones
- Network-centric partition can be applied for the partition

One particular example: recursively remove the “weakest” tie

- Find the edge with the least strength
- Remove the edge and update the corresponding strength of each edge

Recursively apply the above two steps until a network is discomposed into desired number of connected components.

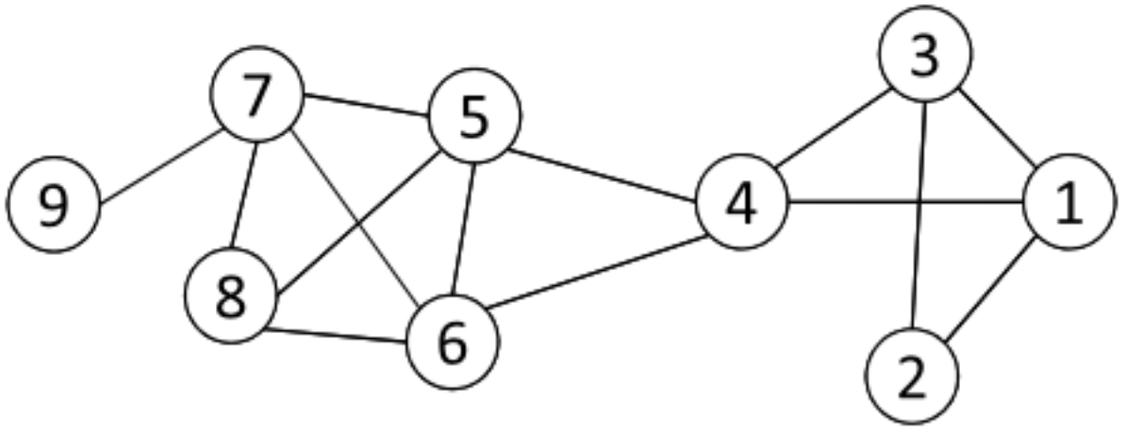
Each component forms a community

Edge Betweenness

The strength of a tie can be measured by **edge betweenness**

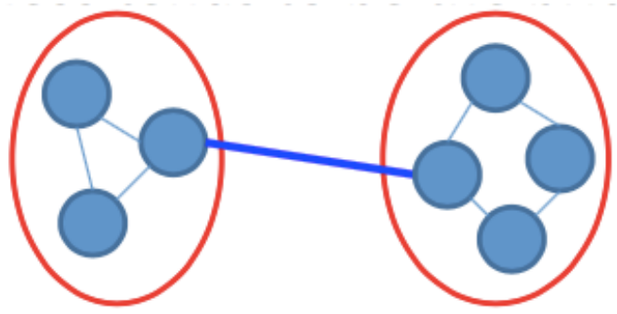
Edge betweenness: the number of shortest paths that pass along with the edge

$$\text{edge-betweenness(e)} = \Sigma_{s<t} \frac{\sigma_{st}(e)}{\sigma_{s,t}}$$

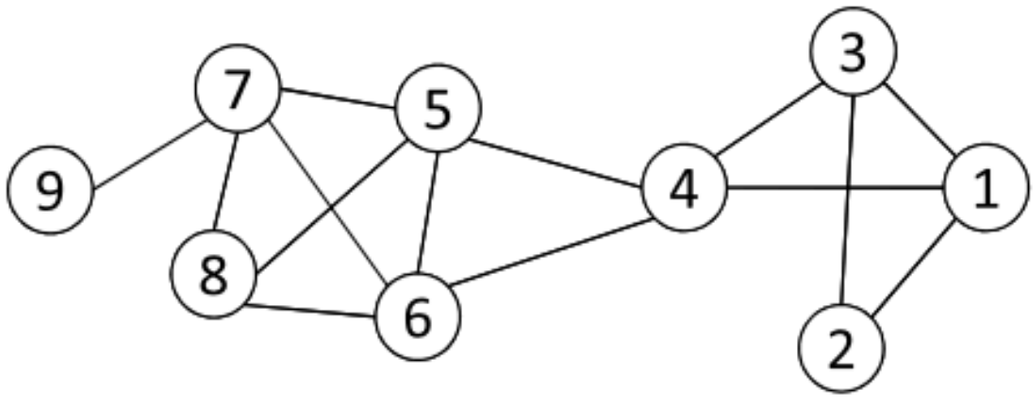


The edge betweenness of e(1, 2) is 4, as all the shortest paths from 2 to {4, 5, 6, 7, 8, 9} have to either pass e(1, 2) or e(2, 3), and e(1,2) is the shortest path between 1 and 2

