Discrete Mathematics and Its Applications

Lecture 7: Graphs: Proximity

MING GAO

DaSE@ECNU
(for course related communications)
mgao@sei.ecnu.edu.cn

Jan. 8, 2019

Outline

1. Community Structures

2. Node Proximity
   - Simple Approaches
   - Graph-theoretic Approaches
   - SimRank
   - Random Walk based Approaches

Community structures

Definition

Community structure indicates that the network divides naturally into groups of nodes with dense connections internally and sparser connections between groups.

Community structures

Definition

Community structure indicates that the network divides naturally into groups of nodes with dense connections internally and sparser connections between groups.

- Global community structures: clustering-based approach, spectral clustering, modularity-based approach, etc.

Community structures

Definition

Community structure indicates that the network divides naturally into groups of nodes with dense connections internally and sparser connections between groups.

- Global community structures: clustering-based approach, spectral clustering, modularity-based approach, etc.
- Local community structures: node-centric community, group-centric community
  - Traditional network: clique, quasi-clique, k-clique, k-core, etc.

Community structures

Definition

Community structure indicates that the network divides naturally into groups of nodes with dense connections internally and sparser connections between groups.

- Global community structures: clustering-based approach, spectral clustering, modularity-based approach, etc.
- Local community structures: node-centric community, group-centric community
  - Traditional network: clique, quasi-clique, k-clique, k-core, etc.
  - Bipartite network: bi-clique, quasi-bi-clique, etc.

Community structures

Definition

Community structure indicates that the network divides naturally into groups of nodes with dense connections internally and sparser connections between groups.

- Global community structures: clustering-based approach, spectral clustering, modularity-based approach, etc.
- Local community structures: node-centric community, group-centric community
  - Traditional network: clique, quasi-clique, k-clique, k-core, etc.
  - Bipartite network: bi-clique, quasi-bi-clique, etc.
  - Signed network: antagonistic community, quasi-antagonistic community, etc.

Global community structures

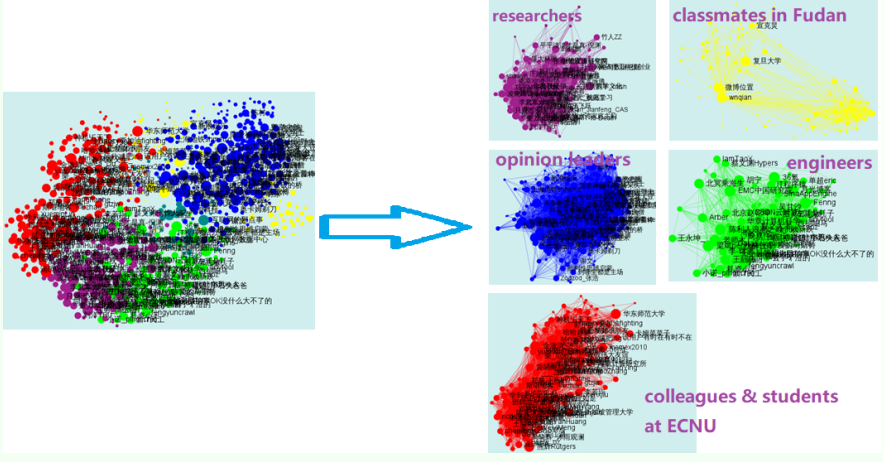
Goal

Partition nodes of a network into disjoint sets.

Global community structures

Goal

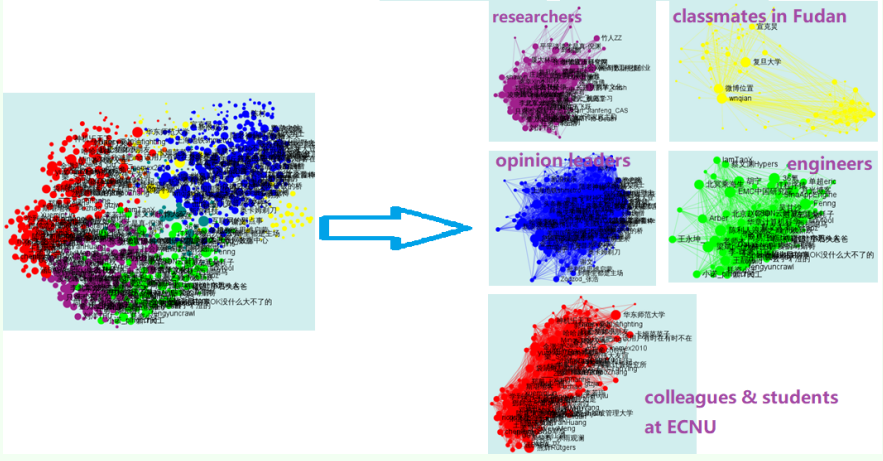
Partition nodes of a network into disjoint sets.



Global community structures

Goal

Partition nodes of a network into disjoint sets.

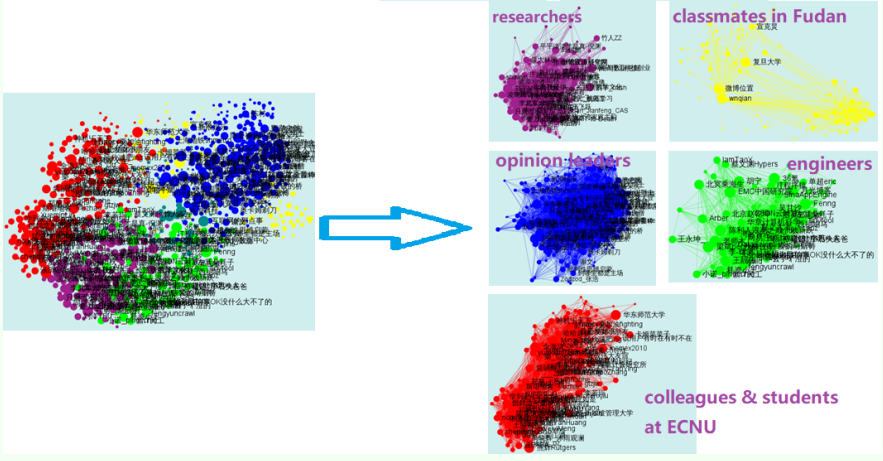


- Clustering based on vertex similarity

Global community structures

Goal

Partition nodes of a network into disjoint sets.

