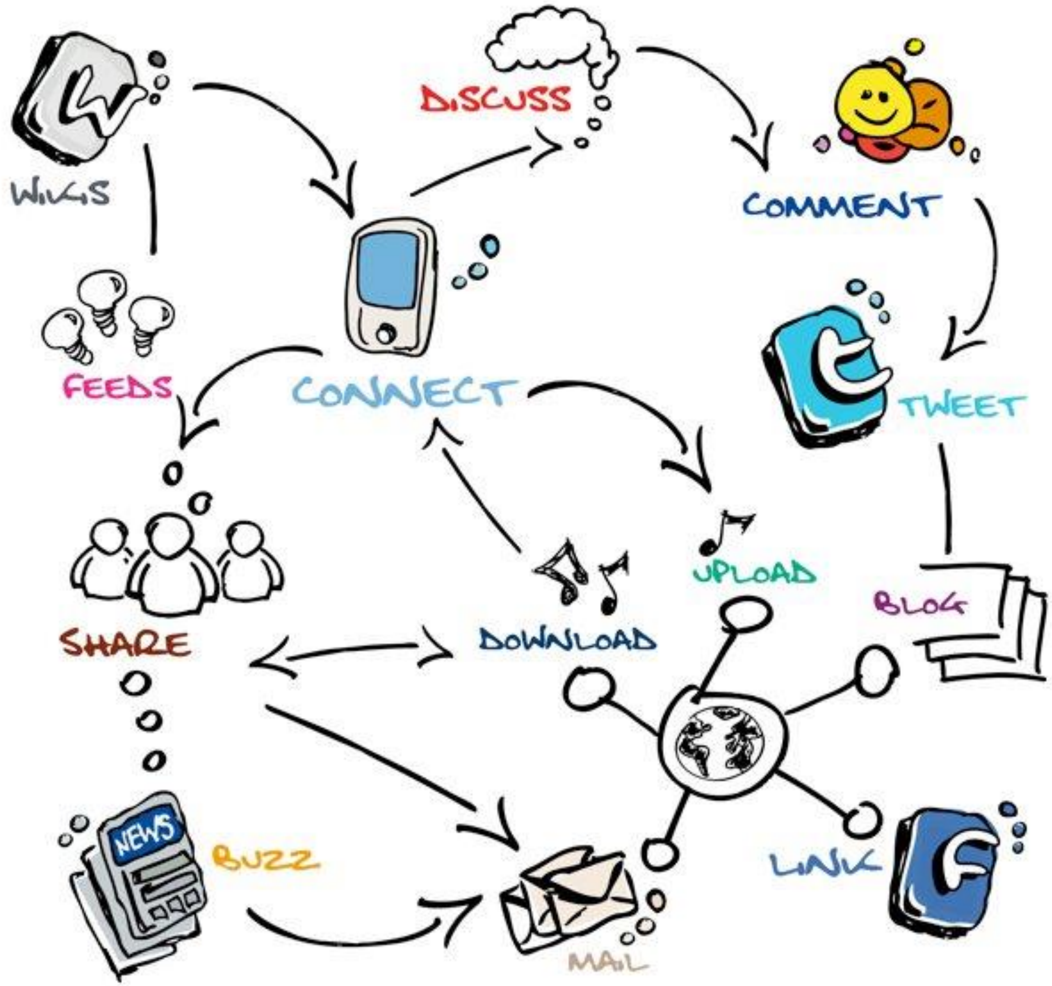


Node Similarity and Structural Roles

Ahmet Onur Durahim

How the Class Fits Together

Properties	Models	Algorithms
Small diameter, Edge clustering	Small-world model, Erdös-Renyi model	Decentralized search
Scale-free	Preferential attachment, Copying model	PageRank, Hubs and authorities
Strength of weak ties, Core-periphery	Kronecker Graphs	Community detection: Girvan-Newman, Modularity
Densification power law, Shrinking diameters	Microscopic model of evolving networks	Link prediction, Supervised random walks
Complex Graph Structure	Graph Neural Networks	Node Classification Graph Representation Learning
Information virality, Memetracking	Independent cascade model, Game theoretic model	Influence maximization, Outbreak detection, LIM

Node Equivalence, Structural Roles and Assortative Mixing

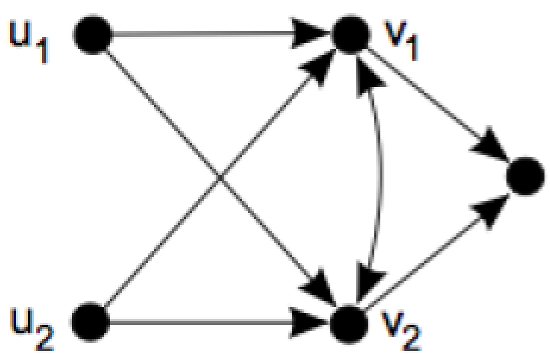
- ***Node equivalence***
  - Structural equivalence
  - Regular equivalence
- ***Node similarity***
  - Jaccard Similarity
  - Cosine Similarity
  - Pearson correlation
- ***Structural Roles***
  - Structural similarity
  - Role generalization and transfer learning
  - Making sense of roles
- ***Assortative mixing***
  - Mixing by value
  - Degree correlation

Patterns of relations

- ***Global, statistical properties of the networks***
  - average node degree (degree distribution)
  - average clustering
  - average (shortest) path length
- ***Local, per vertex properties***
  - node centrality
  - page rank
- ***Pairwise properties***
  - node equivalence
  - node similarity
  - correlation between pairs of vertices (node values)

Structural Equivalence

- ***Structural equivalence:*** two vertices are structurally equivalent if their respective sets of in-neighbors and out-neighbors are the same

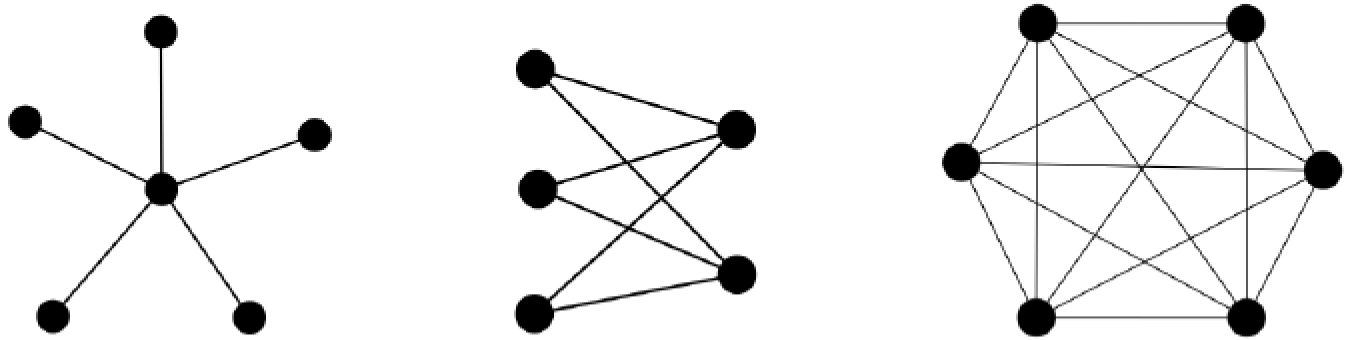


	u1	u2	v1	v2	w
u1	0	0	1	1	0
u2	0	0	1	1	0
v1	0	0	0	1	1
v2	0	0	1	0	1
w	0	0	0	0	0

rows and columns of adjacency matrix of structurally equivalent nodes are identical, “connect to the same neighbors”

Structural Equivalence

- In order for adjacent vertices to be structurally equivalent, they might have self loops
- Sometimes called "***strong structural equivalence***"
  - Sometimes ***relax requirements*** for self loops for adjacent nodes