The edge with the highest betweenness tends to be a bridge between 2 communities



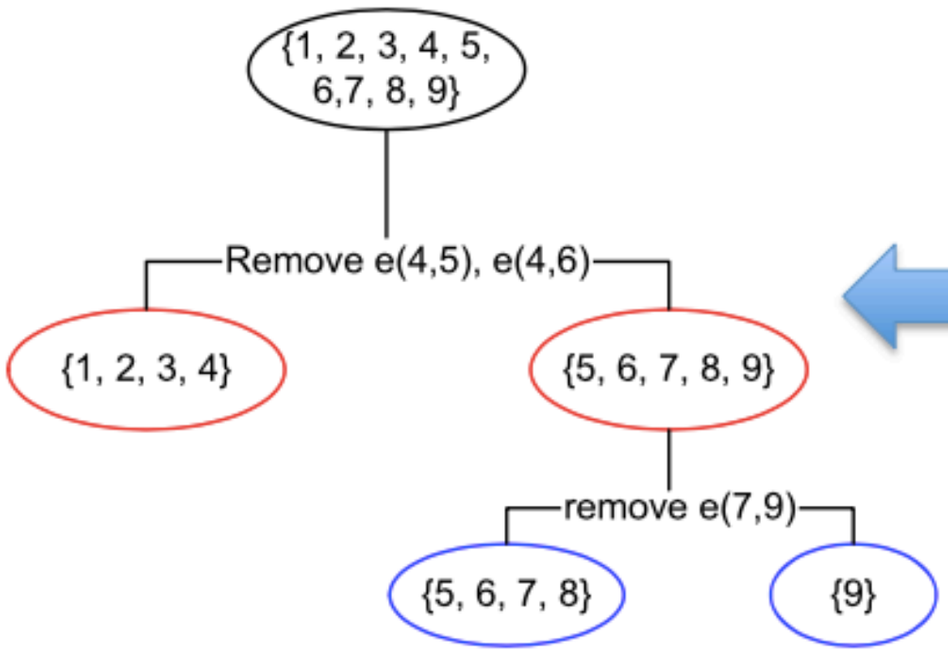
Divisive Clustering based on Edge Betweenness



Initial betweenness value

Table 3.3: Edge Betweenness

	1	2	3	4	5	6	7	8	9
1	0	4	1	9	0	0	0	0	0
2	4	0	4	0	0	0	0	0	0
3	1	4	0	9	0	0	0	0	0
4	9	0	9	0	10	10	0	0	0
5	0	0	0	10	0	1	6	3	0
6	0	0	0	10	1	0	6	3	0
7	0	0	0	0	6	6	0	2	8
8	0	0	0	0	3	3	2	0	0
9	0	0	0	0	0	0	8	0	0



After remove e(4,5), the betweenness of e(4, 6) becomes 20, which is the highest;

After remove e(4,6), the edge e(7,9) has the highest betweenness value 4, and should be removed.

Community Evaluation

Evaluating Community Detection

For groups with a clear and formal definition

- E.g., cliques, k-cliques, k-clubs, …
- Verify if the extracted communities satisfy the definition

For networks with ground truth information

- Measuring the quality of a partition (without ground-truth)
- Normalized Mutual Information (with ground-truth)
- Accuracy of pairwise community memberships

Evaluation without Ground Truth

- For networks without ground truth or semantic information
- This is the most common situation
- A option is to resort cross-validation
  - Extract communities from a (training) network
  - Evaluate the quality of the community detection on a network constructed from a different date or based on a related type of interaction

- **Quantitative evaluation**
  - Modularity
  - Block model approximation error
  - Coverage
  - Performance

Measuring a Clustering Result

Without ground-truth

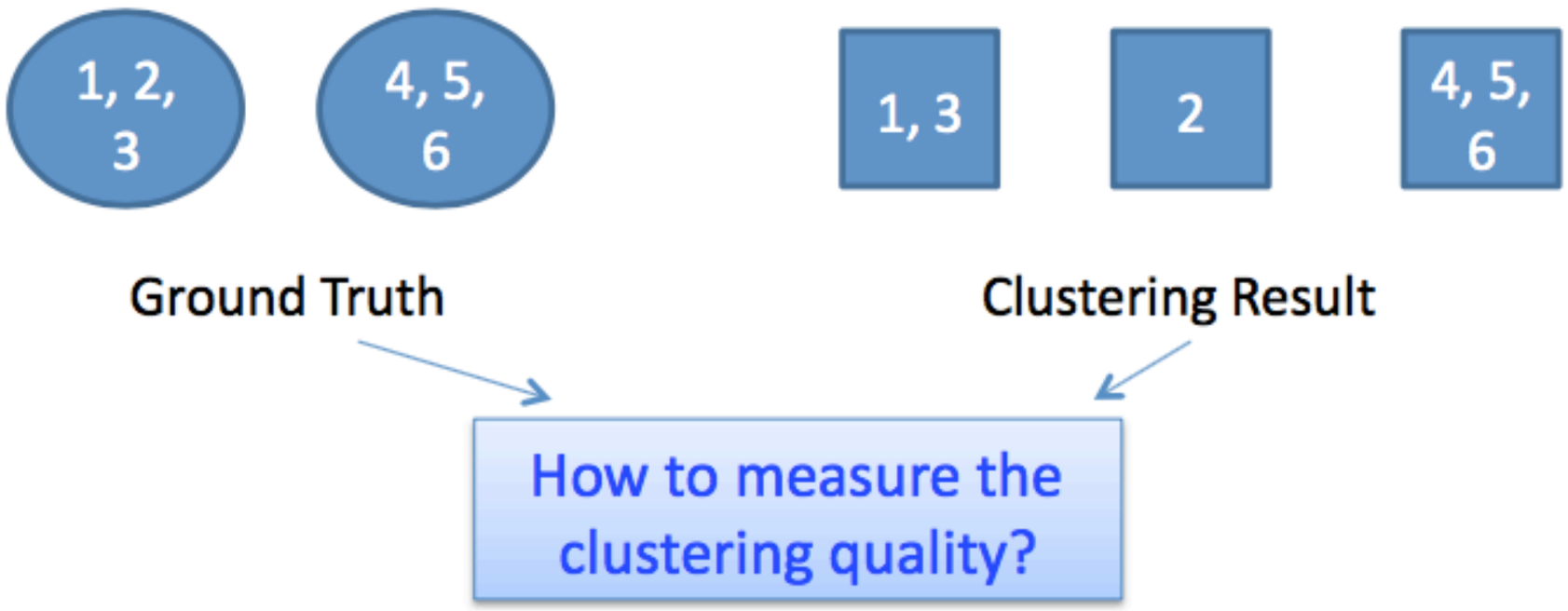
- **Coverage [Fortunato, 2010]**
  - Ratio of the number of intra-community edges to the total number of edges in the graph

