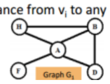
Eccentricity: The eccentricity of a node v_i is the maximum distance from v_i to any other nodes in the graph



- $e(v_i) = \max_j \{d(v_i, v_j)\}$
- E.g., e(A) = 1, e(F) = e(B) = e(D) = e(H) = 2

Radius of a connected graph G: the min eccentricity of any node in G

- $r(G) = \min_i \{e(v_i)\} = \min_i \{\max_j \{d(v_i, v_j)\}\}$
- E.g., $r(G_1) = 1$

Diameter of a connected graph G: the max eccentricity of any node in G

- $d(G) = \max_i \{e(v_i)\} = \max_{i,j} \{d(v_i, v_j)\}$
- E.g., $d(G_1) = 2$

Diameter is sensitive to outliers. Effective diameter: min # of hops for which a large fraction, typically 90%, of all connected pairs of nodes can reach each other

Generalizing to path of arbitrary length, we have: $N_{ij}^{(r)} = [A^r]_{ij}$

When starting and ending at the same vertex *i*, we have: $L_r = \sum_{i=1}^n [A^r]_{ii} = Tr\, A^r$

Matrix A written in the form of $A = UKU^T$, where U is the orthogonal matrix of eigenvalue and K is the diagonal matrix of eigenvalue

$L_r = Tr\,(UKU^T)^r = Tr\,(UK^rU^T) = Tr\,(UU^TK^r) = Tr\,(K^r) = \sum_i k_i^r$

- where k_i is the *i*-th eigenvalue of the adjacency matrix

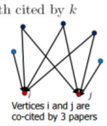
- The clustering coefficient of a node v_i is a measure of the density of edges in the neighborhood of v_i
- Let $G_i = (V_i, E_i)$ be the subgraph induced by the neighbors of vertex v_i, $|V_i| = n_i$ (# of neighbors of v_i), and $|E_i| = m_i$ (# of edges among the neighbors of v_i)
- **Clustering coefficient** of v_i for **undirected network** is

$$C(v_i) = \frac{\#\ edges\ in\ G_i}{max\ \#\ edges\ in\ G_i} = \frac{m_i}{\binom{n_i}{2}} = \frac{2 \times m_i}{n_i(n_i-1)}$$

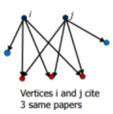
(corresp. to when G_i is a complete graph)

- For directed network, $C(v_i) = \frac{\#\ edges\ in\ G_i}{max\ \#\ edges\ in\ G_i} = \frac{m_i}{n_i(n_i-1)}$
- Clustering coefficient of a graph G: Averaging the local clustering coefficient of all the vertices (Watts & Strogatz): $C(G) = \frac{1}{n}\sum_i C(v_i)$

- Co-citation of vertices i and j: $A_{ik}A_{jk} = 1$ if i and j are both cited by k
  - # of vertices having outgoing edges pointing to both i and j $C_{ij} = \sum_{k=1}^n A_{ik}A_{jk} = \sum_{k=1}^n A_{ik}A_{kj}^T$
  - Co-citation matrix: It is a symmetric matrix $\mathbf{C} = \mathbf{A}\mathbf{A}^T$
  - Diagonal matrix (C_{ii}): total # papers citing i



- Bibliographic coupling of vertices i and j: $A_{ki}A_{kj} = 1$ if i and j both cite k
  - Bibliographic coupling of i and j: $B_{ij} = \sum_{k=1}^n A_{ki}A_{kj} = \sum_{k=1}^n A_{ik}^T A_{kj}$
  - Bibliographic coupling matrix: $\mathbf{B} = \mathbf{A}^T\mathbf{A}$
  - Diagonal matrix (B_{ii}): total # papers cited by *i*



- The projection to one-mode can be written in terms of the incidence matrix B as follows

$$P_{ij} = \sum_{k=1}^g B_{ki}B_{kj} = \sum_{k=1}^g B_{ik}{}^T B_{kj}$$

- Centrality: How "central" a node is in the network
- **Degree centrality**: degree of a node (the higher degree, more important the node)
- **Eccentricity centrality**: the less eccentric, the more central
  - $c(v_i) = 1/e(v_i)$
  - Central node: $e(v_i) = r(G)$ (if it equals the radius of G)
  - Periphery node: $e(v_i) = d(G)$ (if it equals the diameter of G)
  - Often used in facility location, e.g., emergency center
- **Closeness centrality**: the average of the shortest path length from the node to every other node in the network, indicating how close a node is to all other nodes in the network
  - $c(v_i) = 1/\sum_j d(v_i, v_j)$
  - median node v_m if v_m has the smallest total distance $\sum_j d(v_m, v_j)$
  - Facility location, e.g., shopping center, minimize total distance
- **Eigenvector centrality**: Measure the influence of a node in a network, i.e., connections to high-scoring nodes contribute more to the score of the node in question than equal connections to low-scoring nodes
- **Eigenvector centrality**, or **prestige**, importance, or rank of a node v
  - The more nodes point to v, the higher v's prestige
  - The more prestige of a node pointing to v, the higher v's prestige
- Let p(u) be prestige score for node u. Then