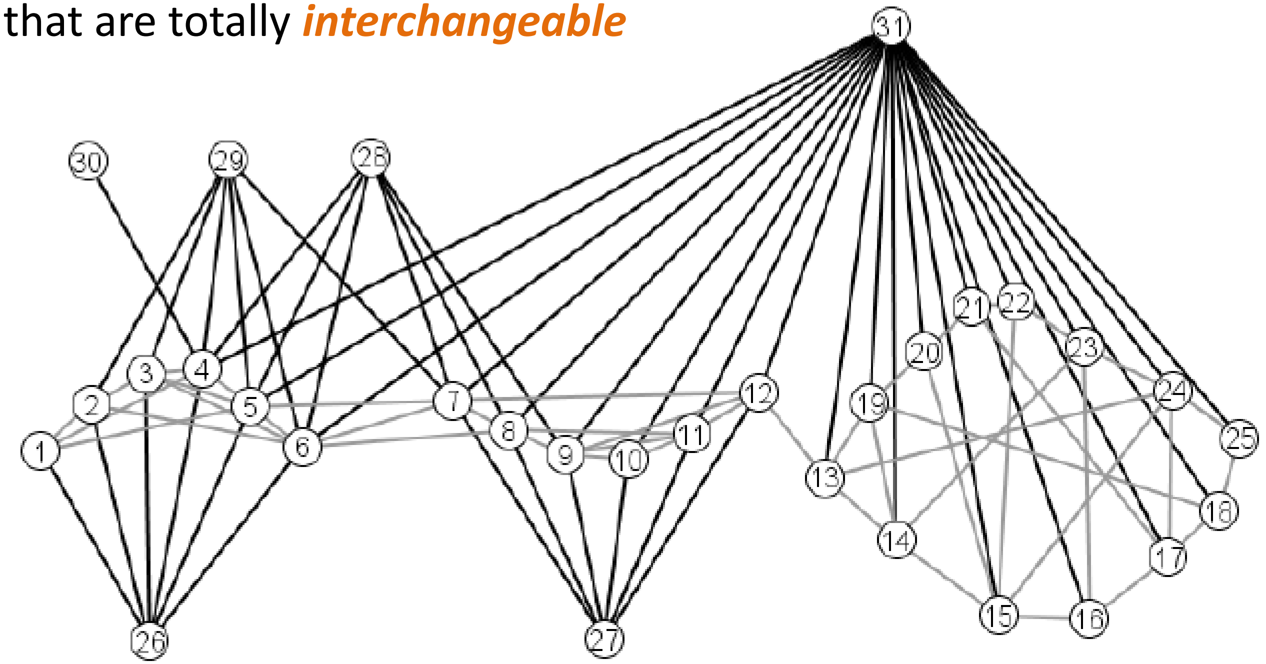
Similarity Measures

- ***Jaccard similarity***

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

SE comes from sociology

structurally equivalent nodes are the people that have positions that are totally ***interchangeable***



Similarity Measures

CS takes **weights** into accounts.
JS does not !

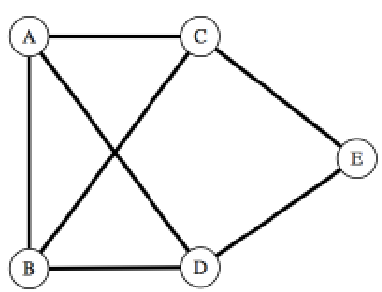
- ***Cosine similarity*** (vectors in *n-dim* space)

$$\sigma(v_i, v_j) = cos(\theta_{ij}) = \frac{\mathbf{v}_i^T \mathbf{v}_j}{|\mathbf{v}_i||\mathbf{v}_j|} = \frac{\sum_k A_{ik} A_{kj}}{\sqrt{\sum A_{ik}^2}\sqrt{\sum A_{jk}^2}}$$

- ***Pearson correlation coefficient***

think of each of the 2 nodes as 2 random variables (adjacency vectors)

$$r_{ij} = \frac{\sum_k (A_{ik} - \langle A_i \rangle)(A_{jk} - \langle A_j \rangle)}{\sqrt{\sum_k (A_{ik} - \langle A_i \rangle)^2}\sqrt{\sum_k (A_{jk} - \langle A_j \rangle)^2}}$$



0	1	0	1	1
1	0	1	0	1
0	1	0	1	0
1	0	1	0	1
1	1	0	1	0

Adjacency Matrix $\mathbf{A}_{ij}$

Similarity Measures

- ***Unweighted undirected graph*** $A_{ik} = A_{ki}$, binary matrix, only 0 and 1 (can simplify the formula)

$$k_i = \sum_k A_{ik} = \sum_k A_{ik}^2 \text{ - node degree}$$
$$n_{ij} = \sum_k A_{ik} A_{kj} = (A^2)_{ij} \text{ - number of shared neighbors}$$
$$\langle A_i \rangle = \frac{1}{n} \sum_k A_{ik}$$

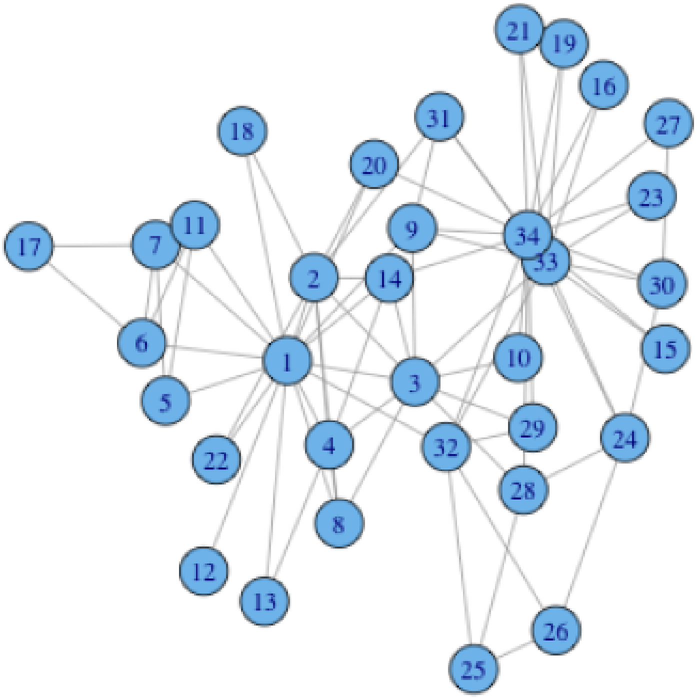
- ***Cosine similarity*** (vectors in *n-dim* space)

$$\sigma(v_i, v_j) = cos(\theta_{ij}) = \frac{n_{ij}}{\sqrt{k_i k_j}}$$

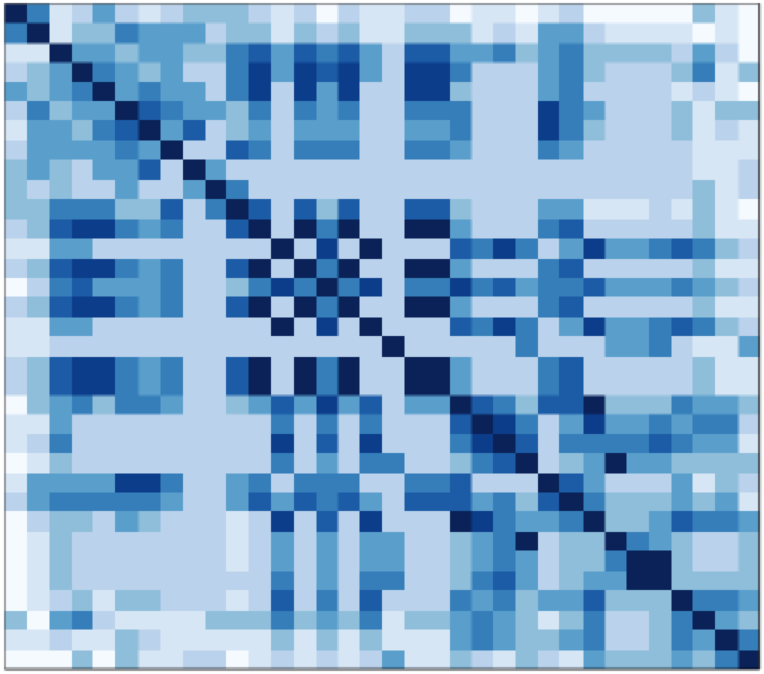
- ***Pearson correlation coefficient***:

$$r_{ij} = \frac{n_{ij} - \frac{k_i k_j}{n}}{\sqrt{k_i - \frac{k_i^2}{n}} \sqrt{k_j - \frac{k_j^2}{n}}}$$

Similarity Matrix



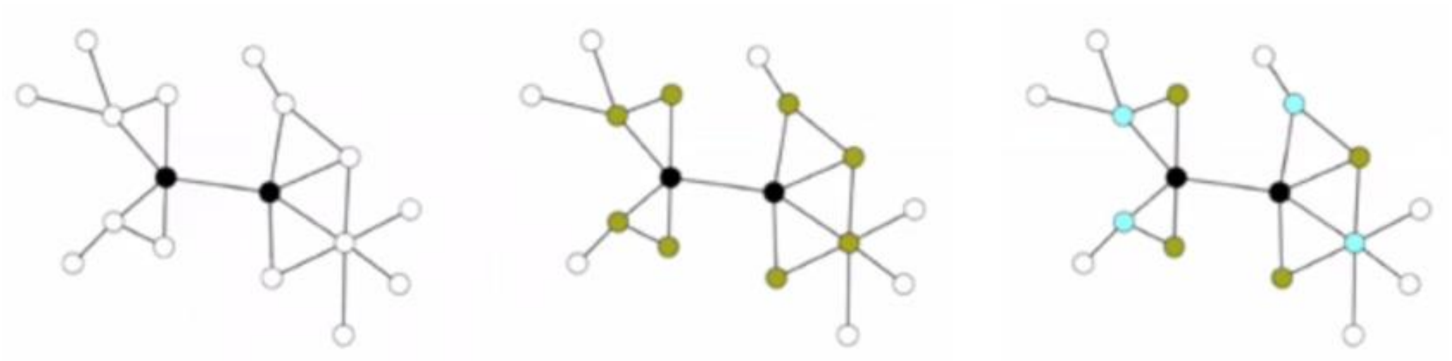
Graph



Node similarity matrix

Regular Equivalence

- ***Regular equivalence:*** two vertices are regularly equivalent if they are equally related to equivalent others