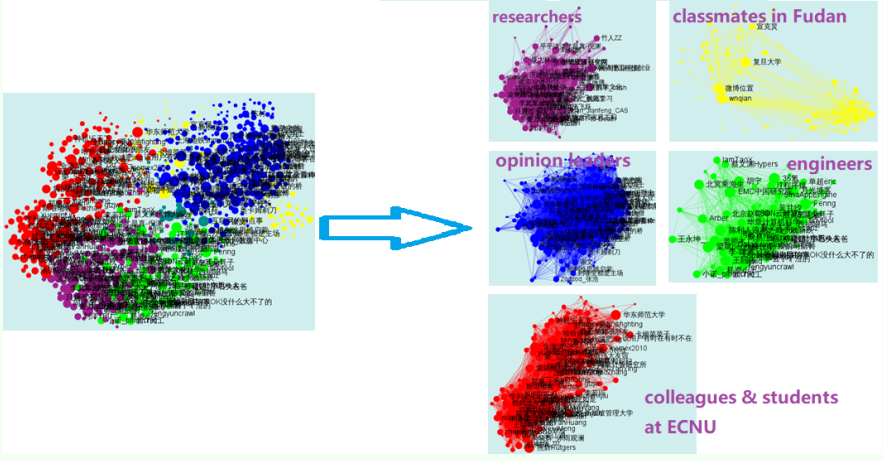


- Clustering based on vertex similarity
- Latent space models

Global community structures

Goal

Partition nodes of a network into disjoint sets.

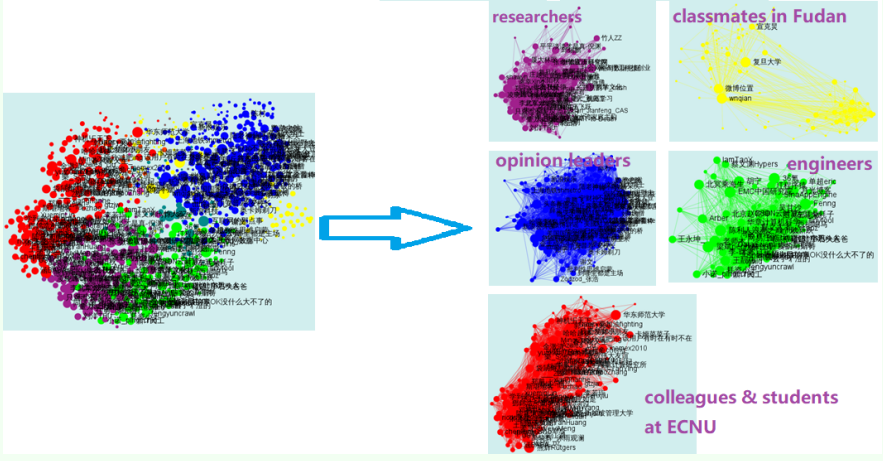


- Clustering based on vertex similarity
- Latent space models
- Spectral clustering

Global community structures

Goal

Partition nodes of a network into disjoint sets.

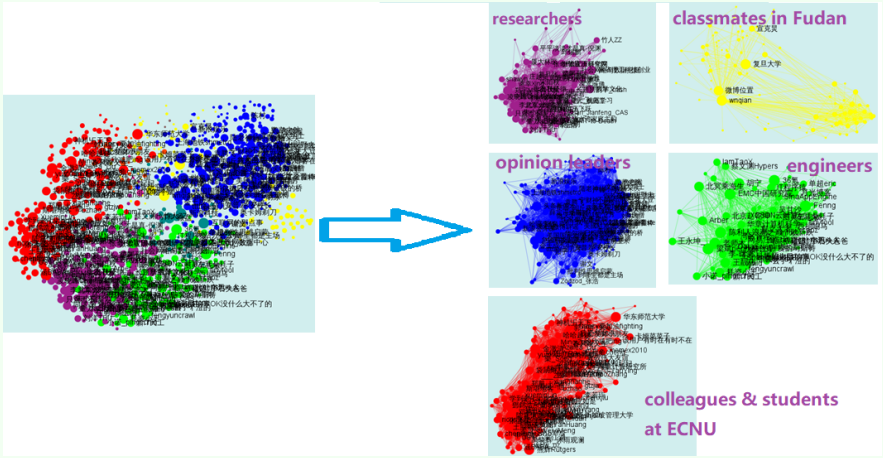


- Clustering based on vertex similarity
- Latent space models
- Spectral clustering
- Modularity maximization

Global community structures

Goal

Partition nodes of a network into disjoint sets.



- Clustering based on vertex similarity
- Latent space models
- Spectral clustering
- Modularity maximization

In the study of complex networks, a network is said to have community structure if the nodes of the network can be easily grouped into (potentially overlapping) sets of nodes such that each set of nodes is densely connected internally.

Clustering based on vertex similarity

K-means

- $v_1, \cdots, v_n$ are vertices of a graph;

Clustering based on vertex similarity

K-means

- $v_1, \cdots, v_n$ are vertices of a graph;
- Each vertex v_i will be assigned to one and only one cluster;

Clustering based on vertex similarity

K-means

- $v_1, \cdots, v_n$ are vertices of a graph;
- Each vertex v_i will be assigned to one and only one cluster;
- $C(i)$ denotes cluster number for vertex v_i;

Clustering based on vertex similarity

K-means

- $v_1, \cdots, v_n$ are vertices of a graph;
- Each vertex v_i will be assigned to one and only one cluster;
- $C(i)$ denotes cluster number for vertex v_i;
- Similarity measure or dissimilarity measure: Euclidean distance metric or Jaccard coefficient;