$$p(v) = \sum_u A(u,v) \cdot p(u) = \sum_u A^T(v,u) \cdot p(u)$$

- Written in vector form: $\mathbf{p}' = \mathbf{A}^T\mathbf{p}$
- At k-th iteration, we have $\mathbf{p}_k = (\mathbf{A}^T)^k \mathbf{p}_0$
  - Vetor $\mathbf{p}_k$ converges to the dominant eigenvector of $\mathbf{A}^T$ with increasing k
- Pagerank score computation: $\mathbf{p}' = (1-\alpha)\mathbf{N}^T\mathbf{p} + \alpha\mathbf{N}_r{}^T\mathbf{p} = ((1-\alpha)\mathbf{N}^T + \alpha\mathbf{N}_r{}^T)\mathbf{p} = \mathbf{M}^T\mathbf{p}$

- For a specific query, a page of high Pagerank score may not be that relevant
- HITS (Hyperlink Induced Topic Search) computes two values for a page
  - Authority score: analogous to pagerank/prestige scores
  - Hub score: based on how many "good" pages it points to
- How is HITS query-based?
  - first uses standard search engines to retrieve the set of relevant pages
  - then expands the set to include any page that point to or is pointed to by) some pages in the set
  - Any pages originating from the same host are eliminated
  - HITS is only applied on this expanded query-specific graph G
- Computation: $a(v) = \sum_u \mathbf{A}^T(v,u) \cdot h(u)$ $\quad h(v) = \sum_u \mathbf{A}(v,u) \cdot a(u)$
- In matrix computation (essentially two eigenvector computation):

$\mathbf{a}_k = \mathbf{A}^T\mathbf{h}_{k-1} = \mathbf{A}^T(\mathbf{A}\mathbf{a}_{k-2}) = (\mathbf{A}^T\mathbf{A})\mathbf{a}_{k-2}$ $\quad \mathbf{h}_k = \mathbf{A}\mathbf{a}_{k-1} = \mathbf{A}(\mathbf{A}^T\mathbf{h}_{k-2}) = (\mathbf{A}\mathbf{A}^T)\mathbf{h}_{k-2}$

- $\mathbf{AA}^T$ is the co-citation matrix and $\mathbf{A}^T\mathbf{A}$ is the bibliographic coupling matrix. Authority centrality is eigenvector centrality for the co-citation network

- A typical network has the following common properties:
  - *Few* connected components:
    - often only 1 or a small number, independent of network size
  - *Small* diameter:
    - often a constant independent of network size (like 6)
    - growing only logarithmically with network size or even shrink?
    - typically exclude infinite distances
  - A *high* degree of clustering:
    - considerably more so than for a random network
  - A *heavy-tailed* degree distribution:
    - a small but reliable number of high-degree vertices
    - often of *power law* form
- Real network: **large, sparse** (# of edges |E| = O(n), n: # of nodes)
- **Small-world property**: Avg. path length μ_L scales logarithmically with n (# of nodes in the graph): $\mu_L \propto \log n$
  - Ultra-small-world property: $\mu_L \ll \log n$
- **Scale-free property (power law distribution)**: most nodes have very small degree, but a few hub nodes have high degrees
  - The probability that a node has degree k: $f(k) \propto k^{-\gamma}$
  - log-log plot shows a straight line: $\log f(k) = \log(\alpha k^{-\gamma}) = -\gamma \log k + \log \alpha$
- **Clustering effect**:
  - Two nodes are more likely to be connected if they share a common neighbor
  - Clustering effect: a high clustering coefficient for graph G
  - C(k): avg clustering coefficient for nodes with degree k
  - Power law relationship between C(k) and k: $C(k) \propto k^{-\gamma}$
- ***Erdös-Rényi Random graph model:***
  - Gives few components and small diameter
  - does not give high clustering and heavy-tailed degree distributions
  - is the mathematically most well-studied and understood model
- ***Watts-Strogatz small world graph model:***
  - gives few components, small diameter and high clustering
  - does not give heavy-tailed degree distributions
- ***Barabási-Albert Scale-free model:***
  - gives few components, small diameter and heavy-tailed distribution
  - does not give high clustering
- The ER graphs fail to explain two important properties observed in real-world networks:
  - By assuming a constant and independent probability of two nodes being connected, they do not account for local clustering, i.e., having a low *clustering coefficient*
  - Do not account for the formation of hubs. Formally, the degree distribution of ER graphs converges to a Poisson distribution, rather than a *power law* observed in most real-world, *scale-free networks*