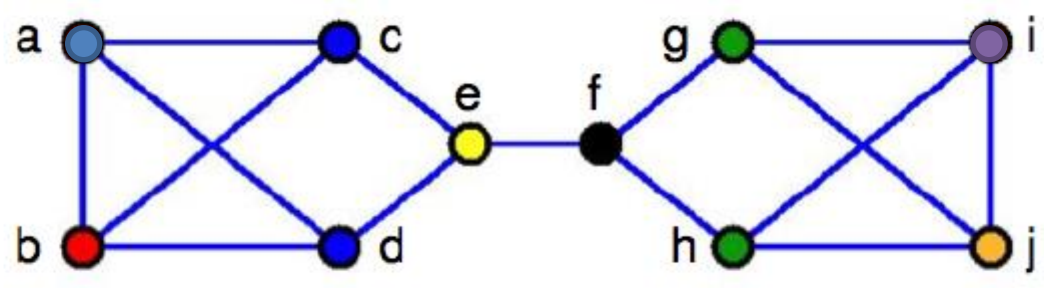


- Equivalent definition in terms of ***role assignment*** (***coloring***): vertices that are colored the same, have the same colors of their neighborhoods

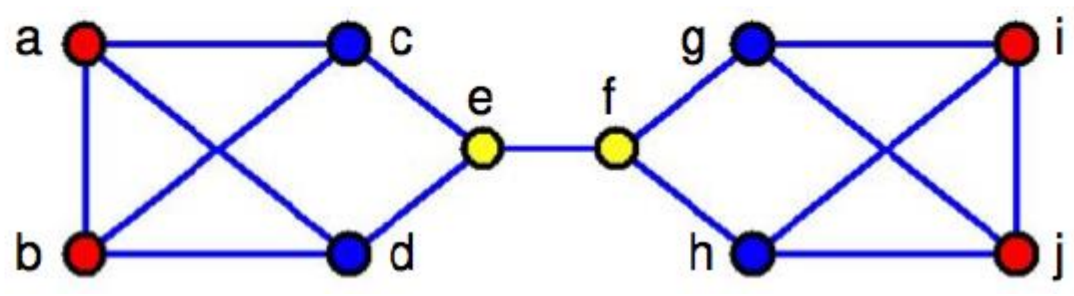
White and Reitz,1983; Everette and Borgatti, 1991

Equivalence example

- ***structural equivalence***



- ***regular equivalence***



e and f play similar roles

Regular Equivalence

- ***Recursive definition:*** two vertices are regularly equivalent if they are equally related to equivalent others.
  - Quantitative measure of regular equivalence, σ_{ij} - similarity score

$$\sigma_{ij} = \alpha \sum_{k,l} A_{ik} A_{jl} \sigma_{kl}$$

$$\boldsymbol{\sigma} = \alpha \mathbf{A} \boldsymbol{\sigma} \mathbf{A}$$

- σ_{ij} is a metric for ***pair of nodes*** (not for a single node like pagerank or katz centrality)
- α is used for normalization
- There are ***two steps*** here A_{ik} and A_{jl}

- should have high σ_{ii} - self similarity

$$\sigma_{ij} = \alpha \sum_{k,l} A_{ik} A_{jl} \sigma_{kl} + \delta_{ij}$$

δ_{ij} = 1, if i=j

δ_{ij} = 0, if i≠j

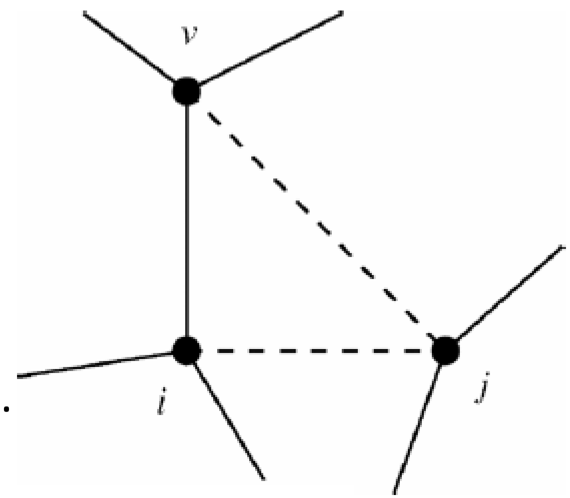
$$\boldsymbol{\sigma} = \alpha \mathbf{A} \boldsymbol{\sigma} \mathbf{A} + \mathbf{I}$$

Regular Equivalence - Revision

- A vertex j is similar to vertex i (dashed line) if i has a network neighbor v (solid line) that is itself similar to j

$$\sigma_{ij} = \alpha \sum_{v} A_{iv}\sigma_{vj} + \delta_{ij}$$

$$\boldsymbol{\sigma} = \alpha \mathbf{A}\boldsymbol{\sigma} + \mathbf{I}$$



- Closed form solution:

$$\boldsymbol{\sigma} = (\mathbf{I} - \alpha \mathbf{A})^{-1}$$

All the paths of length 1, 2, 3, … participate in the definition. On the previous definition only even numbered paths were included.

$$\mathbf{S} = \mathbf{I} + \phi\mathbf{A} + \phi^2\mathbf{A}^2 + \ldots$$

element $[\mathbf{A}^l]_{ij}$ is equal to the number of (possibly self-intersecting) network paths of length l from i to j.

[Leicht, Holme, and Newman, 2006]()

SimRank

- Find nodes (***groups of nodes***) that ***play the same/similar roles in the network***
- Values may be used as ***features to predict links btw nodes***

- **s(a, b)** – *similarity between a and b*
- I() – set of ***in-neighbours*** (constant *c ϵ (0, 1)*)

$$s(a,b) = \begin{cases} 1, \ a = b, \\ 0, \text{ if } I(a) = \emptyset \text{ or } I(b) = \emptyset, \\ \dfrac{c}{|I(a)||I(b)|} \displaystyle\sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s(I_i(a), I_j(b)), \text{ otherwise} \end{cases}$$

- Previous models are based on ***structural ideas*** like ***Katz centrality***
- This is in some way similar to ***PageRank*** (talking about ***random walk***)

- Matrix notation:

$$S_{ij} = \frac{C}{k_i k_j} \sum_{k,m} A_{ki} A_{mj} S_{km}$$

Jeh and Widom, 2002

two objects are similar if they are referenced by the similar objects

Solving SimRank

- One can find a ***solution of the system*** defined by SimRank by using the following ***iterative process***:

$$R_0(a,b) = \begin{cases} 1, & a = b, \\ 0, & \text{otherwise}, \end{cases}$$

$$R_{k+1}(a,b) = \begin{cases} 1, \ a = b, \\ 0, \ \text{if } I(a) = \emptyset \text{ or } I(b) = \emptyset, \\ \dfrac{c}{|I(a)||I(b)|} \displaystyle\sum_{v \in I(a)} \sum_{w \in I(b)} R_k(w,v), \ \text{otherwise}. \end{cases}$$

- It is shown in [Jeh and Widom]() that ***R_k(a, b) converges to s(a, b)***

Example from SimRank Article

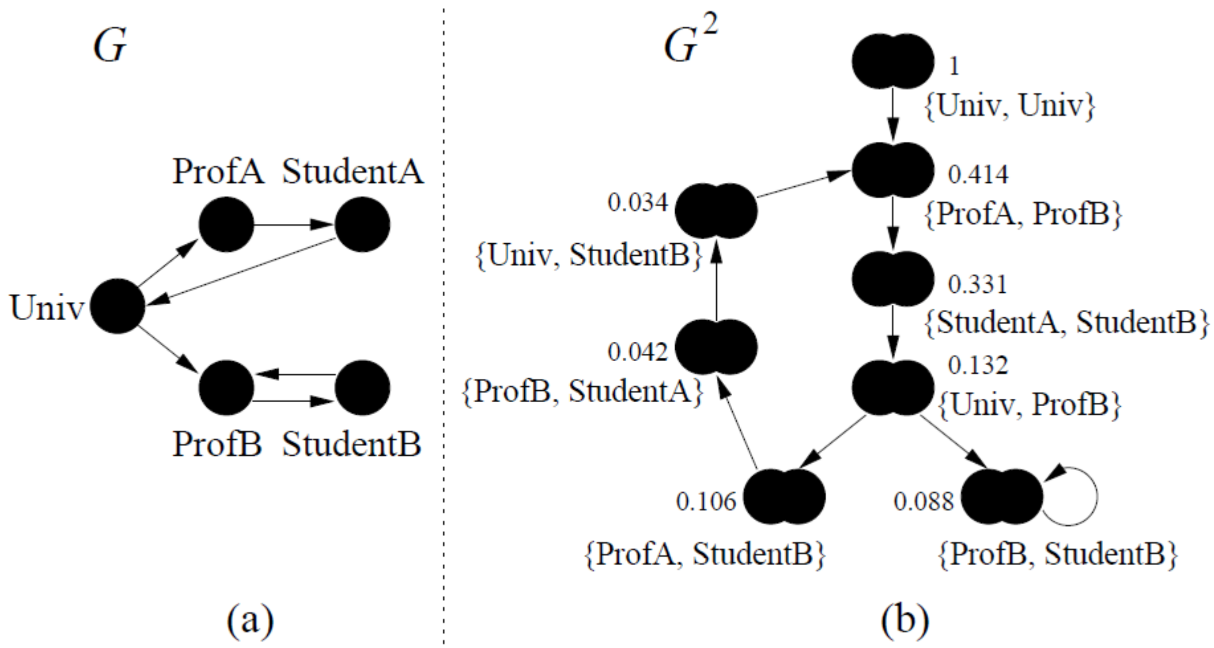


Figure 1: A small Web graph G and simplified node-pairs graph G^2. SimRank scores using parameter $C = 0.8$ are shown for nodes in G^2.