- **Performance [Fortunato, 2010]**
  - Ratio of the number of intra-community edges plus inter-community non-edges with the total number of potential edges

- Works only with partitions

Measuring a Clustering Result

With ground-truth



- The number of communities after grouping can be different from the ground truth
- No clear community correspondance between clustering result and the ground truth
- Normalized Mutual Information can be used

Normalized Mutual Information

Entropy

- The information contained in a distribution

$$H(X) = \sum_{x \in X} p(x) \log p(x)$$

Mutual Information

- The shared information between two distributions

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left(\frac{p(x,y)}{p_1(x) p_2(y)} \right)$$

Normalized Mutual Information (between 0 and 1)

$$NMI(X;Y) = \frac{I(X;Y)}{\sqrt{H(X)H(Y)}}$$

Consider a partition as a distribution (probability of one node falling into one community), we can compute the matching between two clusterings

Normalized Mutual Information

$$H(X) = \sum_{x \in X} p(x) \log p(x)$$

$$H(\pi^a) = \sum_{h}^{k^{(a)}} \frac{n_h^a}{n} \log(\frac{n_h^a}{n})$$

$$H(\pi^b) = \sum_{\ell}^{k^{(b)}} \frac{n_\ell^b}{n} \log(\frac{n_\ell^b}{n})$$

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left(\frac{p(x,y)}{p_1(x) p_2(y)} \right) \Rightarrow I(\pi^a, \pi^b) = \sum_{h} \sum_{\ell} \frac{n_{h,\ell}}{n} \log \left(\frac{\frac{n_{h,\ell}}{n}}{\frac{n_h^a}{n} \frac{n_\ell^b}{n}} \right)$$

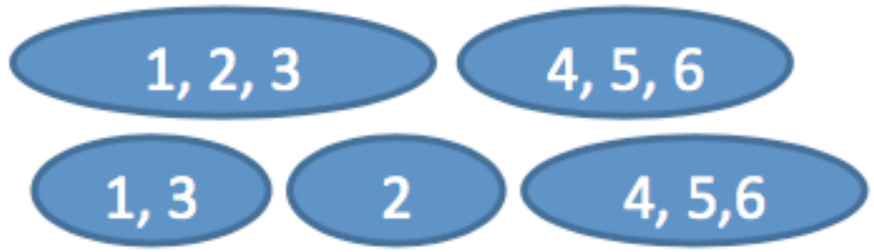
$$NMI(X;Y) = \frac{I(X;Y)}{\sqrt{H(X)H(Y)}}$$

$$NMI(\pi^a, \pi^b) = \frac{\sum_{h=1}^{k^{(a)}} \sum_{\ell=1}^{k^{(b)}} n_{h,\ell} \log \left(\frac{n \cdot n_{h,l}}{n_h^{(a)} \cdot n_\ell^{(b)}} \right)}{\sqrt{\left(\sum_{h=1}^{k^{(a)}} n_h^{(a)} \log \frac{n_h^a}{n} \right) \left(\sum_{\ell=1}^{k^{(b)}} n_\ell^{(b)} \log \frac{n_\ell^b}{n} \right)}}$$

Normalized Mutual Information

Example

- Partition a: [1, 1, 1, 2, 2, 2]
- Partition b: [1, 2, 1, 3, 3, 3]



$n = 6$

$k^{(a)} = 2$

$k^{(b)} = 3$

	n_h^a
h=1	3
h=2	3

	n_l^b
l=1	2
l=2	1
l=3	3

$n_{h,l}$	l=1	l=2	l=3
h=1	2	1	0
h=2	0	0	3

$$NMI(\pi^a, \pi^b) = \frac{\sum_{h=1}^{k^{(a)}} \sum_{\ell=1}^{k^{(b)}} n_{h,\ell} \log\left(\frac{n \cdot n_{h,l}}{n_h^{(a)} \cdot n_\ell^{(b)}}\right)}{\sqrt{\left(\sum_{h=1}^{k^{(a)}} n_h^{(a)} \log \frac{n_h^a}{n}\right)\left(\sum_{\ell=1}^{k^{(b)}} n_\ell^{(b)} \log \frac{n_\ell^b}{n}\right)}} \quad =0.8278$$

Evaluation with Semantics

- For networks with semantics
  - Networks come with semantic or attribute information of nodes or connections
  - Human subjects can check whether the extracted communities are coherent and homogeneous