Clustering based on vertex similarity

K-means

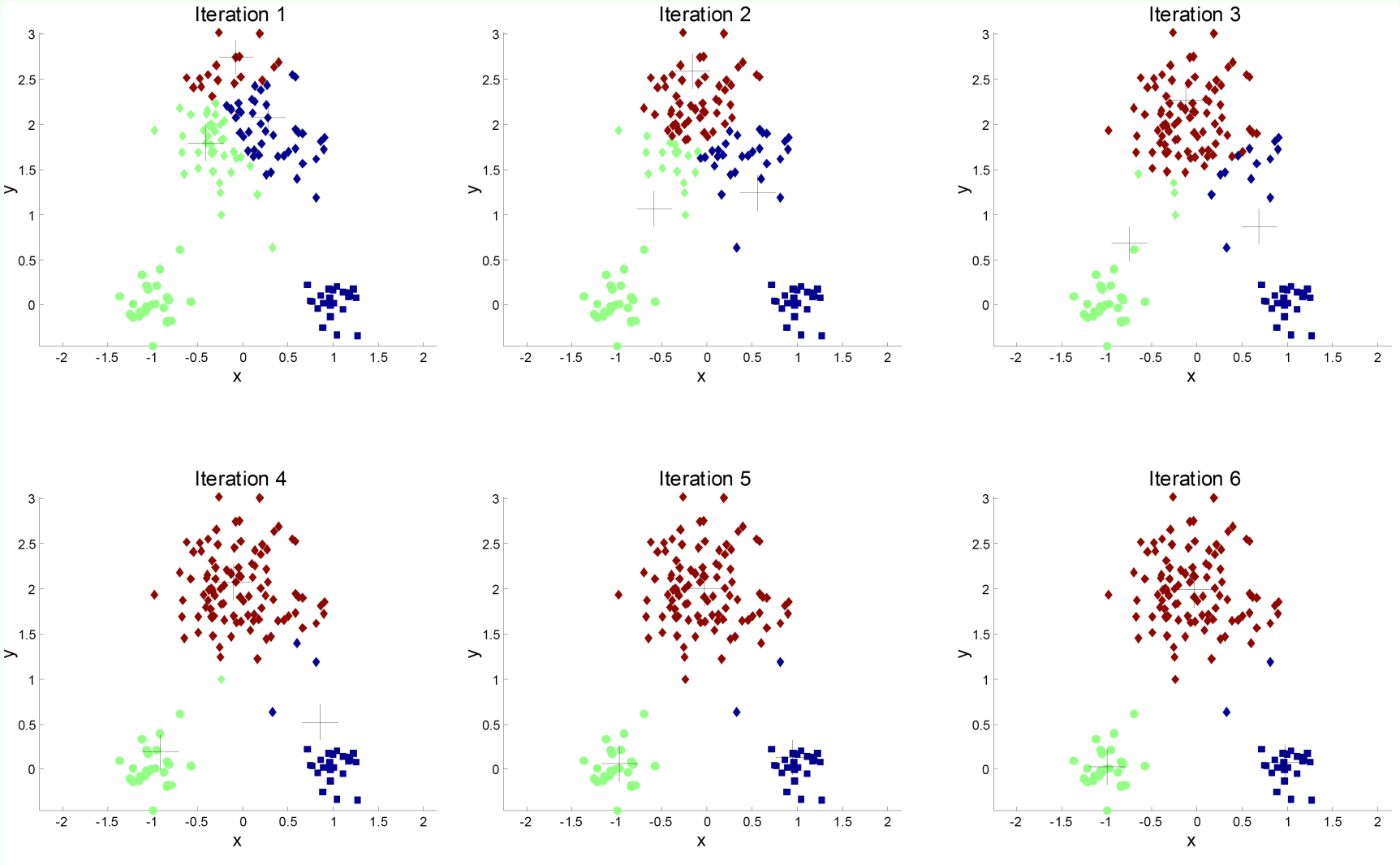
- $v_1, \cdots, v_n$ are vertices of a graph;
- Each vertex v_i will be assigned to one and only one cluster;
- $C(i)$ denotes cluster number for vertex v_i;
- Similarity measure or dissimilarity measure: Euclidean distance metric or Jaccard coefficient;
- K-means minimizes within-cluster point scatter:

$$W(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(j)=k} \| x_i - x_j \|^2$$

$$= \sum_{k=1}^{K} N_k \sum_{C(i)=k} \| x_i - m_k \|^2,$$

where N_k is the number of vertices in $k-$th cluster

K-means example



Spectral clustering

Let $\mathcal{L}$ be normalized Laplacian of graph G, the algorithm partitions nodes into two sets (B_1, B_2) based on the eigenvector $\mathbf{v}$ corresponding to the second-smallest eigenvalue of $\mathcal{L}$. Partitioning may be done in various ways:

Spectral clustering

Let $\mathcal{L}$ be normalized Laplacian of graph G, the algorithm partitions nodes into two sets (B_1, B_2) based on the eigenvector $\mathbf{v}$ corresponding to the second-smallest eigenvalue of $\mathcal{L}$. Partitioning may be done in various ways:

- Assign all nodes whose component in $\mathbf{v}$ satisfies certain condition in B_1, and B_2 otherwise, e.g., larger than median, the sign of each entry of $\mathbf{v}$.

Spectral clustering

Let $\mathcal{L}$ be normalized Laplacian of graph G, the algorithm partitions nodes into two sets (B_1, B_2) based on the eigenvector $\mathbf{v}$ corresponding to the second-smallest eigenvalue of $\mathcal{L}$. Partitioning may be done in various ways:

- Assign all nodes whose component in $\mathbf{v}$ satisfies certain condition in B_1, and B_2 otherwise, e.g., larger than median, the sign of each entry of $\mathbf{v}$.
- The algorithm can be used for hierarchical clustering by repeatedly partitioning the subsets in this fashion.

Modularity

Idea

Graph has community structure, if it is different from random graph (not expected to have community structure for random graph).

Modularity

Idea

Graph has community structure, if it is different from random graph (not expected to have community structure for random graph).

- Modularity [Newman 2006]:

$$M = \frac{1}{2m} \sum_{i,j} (A_{ij} - \frac{d(i)d(j)}{2m}) \delta(C_i, C_j).$$

where m and C_i denote $\#$ edges and the $i-$th community in the graph.

Modularity

Idea

Graph has community structure, if it is different from random graph (not expected to have community structure for random graph).

- Modularity [Newman 2006]:

$$M = \frac{1}{2m} \sum_{i,j} (A_{ij} - \frac{d(i)d(j)}{2m}) \delta(C_i, C_j).$$

  where m and C_i denote # edges and the $i-$th community in the graph.

- Compares the number of edges within a community with the expected such number in a corresponding random graph.

Modularity

Idea

Graph has community structure, if it is different from random graph (not expected to have community structure for random graph).