6.1 Experiment with Wikipedia

We used Simple English Wikipedia corpus to find semantic relatedness between concepts. We had 150495 entities(order of our matrix) which was a slightly higher than the number of articles in the given wiki at the time of writing, because some of those entities are redirection stubs. We used undirected graph representation of inter-wiki links which gave us 4454023 non zero elements in adjacency matrix. For graph that large direct computation of SimRank is infeasible. We used the following parameters for the experiment: $c = 0.3$, rank = 6000, no-oversampling and did ten iterations. While original paper [4] suggests $c = 0.8$ later it was suggested to use $c = 0.6$ for better results [6] and we choose $c = 0.3$ because it gave us subjectively better results. In experiment we used virtual server (VZ container) with 16 CPU cores and 100GiB RAM available (host node has 32 cores: 4 CPUs, each is 8-core AMD Opteron Processor 6272, 128GiB RAM). With this setup computations took roughly 40 hours.

Some examples provided in the table below. The first row is the word for which most similar words were queried, then in each of the columns most similar words are listed ordered by their SimRank score. The scores themselves would take too much space (they differ in 4-th or 5-th significant figures) and hence are omitted.

GNU	Earth (planet)	Liquid
Richard Matthew Stallman	Planet earth	Plasma (matter)
Linux operating system	SOL III	Matters
*nix	Geomagnetic	Particle theory of matter
Debian linux	Kola superdeep borehole	Potable water
Linux (kernel)	Oblate	Watery
Hurd	Guns, Germs, and Steel	Dihydrogen monoxide
Kernel (Computer science)	Ganges Plain	Hematological

Node Similarities

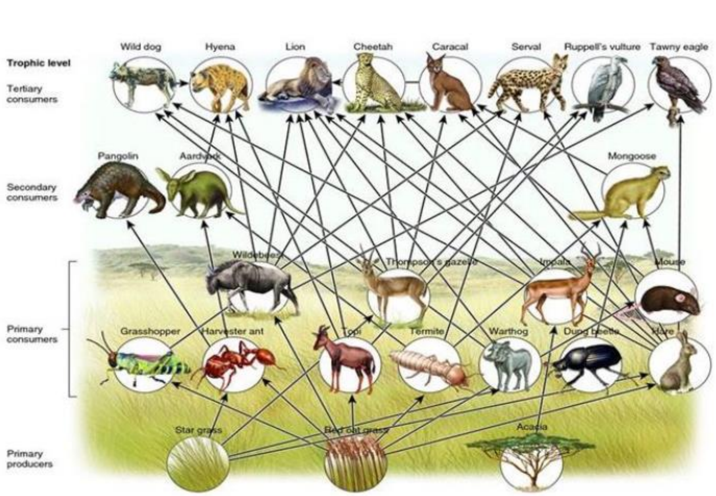
- Finding of the similar network nodes are important since we many want to
  - find ***nodes*** that ***play the same role*** in the network
  - find ***similar groups of nodes*** that ***play similar roles***

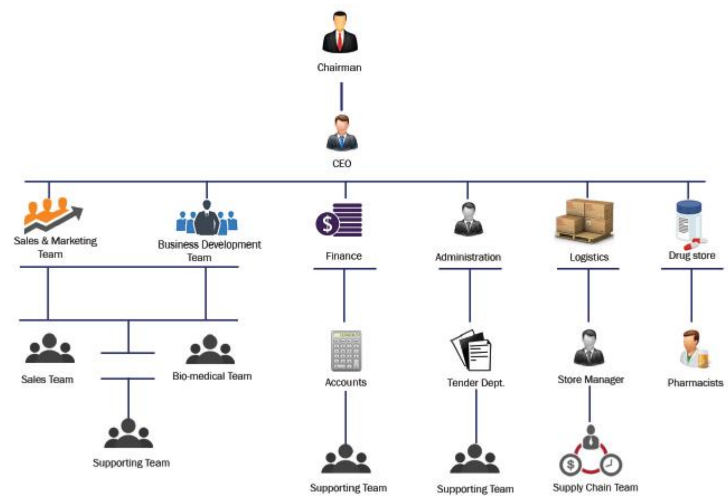
- Node similarity values may also used as features to
  - ***predict links*** between nodes
  - ***cluster objects,*** e.g., for ***collaborative filtering in a recommender system***
    - in which “similar” users and items are grouped based on the users’ preferences
  - ***extract the communities*** formed within the network
  - ***“find-similar-document” query*** on traditional text corpora or the World-Wide Web

Structural Roles in Networks

What are Roles?

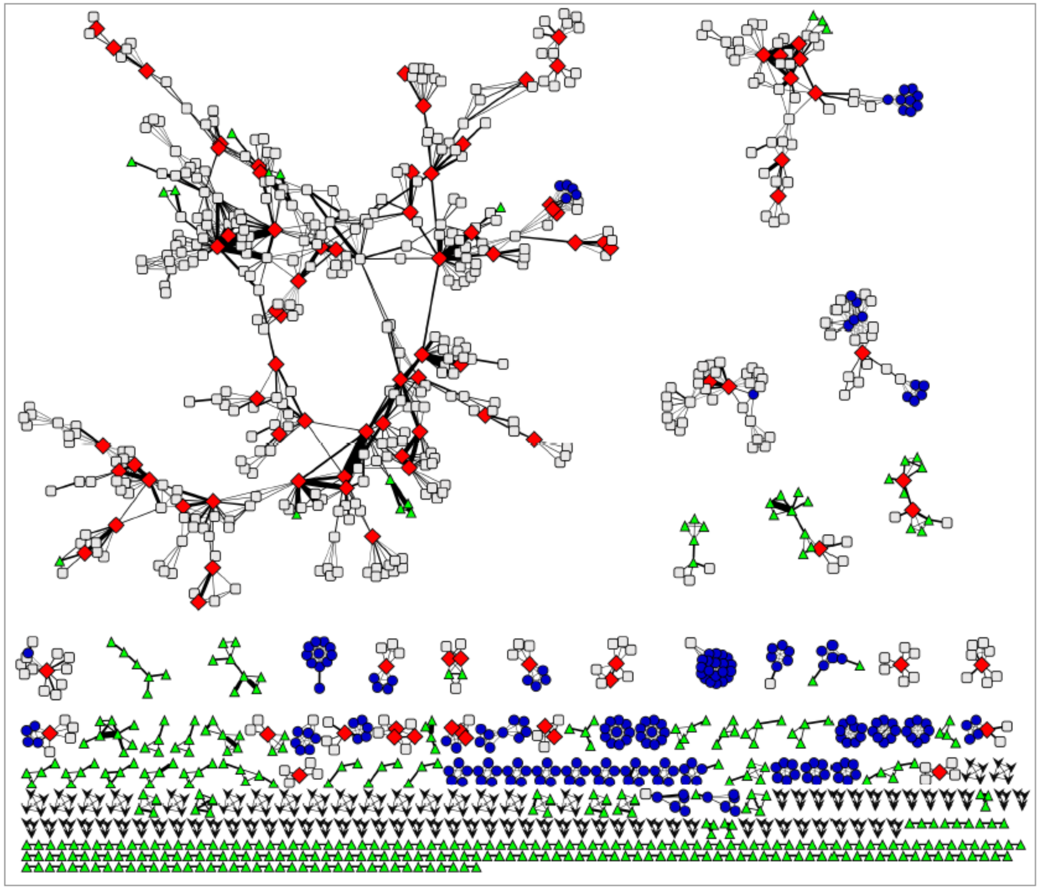
- ***Roles are "functions" of nodes in a network***:
  - Roles of ***species*** in ***ecosystems***
  - Roles of ***individuals*** in ***companies***





- Roles are measured by ***structural behaviors***:
  - centers of stars
  - members of cliques
  - peripheral nodes
  - …