- Simple Ranking
  - Proportional to # of publications of an author and a venue
  - Considers only **immediate neighborhood** in the network

$$\begin{cases} \vec{r}_X(x) = \dfrac{\sum_{j=1}^n W_{XY}(x,j)}{\sum_{i=1}^m \sum_{j=1}^n W_{XY}(i,j)} \\ \vec{r}_Y(y) = \dfrac{\sum_{i=1}^n W_{XY}(i,y)}{\sum_{i=1}^m \sum_{j=1}^n W_{XY}(i,j)} \end{cases}$$

- Clustering and ranking are mutually enhanced
  - Better clustering: Rank distributions for clusters are more distinguishing from each other
  - Better ranking: Better metric for objects is learned from the ranking

Authority Ranking

- Methodology: **Propagate** the ranking scores in the network over different types
- Rule 1: Highly ranked authors publish *many* papers in highly ranked venues

$$\vec{r}_Y(j) = \sum_{i=1}^m W_{YX}(j,i)\vec{r}_X(i)$$

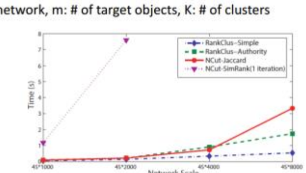
- Rule 2: Highly ranked venues attract *many* papers from *many* highly ranked authors

$$\vec{r}_X(i) = \sum_{j=1}^n W_{XY}(i,j)\vec{r}_Y(j)$$

- Rule 3: The rank of an author is enhanced if he or she co-authors with *many* highly ranked authors

$$\vec{r}_Y(i) = \alpha \sum_{j=1}^m W_{YX}(i,j)\vec{r}_X(j) + (1-\alpha)\sum_{j=1}^n W_{YY}(i,j)\vec{r}_Y(j)$$

- **E-step** assigns objects to clusters according to the current probabilistic clustering or parameters of probabilistic clusters
- **M-step** finds the new clustering or parameters that minimize the sum of squared error (SSE) or maximize the expected likelihood
- Conditional rank distribution as cluster feature
  - For each cluster C_k, *the conditional rank scores,* $r_X|C_k$ *and* $r_Y|C_k$*, can be viewed as* conditional rank distributions of *X and Y, which are the features for cluster* C_k
- Cluster membership as object feature
  - From $p(k|o_i) \propto p(o_i|k)p(k)$, *the higher its conditional* rank in a cluster ($p(o_i|k)$), *the higher possibility an object will belong to that cluster* ($p(k|o_i)$)
  - Highly ranked attribute object has more impact on determining the cluster membership of a target object
- Parameter estimation using the Expectation-Maximization algorithm
  - E-step: Calculate the distribution $p(z = k|y_j, x_i, \Theta)$ *based on* the current value of Θ
  - M-Step: Update Θ according to the current *distribution*
- At each iteration, |E|: edges in network, m: # of target objects, K: # of clusters
  - Ranking for sparse network
    - ~O(|E|)
  - Mixture model estimation
    - ~O(K|E|+mK)
  - Cluster adjustment
    - ~O(mK^2)
- In all, linear to |E|
  - ~O(K|E|)



Comparison of algorithms on execution time

- Note: SimRank will be at least quadratic at each iteration since it evaluates distance between every pair in the network

- Random walk (RW):
  - The probability of random walk starting at *x* and ending at *y*, with meta-path *P*

$$s(x,y) = \sum_{p \in P} prob(p)$$

  - Used in Personalized PageRank (P-Pagerank) (Jeh and Widom 2003)
  - Favors **highly visible** objects (i.e., objects with large degrees)
- Pairwise random walk (PRW):
  - The probability of pairwise random walk starting at (*x, y*) and ending at a common object (say *z*), following a meta-path (P_1, P_2)

$$s(x,y) = \sum_{(p_1,p_2) \in (P_1,P_2)} prob(p_1)prob(p_2)$$

  - Used in SimRank (Jeh and Widom 2002)
  - Favors ***pure*** objects (i.e., objects with highly skewed distribution in