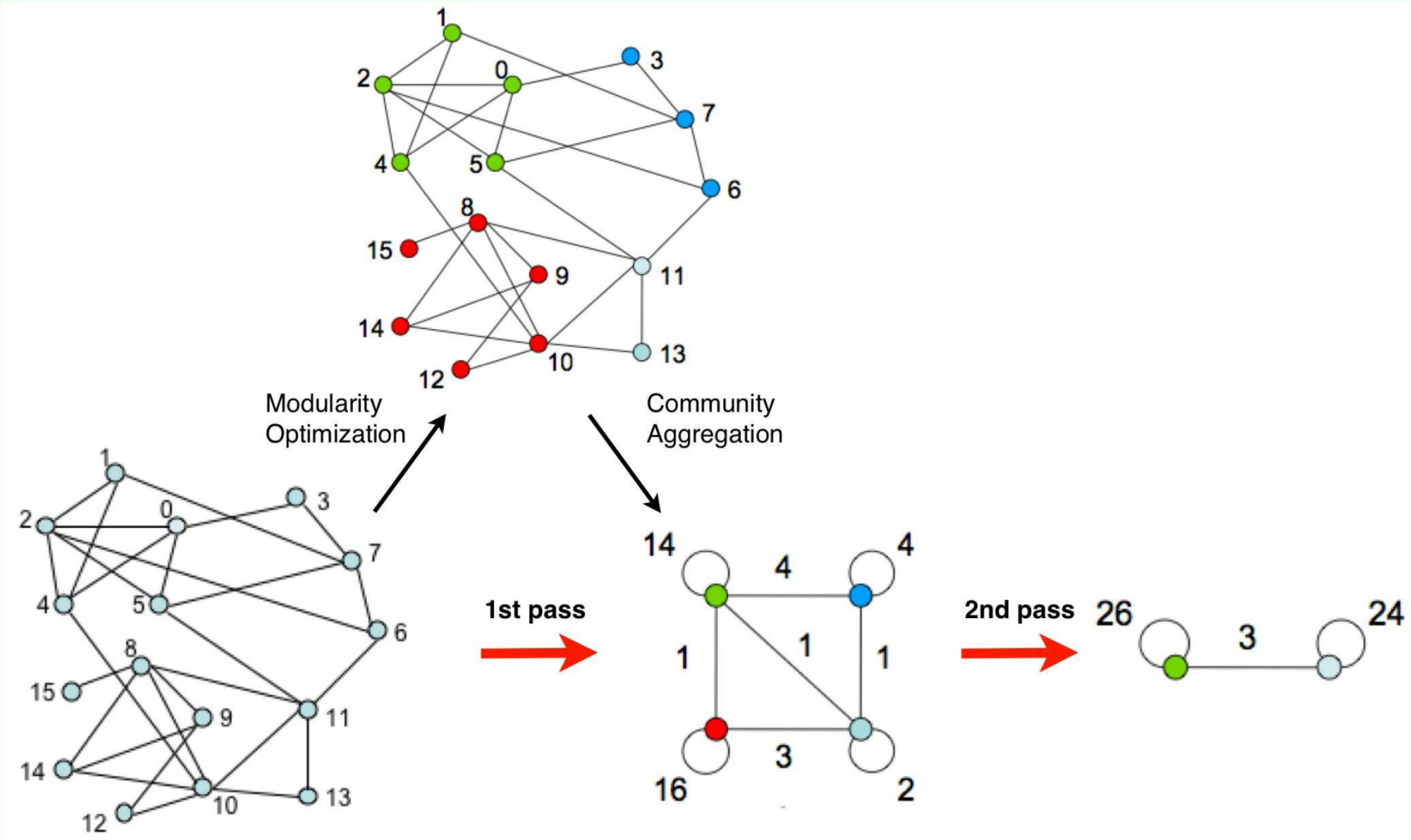
- Modularity [Newman 2006]:

$$M = \frac{1}{2m}\sum_{i,j}(A_{ij} - \frac{d(i)d(j)}{2m})\delta(C_i, C_j).$$

  where m and C_i denote # edges and the $i-$th community in the graph.

- Compares the number of edges within a community with the expected such number in a corresponding random graph.
- It can be used as a measure to evaluate the communities quality.