Example of Roles



Network Science Co-authorship Graph
[Newman 2006]

- ◆ centers of stars
- ● members of cliques
- ▢ peripheral nodes
- ▲ isolated nodes

Roles vs. Groups in Networks

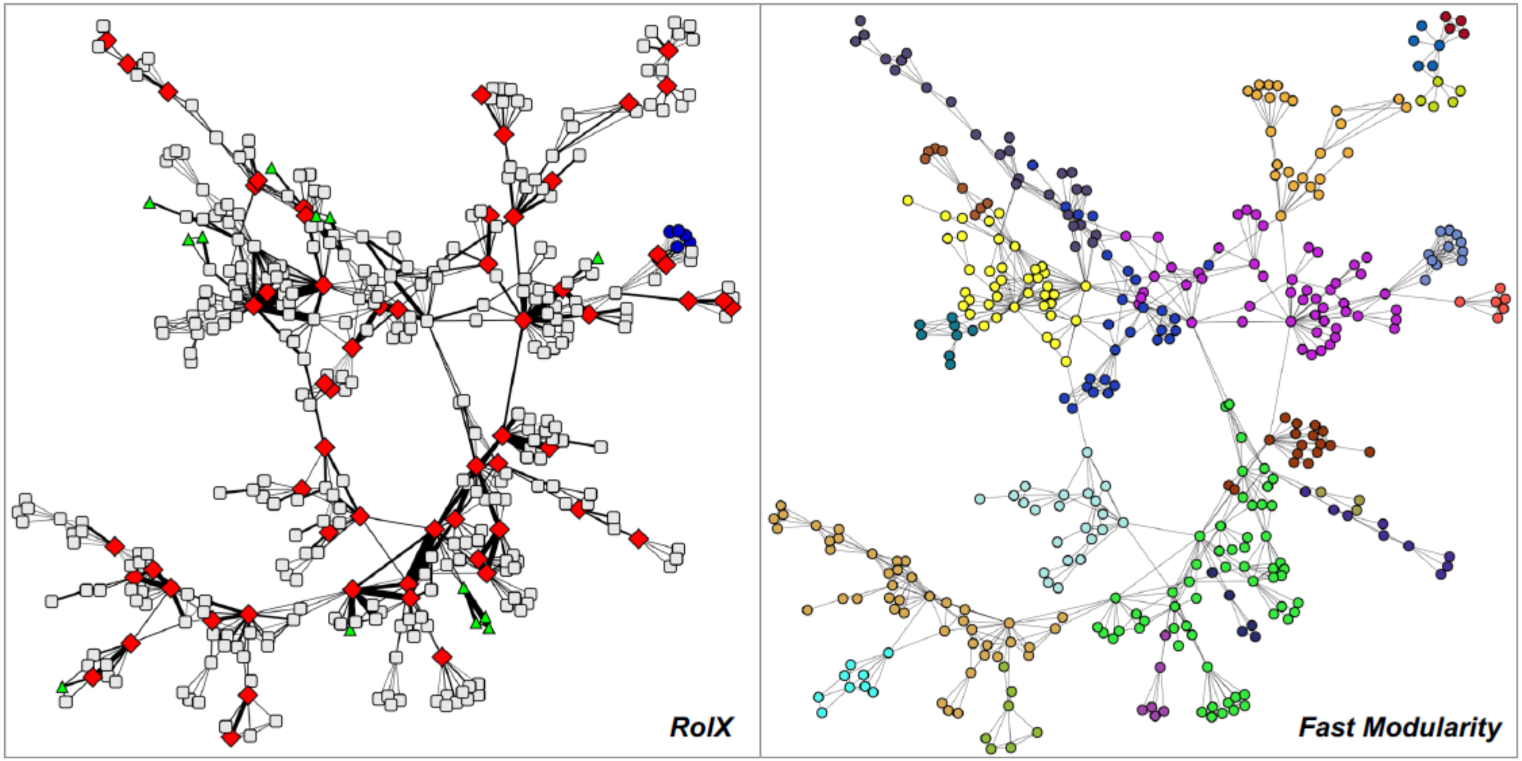
- ***Role:*** A collection of nodes which have similar positions in a network
  - Roles are based on the similarity of ties between subsets of nodes
  - Different from ***groups/communities***
    - Group is formed based on adjacency, proximity or reachability
    - typically adopted in current data mining
- ***Nodes with the same role need not be in direct, or even indirect interaction with each other***

Roles vs. Groups in Networks

- **Roles:**
  - A group of nodes with similar structural properties
- **Communities/Groups:**
  - A group of nodes that are well-connected to each other
- Roles and communities **are complementary**
- Consider the social network of a CS Dept.:
  - **Roles:** Faculty, Staff, Students
  - **Communities:** AI Lab, CyberSec. Lab, Robotics Lab

Roles vs. Groups in Networks

Roles | Communities

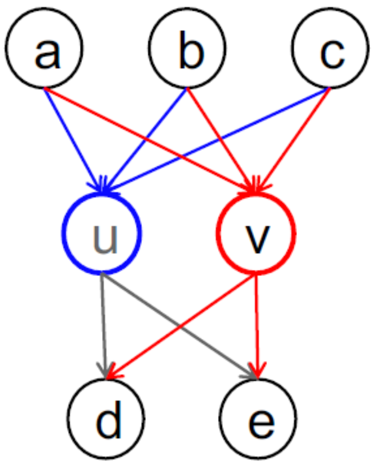


Henderson, *et al.*, KDD 2012

Clauset, *et al.*, Phys. Rev. E 2004

Roles: Node Equivalence (More Formally)

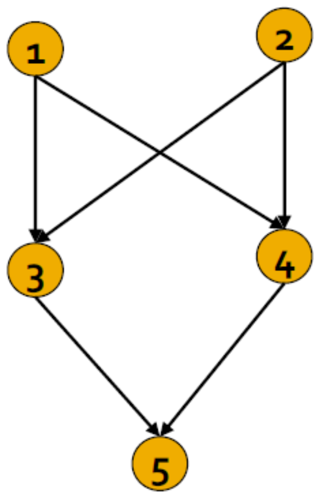
- ***Structural equivalence:*** Nodes u and v are structurally equivalent if they have the same relationships ***to all other nodes*** **[Lorrain & White 1971]**
  - Structurally equivalent nodes are likely to be similar in other ways - i.e., friendships in social networks



Structural Equivalence

- Nodes u and v are ***structurally equivalent:***
  - for all the other nodes k, node u has tie to k iff node v has tie to k

Example:



Adjacency matrix

	1	2	3	4	5
1	-	0	1	1	0
2	0	-	1	1	0
3	0	0	-	0	1
4	0	0	0	-	1
5	0	0	0	0	-

* nodes ***1 and 2***, ***3 and 4*** are structurally equivalent

Discovering Structural Roles and its applications

- Structural similarity
- Role generalization and transfer learning
- Making sense of roles

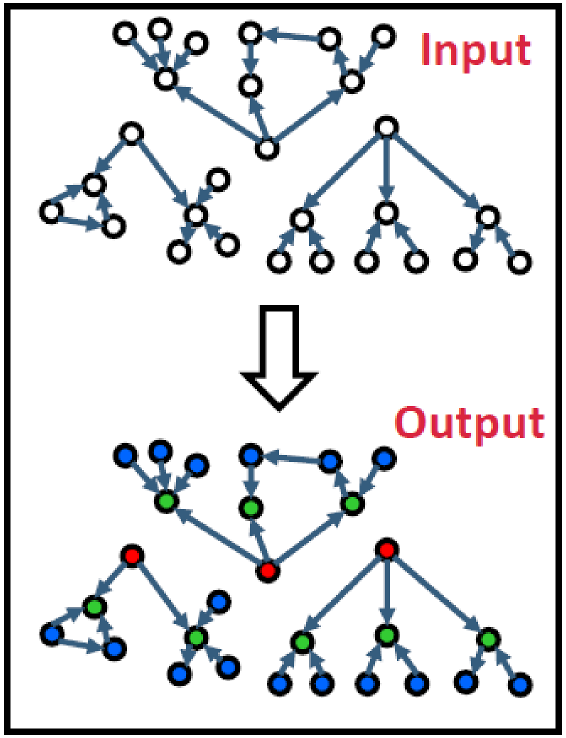
Importance of the Roles

Task	Example Application
Role query	Identify individuals with similar behavior to a known target
Role outliers	Identify individuals with unusual behavior
Role dynamics	Identify unusual changes in behavior
Identity resolution	Identify, de-anonymize, individuals in a new network
Role transfer	Use knowledge of one network to make predictions in another another
Network comparison	Compute similarity of networks, determine compatibility for knowledge transfer