- **Evaluation is qualitative**

- It is intuitive and helps in understanding a community

An *animal* community

A *health* community

2. Link Prediction

Outline

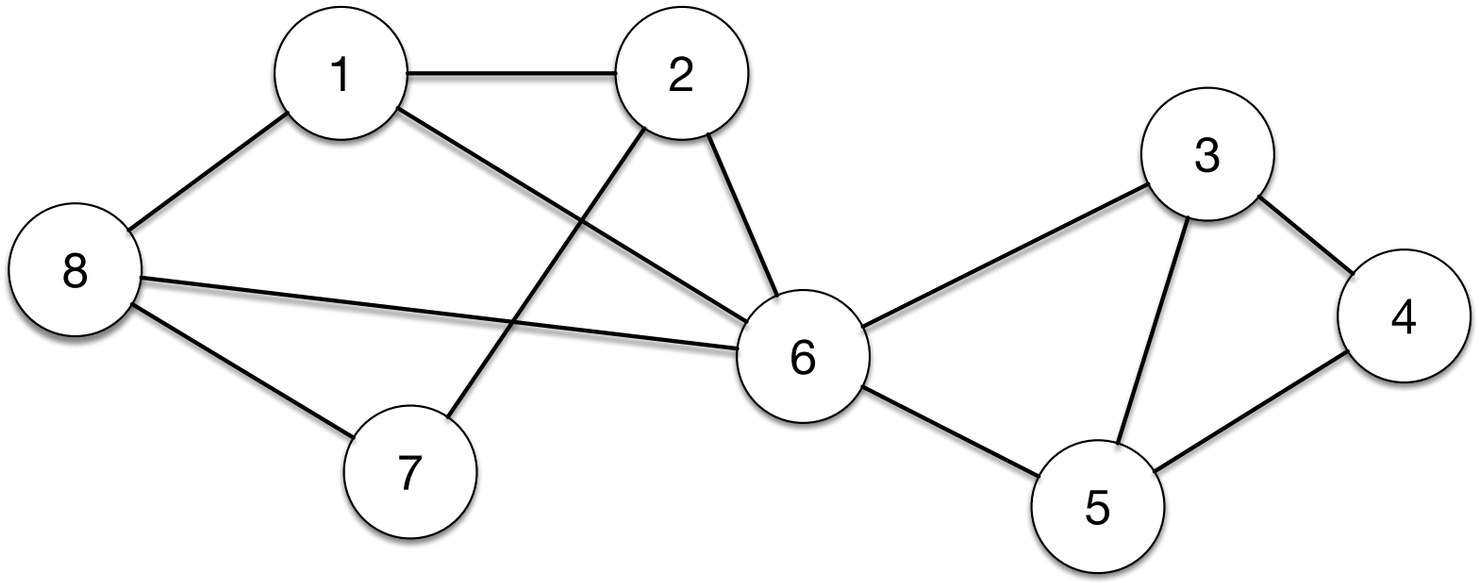
1. Link prediction problem
2. Unsupervised approach
3. Supervised approach

Link prediction

- **Link prediction.** Given a snapshot of a dynamic network at time t, predict edges added in the interval (t,t')
- **Link completion.** Given a network, infer links that are consistent with the structure, but missing
- **Link reliability.** Estimate the reliability of given links in the network
- **What to predict?**
  - Link existence
  - Link weight
  - Link type

Link prediction

Why it is challenging



- Number of missing edges = $|V| (|V| - 1)/2 - |E|$
- In sparse graphs, $|E| << |V|^2$
- Probability of correct random guess $O(1/|V|^2)$

Scoring algorithm

- Link prediction by proximity scoring

  1. For each pair of nodes compute proximity score c(v,v')
  2. Sort all pairs by the decreasing score
  3. Select top n pairs (or above some threshold) as new links

- Many metrics have been summarised in :

David Liben-Nowell and Jon Kleinberg. 2007. The link-prediction problem for social networks. *J. Am. Soc. Inf. Sci. Technol.* 58, 7 (May 2007), 1019-1031.

Scoring functions

- Based on the local neighbourhood of v_i and v_j
  - Number of common neighbours

$$|\mathcal{N}(v_i) \cap \mathcal{N}(v_j)|$$

  - Jaccard's coefficient

$$\frac{|\mathcal{N}(v_i) \cap \mathcal{N}(v_j)|}{|\mathcal{N}(v_i) \cup \mathcal{N}(v_j)|}$$

  - Adamic / Adar

$$\sum_{v \in \mathcal{N}(v_i) \cap \mathcal{N}(v_i)} \frac{1}{log|\mathcal{N}(v)|}$$

Scoring functions

- Based on paths and ensemble of paths between v_i and v_j
  - Shortest path

$$-min\{path_{ij} > 0\}$$

  - Katz score : sums over all possible paths between i and j, giving higher weight to shorter paths

$$\sum_{l=1}^{\infty} \beta^{(l)} |paths_{ij}^{(l)}|$$

- Personalized (rooted) PageRank

$$PR = \alpha(D^{-1}A)^T PR + (1 - \alpha)$$

Scoring functions

- Consider a random walk which starts at x and iteratively moves to a neighbor of x chosen uniformly at random from the neighbors of x.
  - Hitting time: $-H_{ij}$ (the expected # of steps for the RW from i to j)
  - Commute time: $-(H_{ij} + H_{ji})$ (# of steps from I to j then back to i)
  - Normalized hitting / commute time:

$$-(H_{ij}\pi_j + H_{ji}\pi_i)$$

with π_i (resp. π_j) be the stationary probability of v_i (resp. v_j)