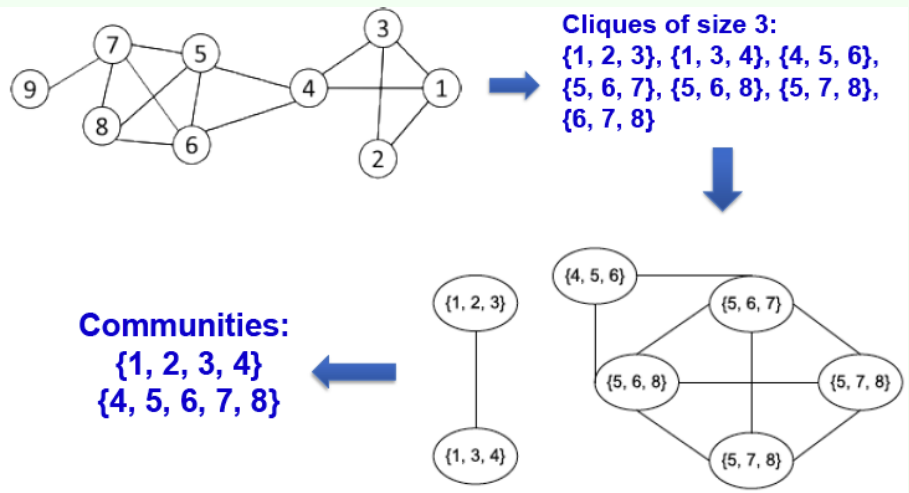
Louvain method



Clique

Definition

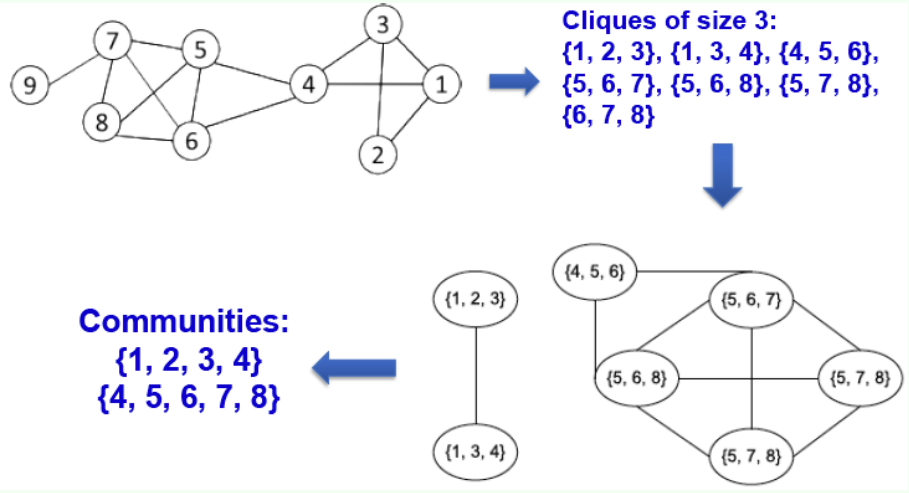
A clique is a subset of vertices of an undirected graph such that its induced subgraph is complete. A **maximal** clique is a clique that cannot be extended by including one more adjacent vertex.
Normally use cliques as a core or a seed to find larger communities.



Clique

Definition

A clique is a subset of vertices of an undirected graph such that its induced subgraph is complete. A **maximal** clique is a clique that cannot be extended by including one more adjacent vertex.
Normally use cliques as a core or a seed to find larger communities.

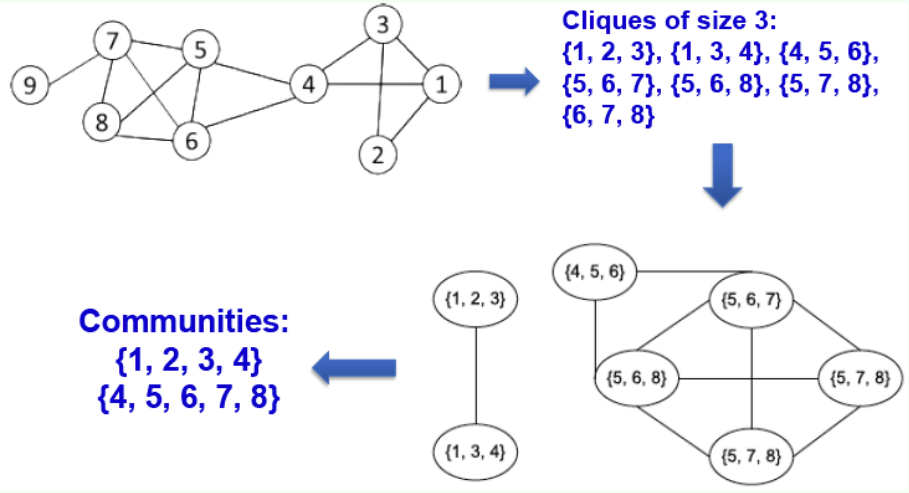


- Find out all cliques of size k in a given network (NP-complete)

Clique

Definition

A clique is a subset of vertices of an undirected graph such that its induced subgraph is complete. A **maximal** clique is a clique that cannot be extended by including one more adjacent vertex.
Normally use cliques as a core or a seed to find larger communities.

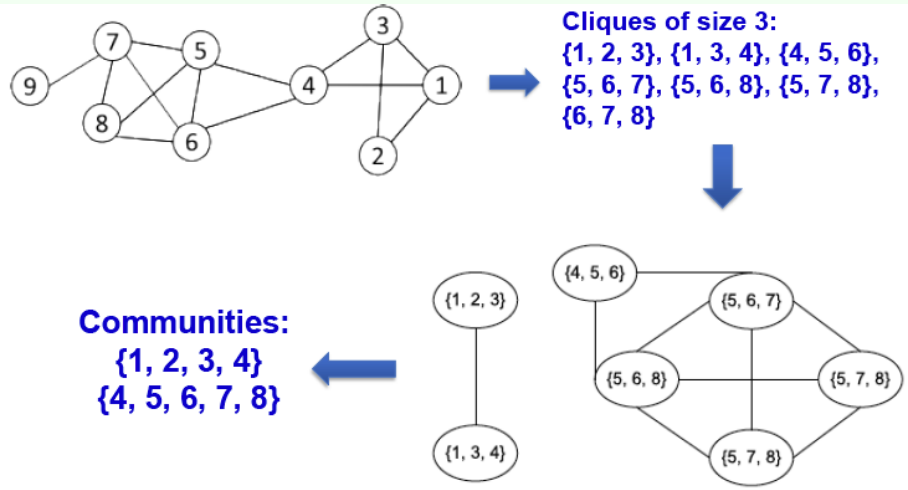


- Find out all cliques of size k in a given network (NP-complete)
- Construct a clique graph. Two cliques are adjacent if they share $k-1$ nodes

Clique

Definition

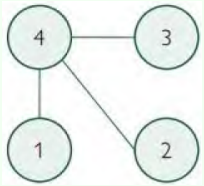
A clique is a subset of vertices of an undirected graph such that its induced subgraph is complete. A **maximal** clique is a clique that cannot be extended by including one more adjacent vertex.
Normally use cliques as a core or a seed to find larger communities.



- Find out all cliques of size k in a given network (NP-complete)
- Construct a clique graph. Two cliques are adjacent if they share $k-1$ nodes
- Each connected component in the clique graph forms a community

Extensions of clique

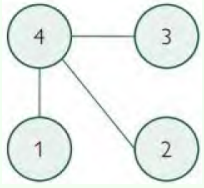
$k-$clique



Maximal subgroup, where the largest geodesic distance between any pair of nodes is not greater than k. It is a clique if $k = 1$.

Extensions of clique

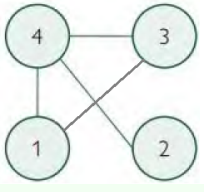
$k-$clique



Maximal subgroup, where the largest geodesic distance between any pair of nodes is not greater than k. It is a clique if $k = 1$.

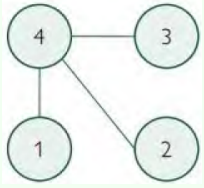
Quasi-clique

Generalize clique to dense subgraph with different definitions (degree, density).



Extensions of clique

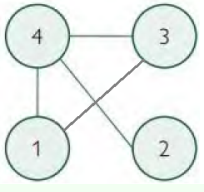
$k-$clique



Maximal subgroup, where the largest geodesic distance between any pair of nodes is not greater than k. It is a clique if $k = 1$.