- SimRank (Jeh and Widom 2002)
  - Base: objects are maximally similar to themselves, i.e., $s_0(a,b) = \begin{cases} 1, & \text{if } a = b, \\ 0, & \text{if } a \neq b. \end{cases}$
  - Induction: Two objects are considered to be similar if they are referenced by similar objects $s(a,b) = \frac{C}{|I(a)||I(b)|}\sum_{i=1}^{|I(a)|}\sum_{j=1}^{|I(b)|} s(I_i(a), I_j(b))$
  - The computation is quite costly: Many efficient computation methods proposed

Glen Jeh and Jennifer Widom. SimRank: A Measure of Structural-Context Similarity. In KDD'02

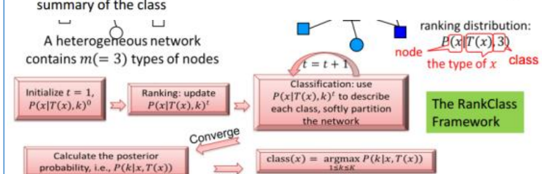
- Personalized PageRank (P-Pagerank) (Jeh and Widom 2003)
  - P-PageRank score *x* is defined as: $x = \alpha Px + (1-\alpha)b$, where P is a transition matrix of the network G, *b* is a stochastic vector, called *personalized vector*, and $\alpha \in (0, 1)$ is the *teleportation constant*
  - Efficient computation methods are also studied (e.g., Maehara, et al., VLDB'14)

Glen Jeh and Jennifer Widom. Scaling Personalized Web Search, In *WWW* 2003

Why User Guidance in Clustering?

- Different users may like to get different clusters for different clustering goals
  - Ex. Clustering authors based on their connections in the network
- **Problem: User-guided clustering with meta-path selection**
- Input:
  - The target type for clustering *T*
  - # of clusters *k*
  - Seeds in some clusters: $L_1, \ldots, L_k$
  - Candidate meta-paths: $P_1, \ldots, P_M$
- Output:
  - Weight of each meta-path: $w_1, \ldots, w_m$
  - Clustering results that are consistent with the user guidance
- Two levels of personalization
  - Data level
    - Most recommendation methods use one model for all users and rely on personal feedback to achieve personalization
  - Model level
    - With different entity relationships, we can learn personalized models for different users to further distinguish their differences
- **Classification in heterogeneous networks**
  - **Knowledge propagation:** Class label knowledge propagated across multi-typed objects through a heterogeneous network
- **GNetMine [Ji et al., PKDD'10]:** Objects are treated equally
- **RankClass [Ji et al., KDD'11]:** Ranking-based classification
  - Highly ranked objects will play more role in classification
  - An object can only be ranked high in some focused classes
  - Class membership and ranking are stat. distributions
  - Let ranking and classification mutually enhance each other!
  - Output: Classification results + ranking list of objects within each class

- Different objects within one class have different importance/visibility!
- The ranking of objects within one class serves as an informative understanding and summary of the class



Graph-Based Ranking

- Intuitive idea: authority propagation
  - Objects linked together are likely to share similar ranking scores within class k
- Update the ranking score of each object by looking t the ranking of its neighbors

$$P(x_{ip}|\chi_i,k)^{t+1} \propto \frac{\boxed{\sum_{j=1}^m \sum_{q=1}^{n_j} \lambda_{ij} S_{ij,pq} P(x_{jq}|\chi_j,k)} + \boxed{\alpha_i P(x_{ip}|\chi_i,k)^0}}{\sum_{j=1}^m \lambda_{ij} + \alpha_i}$$

The initial ranking score

Weighted average of the neighbors' ranking scores

The NetClus Algorithm

- Generate initial partitions for target objects and induce initial net-clusters from the original network
- Repeat // An E-M Framework
  - Build ranking-based probabilistic generative model for each net-cluster
  - Calculate the posterior probabilities for each target object
  - Adjust their cluster assignment according to the new measure defined by the posterior probabilities to each cluster
- Until the clusters do not change significantly
- Calculate the posterior probabilities for each attribute object in each net-cluster

$A - P \rightarrow P - A$	a_i cites a_j
$A - P \leftarrow P - A$	a_i is cited by a_j
$A - P - V - P - A$	a_i and a_j publish in the same venues
$A - P - A - P - A$	a_i and a_j are co-authors of the same authors
$A - P - T - P - A$	a_i and a_j write the same topics
$A - P \rightarrow P \rightarrow P - A$	a_i cites papers that cite a_j
$A - P \leftarrow P \leftarrow P - A$	a_i is cited by papers that are cited by a_j
$A - P \rightarrow P \leftarrow P - A$	a_i and a_j cite the same papers
$A - P \leftarrow P \rightarrow P - A$	a_i and a_j are cited by the same papers