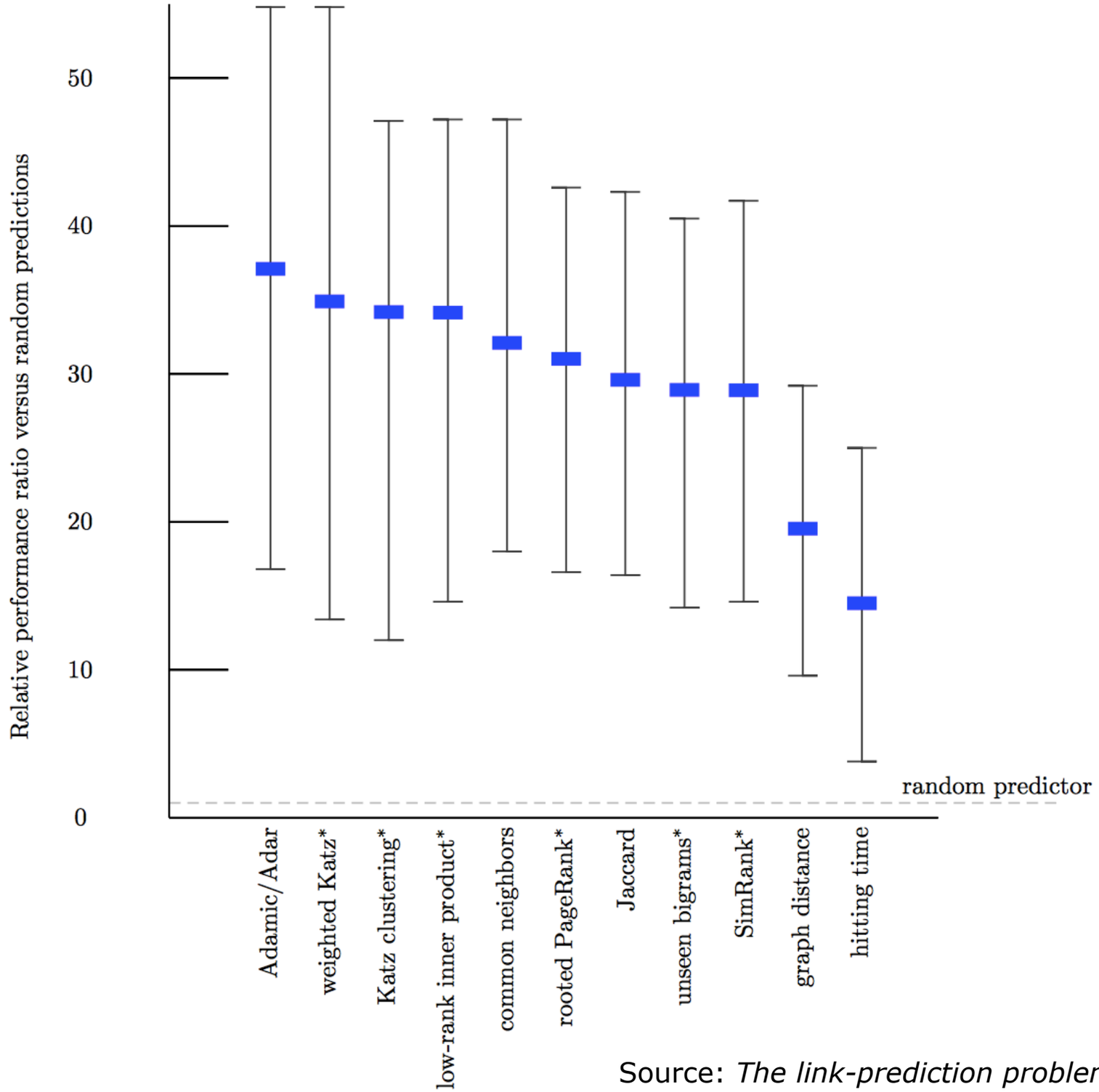
- SimRank:

$$SimRank(v_i, v_j) = \gamma . \frac{\sum_{a \in \mathcal{N}_i} \sum_{b \in \mathcal{N}_j} SimRank(a, b)}{|\mathcal{N}_i| . |\mathcal{N}_j|}$$