Quasi-clique

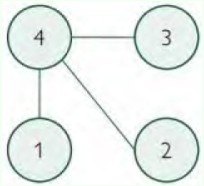
Generalize clique to dense subgraph with different definitions (degree, density).



- Node degree: every node in induced subgraph is adjacent to at least $\gamma(n-1)$ other nodes.

Extensions of clique

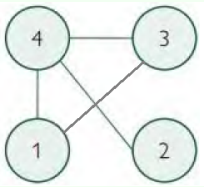
$k-$clique



Maximal subgroup, where the largest geodesic distance between any pair of nodes is not greater than k. It is a clique if $k = 1$.

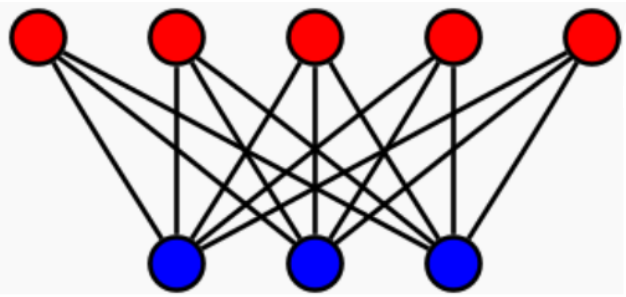
Quasi-clique

Generalize clique to dense subgraph with different definitions (degree, density).



- Node degree: every node in induced subgraph is adjacent to at least $\gamma(n-1)$ other nodes.
- Edge density: Number of edges in subgraph is at least $\gamma n(n-1)/2$, where n denotes # nodes in subgraph.

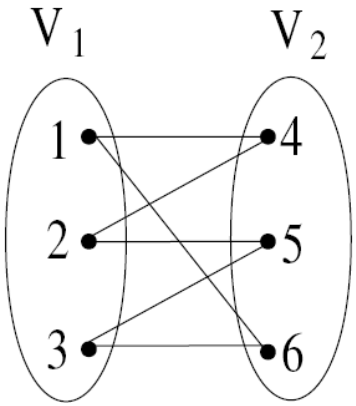
Local community structure in bipartite graph



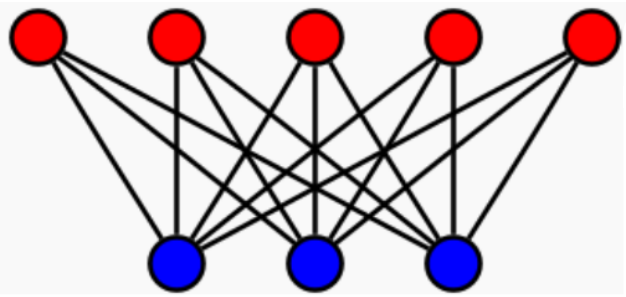
Biclique

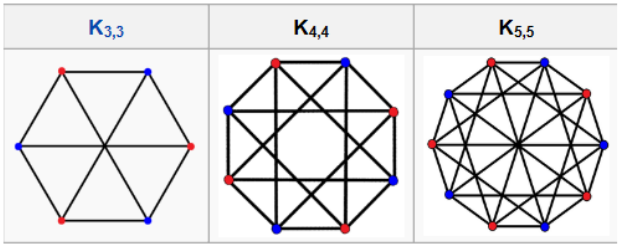
A biclique is a special kind of bipartite graph where every vertex of the first set is connected to every vertex of the second set.

- A complete bipartite graph with partitions of size $|V_1| = m$ and $|V_2| = n$, is denoted $K_{m,n}$.



Local community structure in bipartite graph

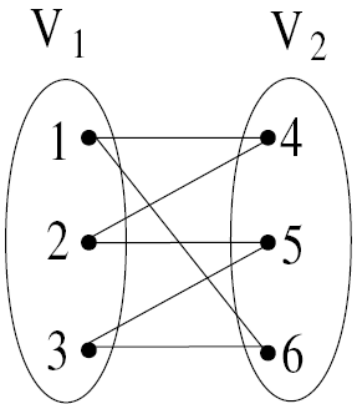




Biclique

A biclique is a special kind of bipartite graph where every vertex of the first set is connected to every vertex of the second set.

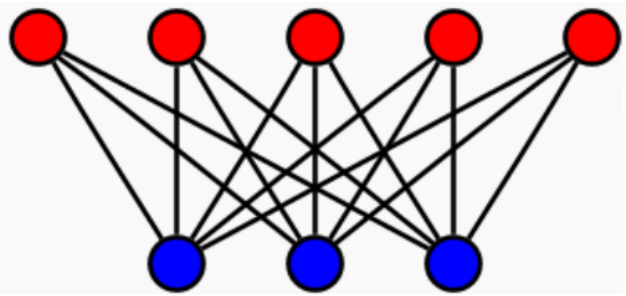
- A complete bipartite graph with partitions of size $|V_1| = m$ and $|V_2| = n$, is denoted $K_{m,n}$.

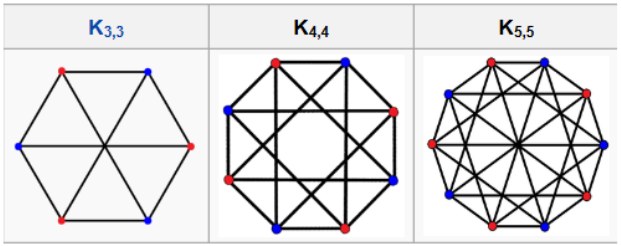


Quasi-bi-clique

The definition of biclique is too strict. A quasi-bi-clique is a dense subgraph to relax the constraint for vertices.

Local community structure in bipartite graph

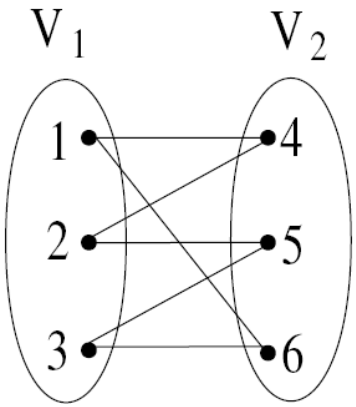




Biclique

A biclique is a special kind of bipartite graph where every vertex of the first set is connected to every vertex of the second set.