Structural Role Discovery Method

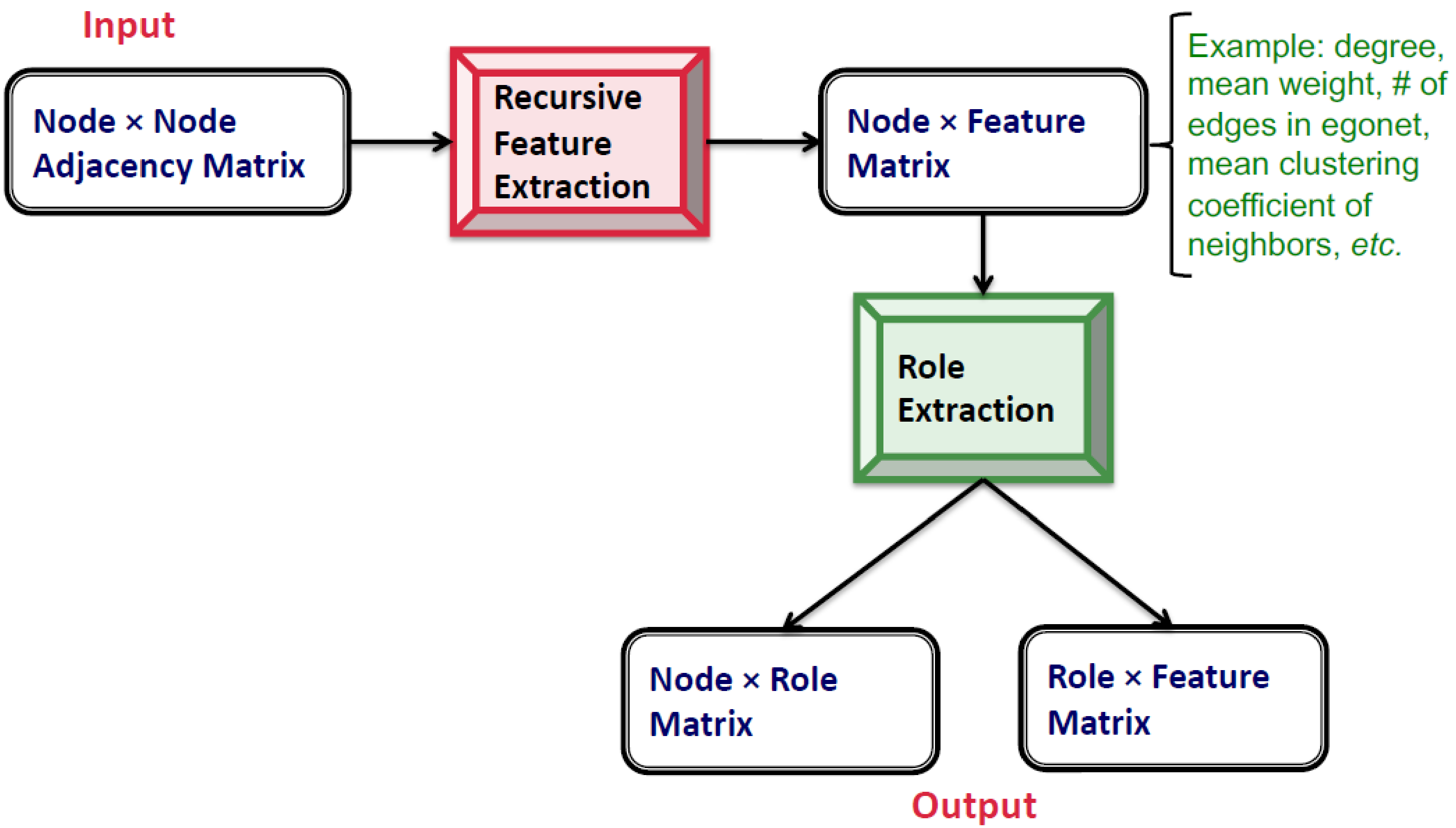
- ***RolX:*** Automatic discovery of nodes' structural roles in networks [[Henderson, et al. 2011b]]
  - Unsupervised learning approach
  - No prior knowledge required
  - Assigns a mixed-membership of roles to each node
  - Scales linearly in #(edges)

Role Discovery



✓ Automated discovery
✓ Behavioral roles
✓ Roles generalize

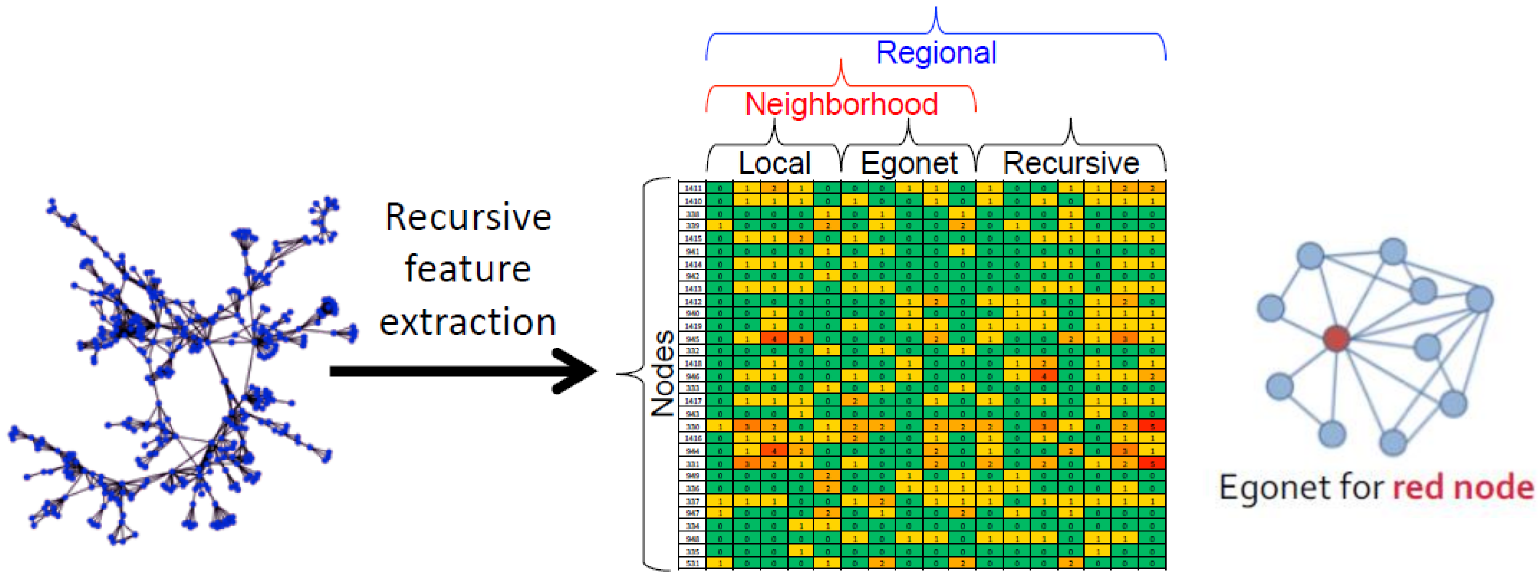
RolX Algorithm



Role Distributions

ReFex: Recursive Feature Extraction

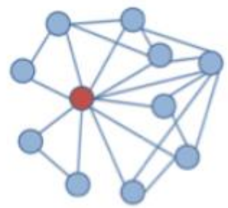
- Recursive feature extraction [[Henderson, et al. 2011a]] turns network connectivity into structural features



- ***Neighborhood features:*** What is a node's connectivity pattern?
- ***Recursive features:*** To what **kinds** of nodes is a node connected?

ReFex: Recursive Feature Extraction

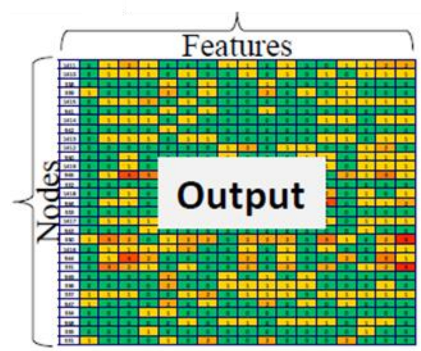
- **Idea:** Aggregate features of a node and use them to **generate new recursive features**
- ***Base set of a node's neighborhood features:***
  - **Local features:** All measures of the node degree:
    - If network is directed, include in- and out-degree, total degree
    - If network is weighted, include weighted feature versions
  - **Egonet features:** Computed on the node's egonet:
    - **Egonet** includes the node, its neighbors, and any edges in the induced subgraph on these nodes
    - #(within-egonet edges),
    - #(edges entering/leaving egonet)



Egonet for **red node**

ReFex: Recursive Feature Extraction

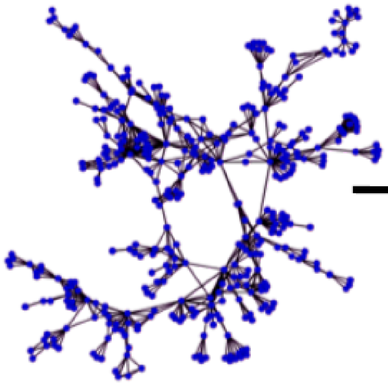
- Start with the ***base set of node features***
- ***Use the set of current node features to generate additional features:***
  - Two types of ***aggregate functions***: ***mean*** and ***sum***
    - *e.g.,* mean value of "unweighted degree" feature between all neighbors of a node
    - Compute means and sums over all current features, including other recursive features
  - Repeat