Scoring functions

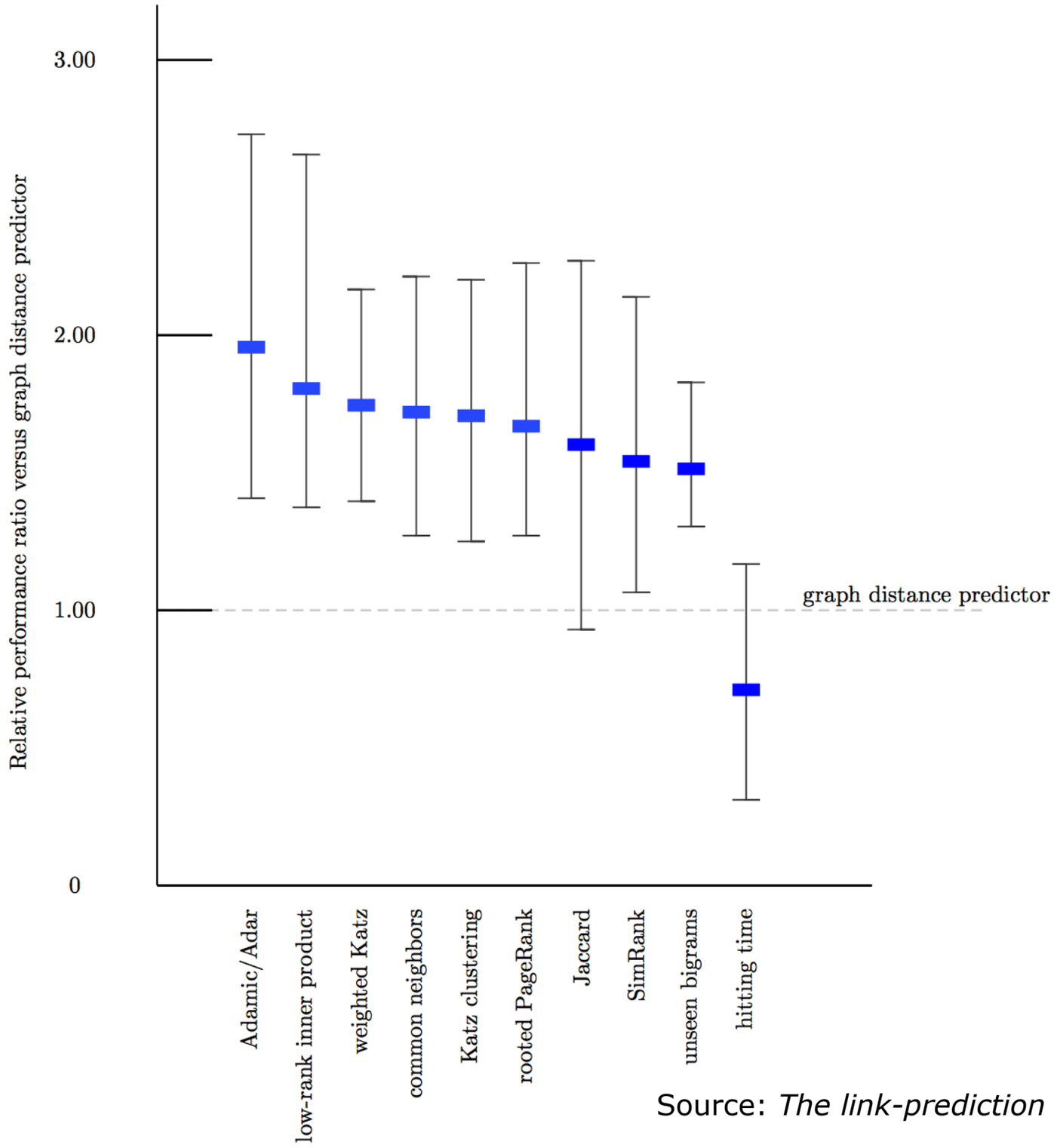
- Preferential attachment (2 alternative versions)
  - $k_i \ . \ k_j = |\mathcal{N}_i| \ . \ |\mathcal{N}_j|$
  - $k_i \ + \ k_j = |\mathcal{N}_i| \ + \ |\mathcal{N}_j|$
- Clustering coefficient (see previous lecture)
  - $CC(v_i) \ . \ CC(v_j)$
  - $CC(v_i) \ + \ CC(v_j)$

Some results



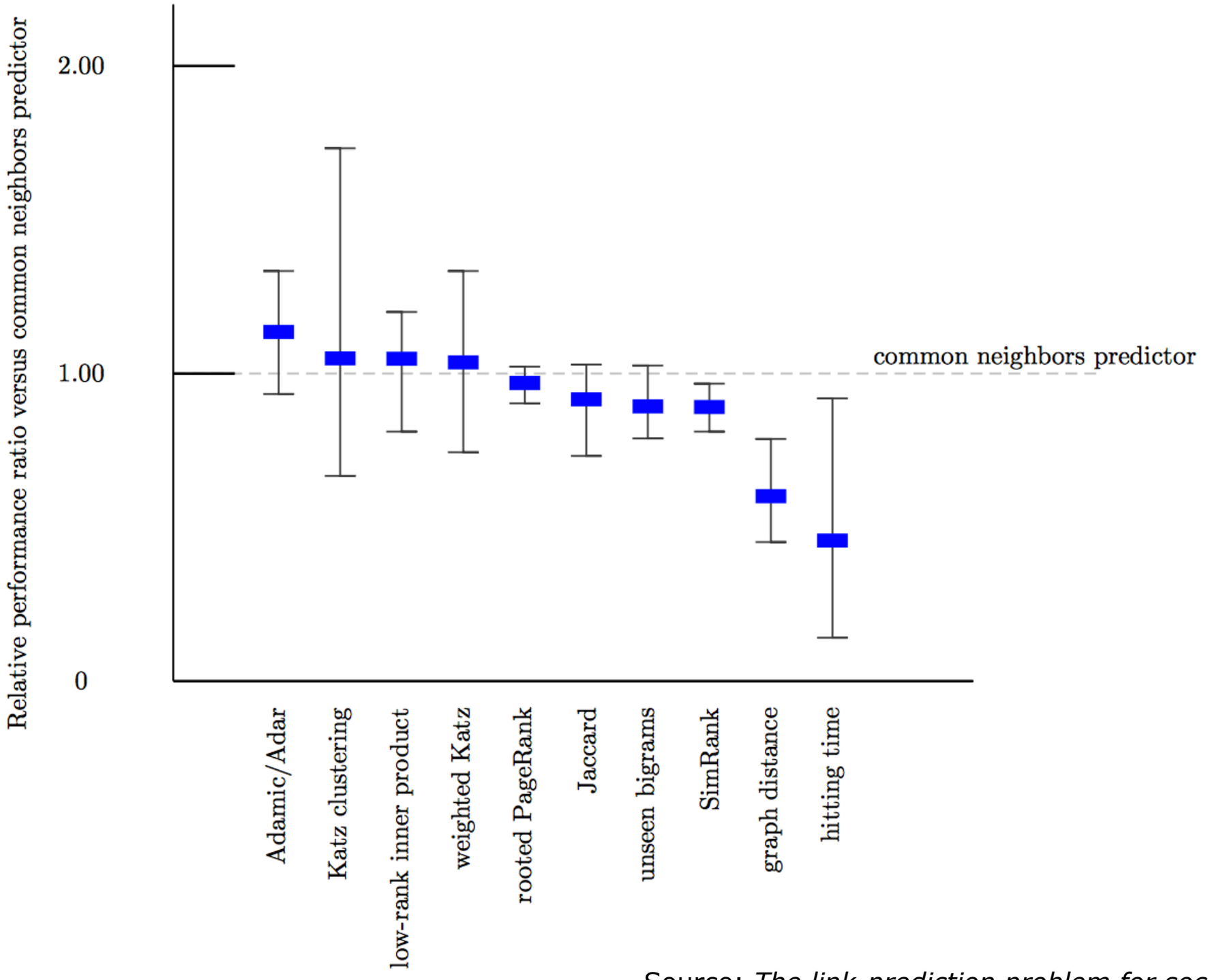
Source: *The link-prediction problem for social networks*