- A complete bipartite graph with partitions of size $|V_1| = m$ and $|V_2| = n$, is denoted $K_{m,n}$.

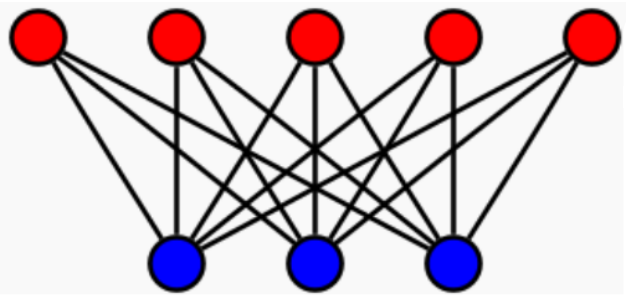


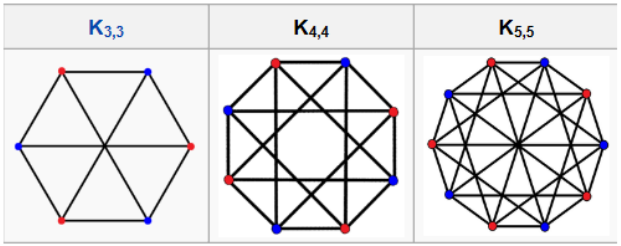
Quasi-bi-clique

The definition of biclique is too strict. A quasi-bi-clique is a dense subgraph to relax the constraint for vertices.

- Relative version of quasi-bi-clique.

Local community structure in bipartite graph

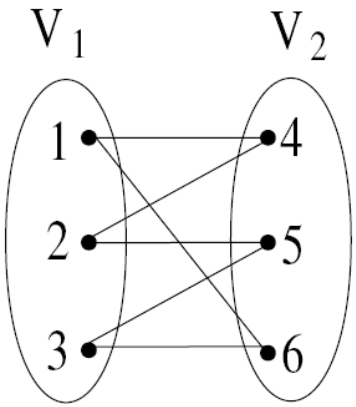




Biclique

A biclique is a special kind of bipartite graph where every vertex of the first set is connected to every vertex of the second set.

- A complete bipartite graph with partitions of size $|V_1| = m$ and $|V_2| = n$, is denoted $K_{m,n}$.



Quasi-bi-clique

The definition of biclique is too strict. A quasi-bi-clique is a dense subgraph to relax the constraint for vertices.

- Relative version of quasi-bi-clique.
- Absolute version of quasi-bi-clique.

Node proximity

Node proximity (= similarity or closeness, but $\neq$ distance) measures:

- Information exchange
- Latency/speed of information exchange
- Likelihood of future link
- Propagation of a product/idea/service/ disease
- Relevance: ranking

Node proximity

Node proximity (= similarity or closeness, but $\neq$ distance) measures:

- Information exchange
- Latency/speed of information exchange
- Likelihood of future link
- Propagation of a product/idea/service/ disease
- Relevance: ranking

Approaches

- Simple approaches
- Graph-theoretic approaches
- SimRank
- Random walk with restarts

Outline

Simple approaches

Similarity metrics

Given a graph $G = (V, E)$, $N(v_i)$ denotes the neighbors of node v_i.

- Common neighbors: $|N(v_i) \cap N(v_j)|$.
- Jaccard coefficient: $\frac{|N(v_i) \cap N(v_j)|}{|N(v_i) \cup N(v_j)|}$.
- Adamic/Adar: $\sum_{v \in N(v_i) \cap N(v_j)} \frac{1}{\log |N(v)|}$.
- Preferential attachment: $|N(v_i)| \times |N(v_j)|$.

Simple approaches

Similarity metrics

Given a graph $G = (V, E)$, $N(v_i)$ denotes the neighbors of node v_i.

- Common neighbors: $|N(v_i) \cap N(v_j)|$.
- Jaccard coefficient: $\frac{|N(v_i) \cap N(v_j)|}{|N(v_i) \cup N(v_j)|}$.
- Adamic/Adar: $\sum_{v \in N(v_i) \cap N(v_j)} \frac{1}{\log |N(v)|}$.
- Preferential attachment: $|N(v_i)| \times |N(v_j)|$.

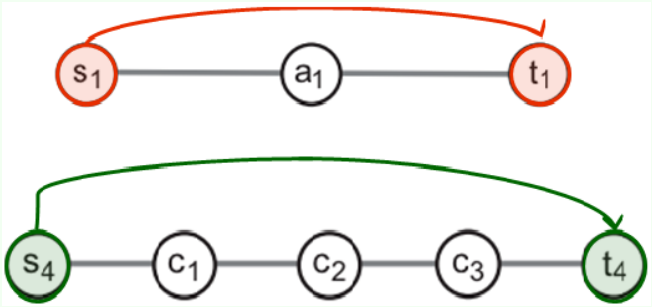
Drawbacks

- Jaccard coefficient treats a graph as a set of transactions which are independent.
- Thus, it loss the topological information of a graph.

Outline

Graph-theoretic approaches

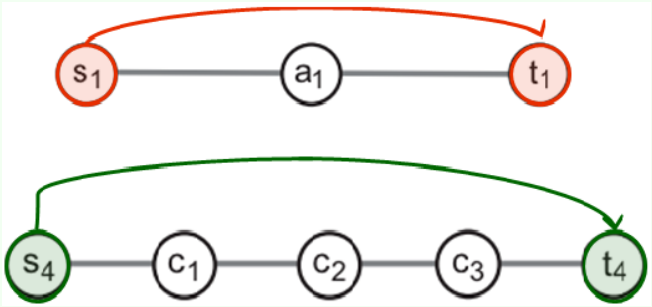
Idea



- (s_1, t_1) is more similar than (s_4, t_4).
- Simple metrics
  - Number of hops
  - Sum of weights of hops

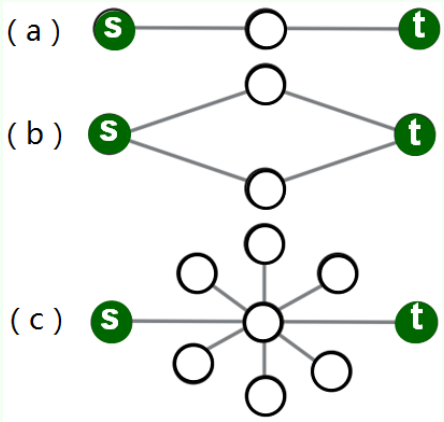
Graph-theoretic approaches

Idea



- (s_1, t_1) is more similar than (s_4, t_4).
- Simple metrics
  - Number of hops
  - Sum of weights of hops