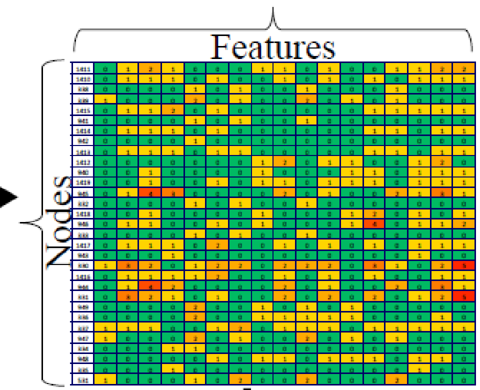
- The number of possible recursive features **grows exponentially** with each recursive iteration:
  - Reduce the number of features using a ***pruning technique***:
    - Look for pairs of features that are highly correlated
    - Eliminate one of the features whenever two features are correlated above a user-defined threshold

Role Extraction

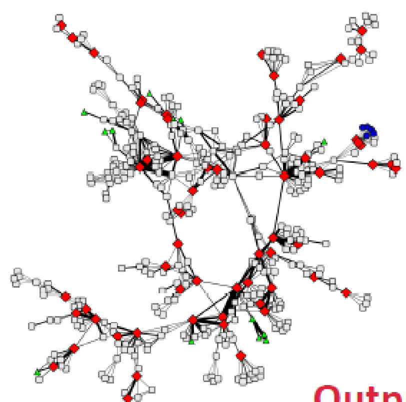
Input



Recursively extract features



Cluster nodes based on extracted features



Output

RolX uses non negative matrix factorization for clustering, MDL for model selection, and KL divergence to measure likelihood

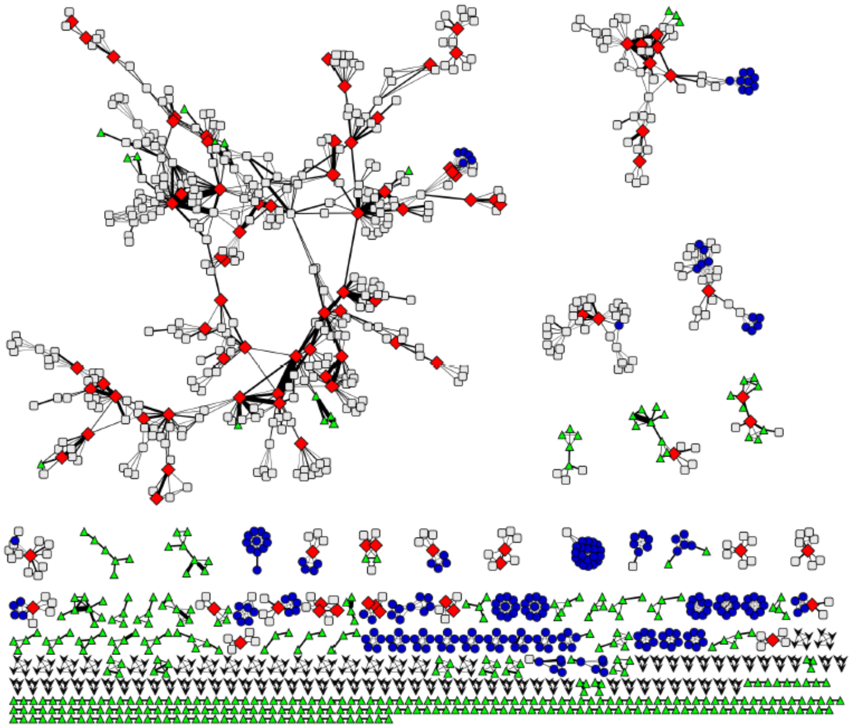
Applications

- ***Role Generalization / Transfer Learning***
  - role effectiveness for the across-network classification task (i.e. network transfer learning)
- ***Structural Similarity***
  - grouping nodes based on their structural similarity
- ***Sensemaking***
  - make "sense" of roles
    - ***NodeSense***: based on node measurements
    - ***NeighborSense***: based on neighbor measurements

Application: ***Structural Similarity***

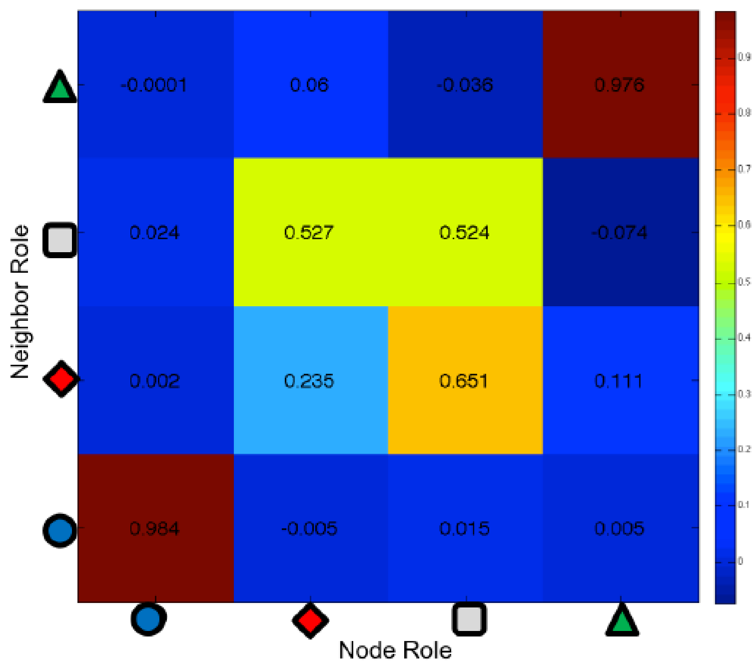
- **Task:** Cluster nodes based on their structural similarity
  – *Exploratory graph mining task (i.e., sensemaking)*
- **Two networks:**
  – ***Network science weighted co-authorship network***
    - (1589) Nodes: Network scientists
    - (2743) Edges: The number of co-authored papers
  – ***Political books co-purchasing network***
    - (105) Nodes: Political books on Amazon
    - (441) Edges: Frequent co-purchasing of books by the same buyers
- **Setup:** For each network
  – Use RolX to assign each node a distribution over the set of discovered structural roles
  – Determine similarity between nodes by comparing their ***role distributions***

Structural Similarity: Co-authorship Network



(a) Role-colored Visualization of the Network

each node is colored by the primary role that *RolX* finds



(b) Role Affinity Heat Map

Affinity matrix (red is high score, blue is low)

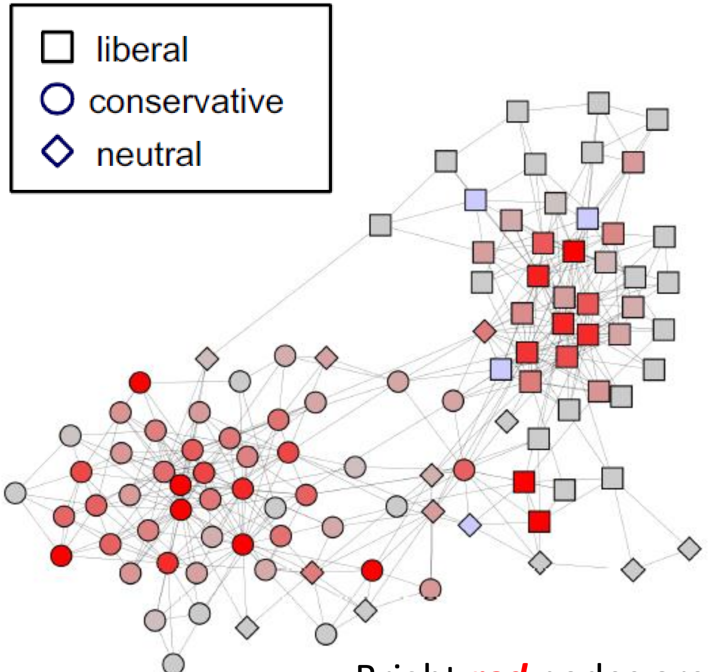
strong homophily for roles #1 and #4

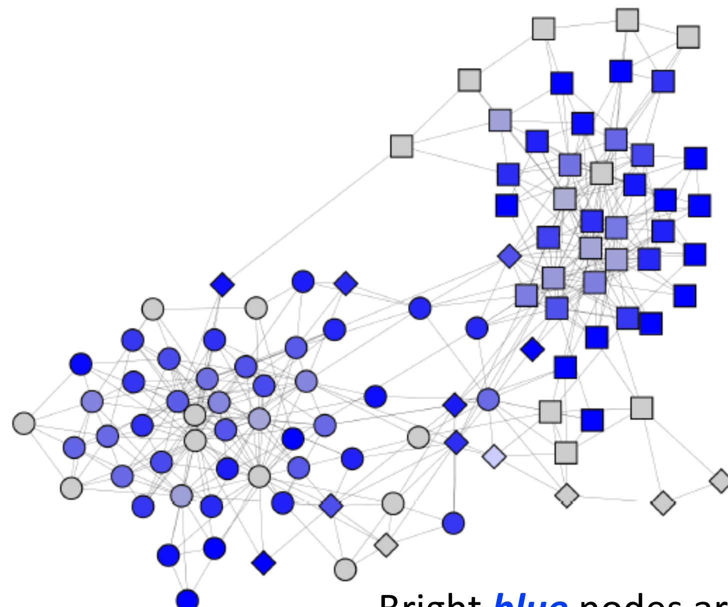
Making sense of roles:

- **Blue circle: Tightly knit**, nodes that participate in tightly-coupled groups
- **Red diamond: Bridge nodes**, that connect groups of nodes
- **Gray rectangle: Main-stream**, majority of nodes, neither a clique, nor a chain
- **Green triangle: Pathy**, nodes that belong to elongated clusters

Structural Similarity: Co-purchasing Network

Book labels (i.e., liberal, conservative, neutral) were not given to role discovery algorithm





Bright ***red*** nodes are ***locally central nodes***

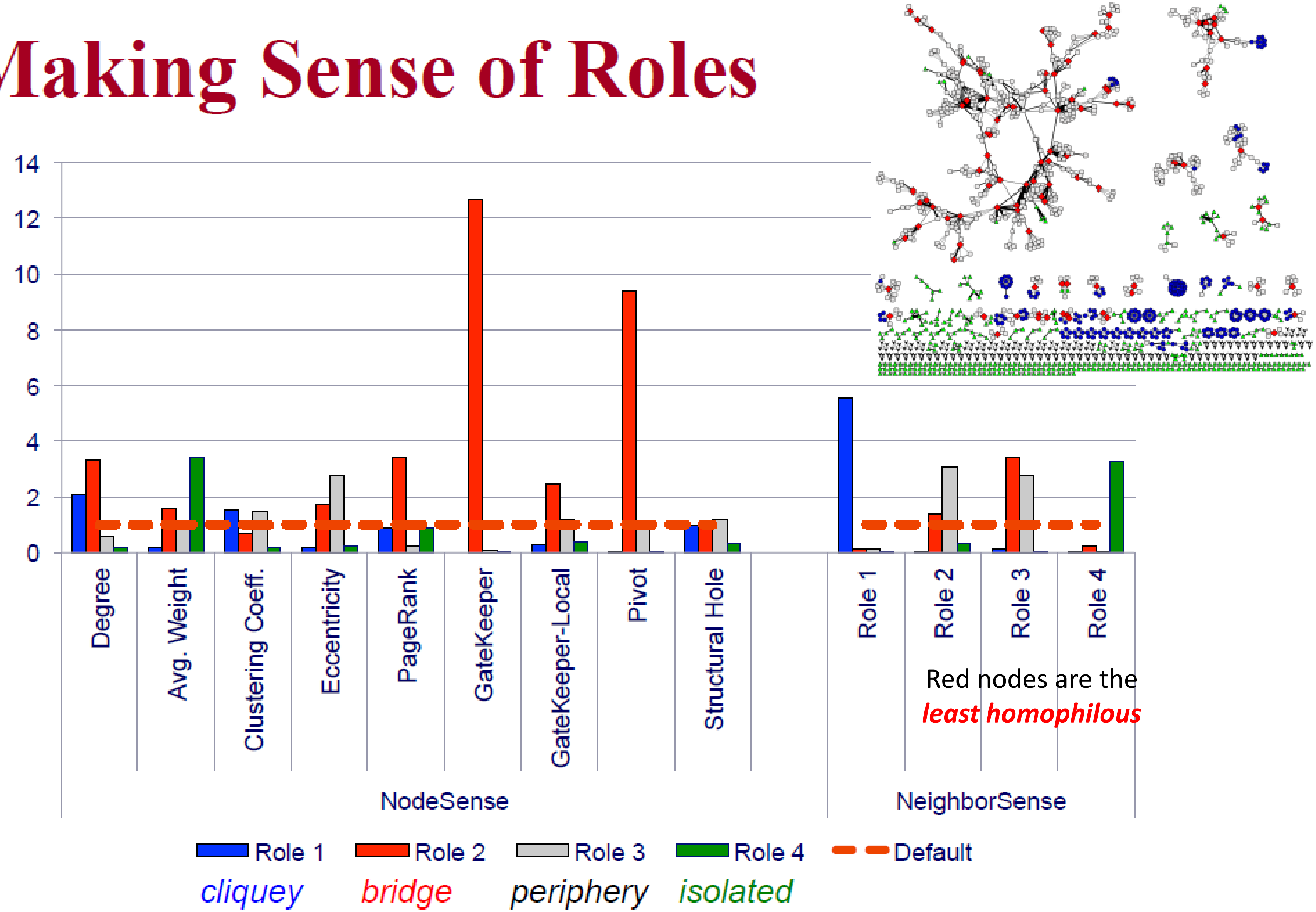
Bright ***blue*** nodes are ***peripheral nodes***

The ***redness*** of a node:
% membership in Role 1 (***local central-ness***)

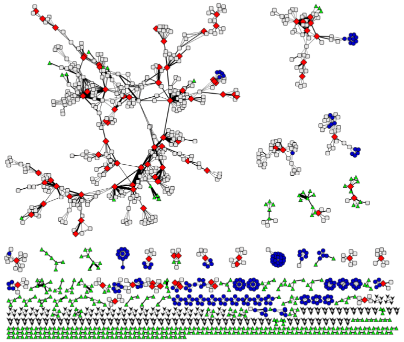
The ***blueness*** of a node:
% membership in Role 2 (***peripheral-ness***)

RolX's roles can be used to find similar nodes in disparate communities

Making Sense of Roles

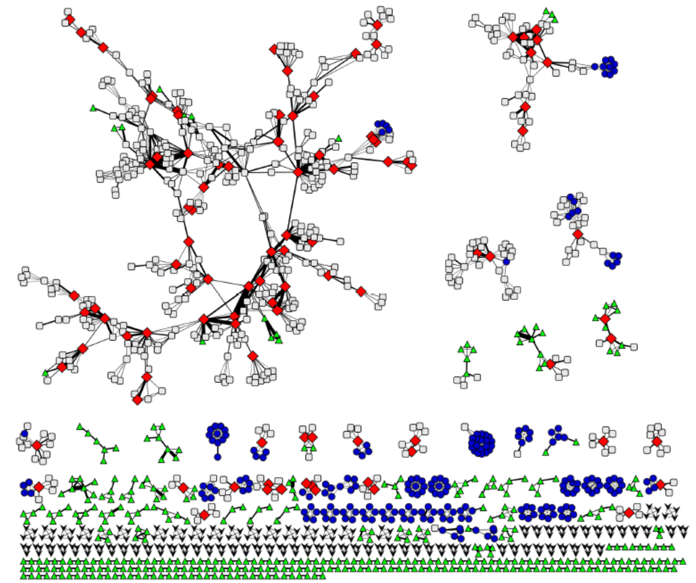


Nodes' Measurements



- ***Degree***
  - number of neighbors
- ***Weighted degree***
  - average weight of a node's links
- ***Clustering coefficient***
- ***Eccentricity***
  - the longest geodesic from a node
- ***PageRank***
- ***Gatekeeper***
  - whether a node is an articulation point for some pairs of nodes
- ***Local gatekeeper***
- ***Pivot***
  - a node with high betweenness
- ***Structural hole***
  - to what extent are a node's links redundant
- ***Peripheral nodes***
  - low degree and high eccentricity

Making Sense of Nodes



- **Role 1 nodes (blue circles):** authors with many coauthors and homophilic neighborhoods
  - high degree, high clustering coefficient, and high homophily
  - NOT gatekeepers (i.e. articulation points for some pairs of nodes)
  - NOT pivotal nodes (i.e., with high betweenness)
- ***Role 2 nodes (red diamonds):*** central and prolific authors
  - high total weight, low clustering coefficient but high degree, high PageRank
  - high affinity for Role 3 nodes (i.e., gray rectangles)
  - Removal severely interrupts graph connectivity because they are often gatekeepers and pivotal nodes
- ***Role 3 nodes (gray squares):*** peripheral authors
  - low degree and high eccentricity (i.e., nodes in the network periphery)
  - NOT gatekeepers, but can be pivotal
    - i.e. do not disconnect the graph, but often increase geodesic lengths when removed
- ***Role 4 nodes (green triangles):*** isolated authors
  - high average edge weight and homophilic neighborhoods
  - links are not redundant (w.r.t. structural holes)
    - BUT have low scores for most other measures (except homophily)

Sensemaking Analysis: useful for large networks which cannot be easily visualized