Some results



Source: *The link-prediction problem for social networks*

Some results



Source: *The link-prediction problem for social networks*

Take away message

- Node-based topological similarity measures (common neighbours, Jaccard, Adamic/Adar, preferential attachment) perform the best but does not scale well

- Path-based topological similarity measure (Katz, Hitting time, rooted PageRank) have to be preferred when dealing with relatively big networks (>10K vertices)

Binary classification

- A challenging classification problem:
  - A very large number of possible edges (quadratic in number of nodes)
  - Highly unbalanced class distribution
    - Positive examples : linear growth with number of nodes
    - Negative example : quadratic growth with number of nodes

A very challenging problem

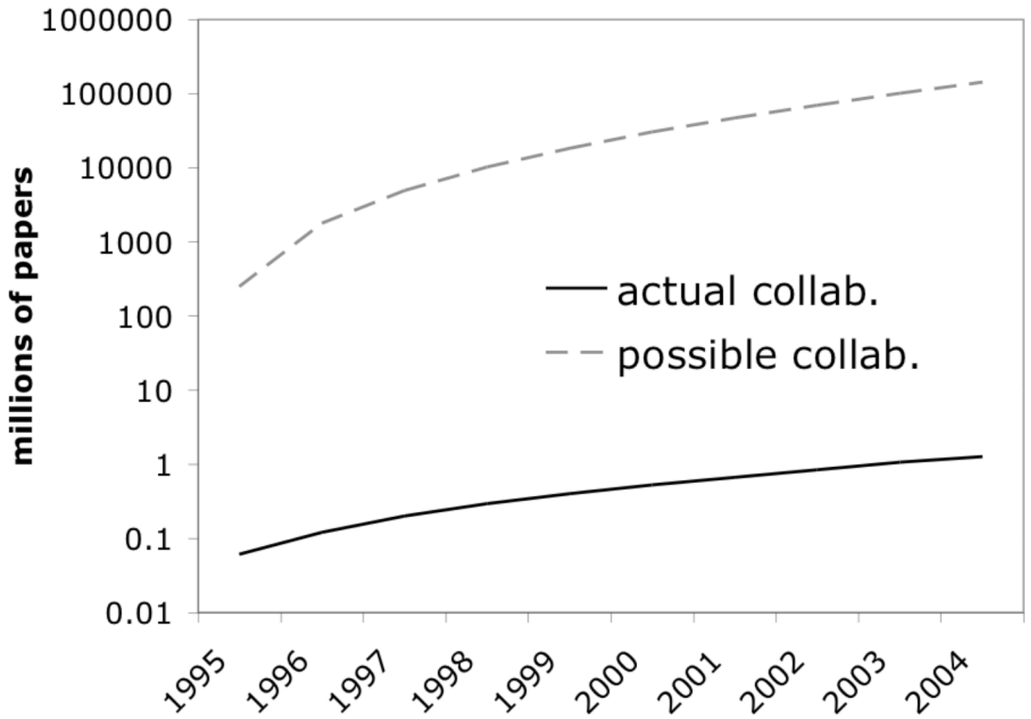


Figure 1. Logarithmic plot of actual and possible collaborations between DBLP authors, 1995-2004.