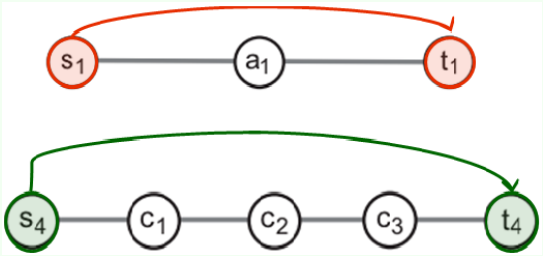
Drawbacks



- (s, t) in (b) more similar than in (a) and (c) because of linked via more paths.
- In (c), s and t are probably unrelated since (s, t) linked via high degree node.

Graph-theoretic approaches cont.

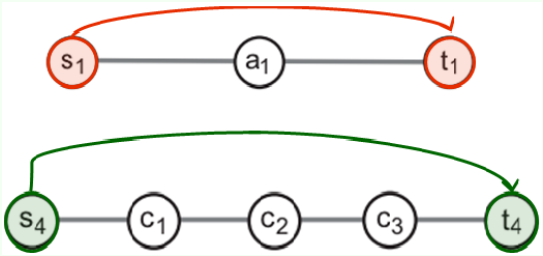
Max-flow approach



- Heavy weighted links and number of paths matter, but path length doesn't.
- The max flow of (s_1, t_1) is the same to that of (s_4, t_4).

Graph-theoretic approaches cont.

Max-flow approach



- Heavy weighted links and number of paths matter, but path length doesn't.
- The max flow of (s_1, t_1) is the same to that of (s_4, t_4).

Katz

$$K = \sum_{l \geq 1} \beta^l |\text{paths}_{v_i, v_j}^{<l>}|,$$

where $\beta \in (0, 1)$ is a pre-defined parameter.

- $\text{paths}_{v_i, v_j}^{<l>}$ is the set of exact l-length paths from v_i to v_j.
- $|\text{paths}_{v_i, v_j}^{<l>}| = 1$ if and only if v_i and v_i are connected by a link.
- $K = \beta A + \beta^2 A^2 + \cdots = (I - \beta A)^{-1} - I$.

Outline

SimRank [G. Jeh and J. Widom KDD 2002]

Idea

Two objects are similar if they are connected to similar objects.

- Iterative computation with initial value

$$s^{(0)}(a,b) = \begin{cases} 1, & a = b; \\ 0, & a \neq b. \end{cases}$$

$$s^{(k+1)}(a,b) = \begin{cases} 1, & a = b; \\ \frac{\alpha}{|N(a)N(b)|} \sum_{c \in N(a)} \sum_{d \in N(b)} s^{(k)}(c,d), & a \neq b. \end{cases}$$

  where $\alpha \in [0,1]$ is a constant.

- However, it needs $O(n^3)$ runtime. Many works improve the runtime.
  - Simrank++ [VLDB 2008]
  - $S = C(A^T S A) + I$ with Kronecker product [EDBT 2010]
  - Parallel SimRank [VLDB 2015]

Outline

Random walk

Hitting time

Hitting time H_{v_i,v_j} is the expected number of steps required for a random walk starting at v_i to reach v_j.

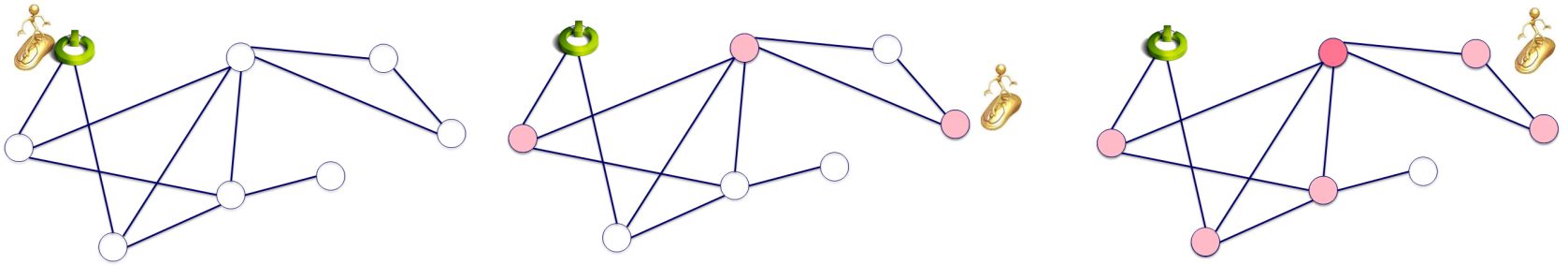
- Similar to Katz, all paths need to enumerate when hitting time is computed, but longer paths have smaller probabilities.
- $H = W + 2W^2 + \cdots = ((I - W)^{-1} - I)(I - W)^{-1}$.

Commute time

Commute time $H_{v_i,v_j} + H_{v_j,v_i}$ is the expected number of steps required for a random walk starting at v_i to reach v_j, then return to v_i.

- For an undirected graph, $H_{v_i,v_j} = H_{v_j,v_i}$.
- Similar, we can compute commute time for every pair of nodes.

Personalized random walk with restarts



Idea

$$\mathbf{r}_i = cW\mathbf{r}_i + (1-c)\mathbf{e}_i,$$

where $\mathbf{r}_i \in R^{n\times 1}$ is ranking score w.r.t. v_i, W is the transition probability matrix, and $\mathbf{e}_i \in R^{n\times 1}$ is start score w.r.t. v_i with $r_{ii} = 1$, 0 otherwise.

- $(I - cD^{-1}A)\mathbf{r}_i = (1-c)\mathbf{e}_i$.
- $\mathbf{r}_i = (1-c)(I - cD^{-1}A)^{-1}\mathbf{e}_i$.

Take-home messages

- Community detection
  - Global structure
  - Local structure
- Node proximity
  - Simple approaches
  - Graph-theoretic approaches
  - SimRank
  - Random walk based approaches