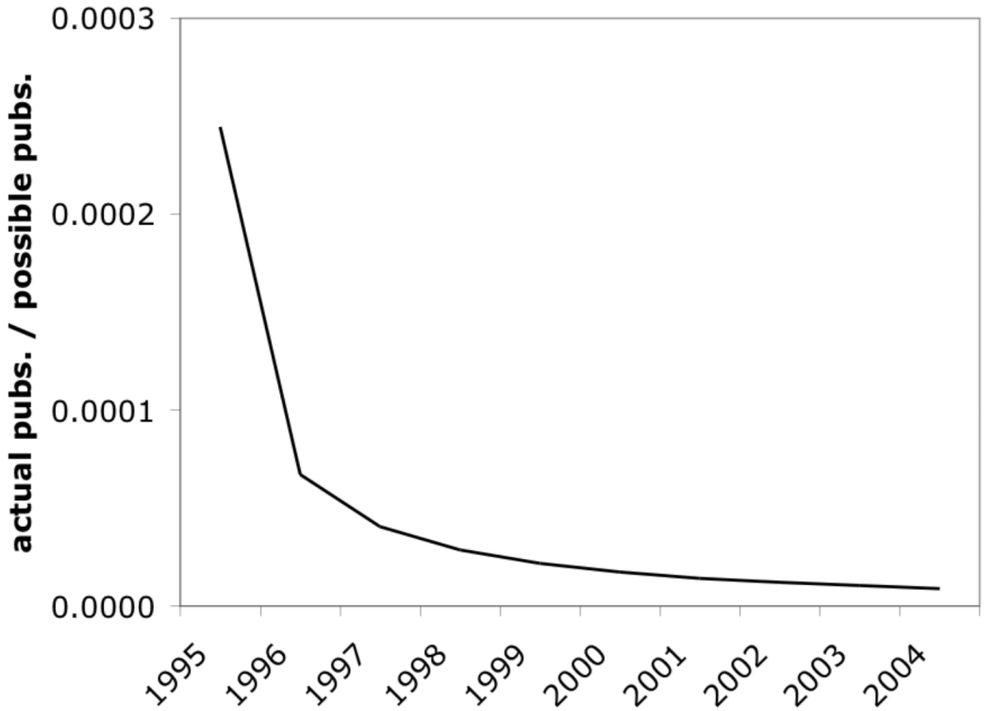


Figure 2. Publications of DBLP authors as a proportion of possible collaborations, 1995-2004.

Source: M. Rattigan, D. Jensen. The case for anomalous link discovery. ACM SIGKDD Explorations Newsletter. v 7, n 2, pp 41-47, 2005

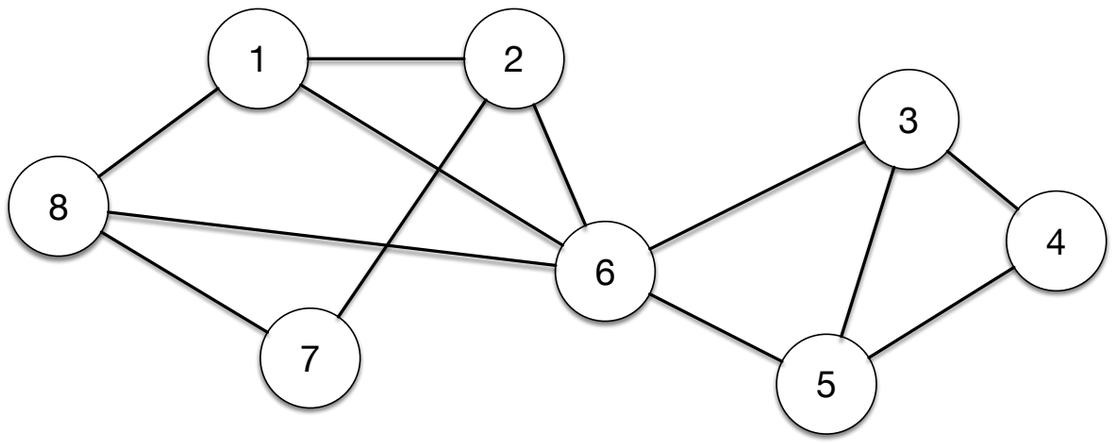
Link prediction by supervised learning

- Supervised learning process
  1. Feature generation
  2. Model training
  3. Testing

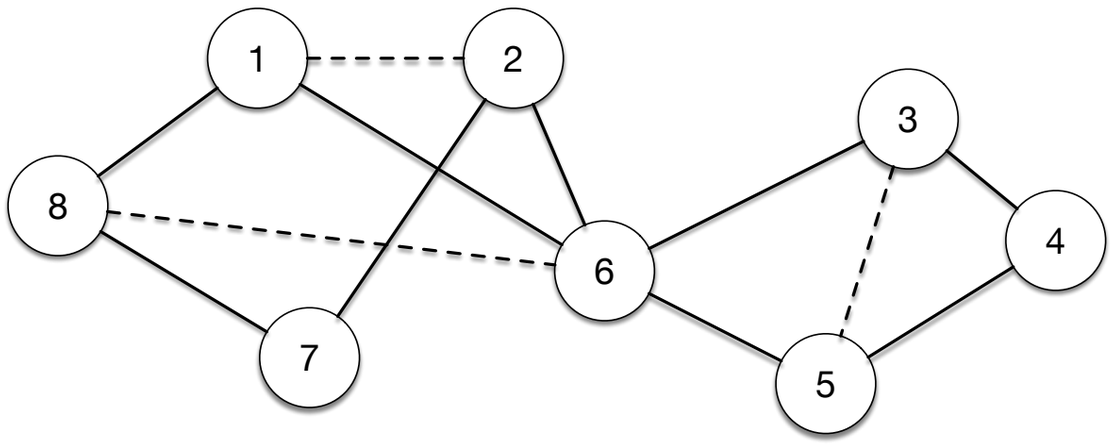
- Features
  - Topological proximity features
  - Aggregated features
  - Content based node proximity features

Evaluation

- Simple « hold out set » evaluation



Whole graph



Training graph

- More sophisticated evaluation method is obviously preferable (cross-validation)

Evaluation metrics

- Precision, recall, F-measure

$$Precision = \frac{TP}{TP + FP}, \; Recall = \frac{TP}{TP + FN}$$

$$F = \frac{2 \, . \, Precision \, . \, Recall}{Precision + Recall}$$

- True rate positive (TPR), False positive rate (FPR), ROC curve, AUC

$$TPR = \frac{TP}{TP + FN}, \; FPR = \frac{FP}{FP + TN}$$