- Training and test pair: <x_i, y_i> = <history feature list, future relationship label>

	A—P—A—P—A	A—P—V—P—A	A—P—T—P—A	A—P→P—A	A—P—A
<Mike, Ann>	4	5	100	3	**Yes = 1**
<Mike, Jim>	0	1	20	2	**No = 0**

- Logistic Regression Model
  - Model the probability for each relationship as

$$p_i = \frac{e^{x_i \beta}}{e^{x_i \beta} + 1}$$

    - β is the coefficients for each feature (including a constant 1)
  - MLE (Maximum Likelihood Estimation)
    - Maximize the likelihood of observing all the relationships in the training data

$$L = \prod_i p_i^{y_i} (1 - p_i)^{(1-y_i)}$$

We study four measures that defines a relationship *R* encoded by a meta path

- Path Count: Number of path instances between authors following *R*

$$PC_R(a_i, a_j)$$

- Normalized Path Count: Normalize path count following *R* by the "degree" of authors

$$NPC_R(a_i, a_j) = \frac{PC_R(a_i,a_j) + PC_{R^{-1}}(a_j,a_i)}{PC_R(a_i,\cdot) + PC_R(\cdot,a_j)}$$

- Random Walk: Consider one way random walk following *R*

$$RW_R(a_i, a_j) = \frac{PC_R(a_i,a_j)}{PC_R(a_i,\cdot)}$$

- Symmetric Random Walk: Consider random walk in both directions

$$SRW_R(a_i, a_j) = RW_R(a_i, a_j) + RW_{R^{-1}}(a_j, a_i)$$

Generate *L* different **meta-path** (path types) connecting users and items → Propagate user implicit feedback along each meta-path → Calculate latent-features for users and items for each meta-path with **NMF** related method

Recommendation Models

Observation 1: Different meta-paths may have different importance

Global Recommendation Model

$$\underbrace{\boxed{\hat{r}(u_i, e_j)}}_{\text{ranking score}} = \sum_{q=1}^{L} \theta_q \cdot \underbrace{\boxed{\hat{U}_i^{(q)} \hat{V}_j^{(q)T}}}_{\text{features for user } i \text{ and item } j} \quad (1)$$

the q-th meta-path

Observation 2: Different users may require different models

Personalized Recommendation Model

$$\hat{r}_p(u_i, e_j) = \sum_{k=1}^{c} \underbrace{\boxed{sim(C_k, u_i)}}_{\text{user-cluster similarity}} \sum_{q=1}^{L} \theta_q^{\{k\}} \cdot \hat{U}_i^{(q)} \hat{V}_j^{(q)T} \quad (2)$$

c total soft user clusters

- Bayesian personalized ranking (Rendle UAI'09)
- Objective function

sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$.

$$\min_{\Theta} - \sum_{u_i \in \mathcal{U}} \boxed{\sum_{(e_a > e_b) \in R_i}} \ln \sigma(\hat{r}(u_i, e_a) - \hat{r}(u_i, e_b)) + \frac{\lambda}{2} \|\Theta\|_2^2 \quad (3)$$

for each correctly ranked item pair
i.e., u_i gave feedback to e_a but not e_b

Soft cluster users with NMF + k-means → For each user cluster, learn one model with Eq. (3) → Generate personalized model for each user on the fly with Eq. (2)

Learning Personalized Recommendation Model

How likely a query manuscript ***q*** will cite a candidate paper ***p*** (suppose **K** interest groups):

$$s(q, p) = \sum_{k=1}^{K} \boxed{\theta_q^{(k)}} \cdot \left\{ \boxed{r^{(k)}(q, p)} + \boxed{f_{\mathcal{P}}^{(k)}(p)} \right\}$$

query's group membership

relative citation score (how likely ***q*** will cite ***p***) within each group

paper relative relevance (query-candidate paper)

paper relative authority (candidate paper)

It is desirable to suggest papers that have *high* relative citation scores across *multiple* related interest groups of the query manuscript

- **Learn each query's group membership: scalability & generalizability**
- **Leverage the group memberships of related attribute objects to *approximate* query's group membership**

$$\theta_q^{(k)} = \boxed{\sum_{\mathcal{X} \in \{\mathcal{A}, \mathcal{V}, \mathcal{T}\}}} \boxed{\sum_{x \in N_{\mathcal{X}}(q)}} \frac{\boxed{\theta_x^{(k)}}}{|N_{\mathcal{X}}(q)|}$$

Different types of attribute objects (***X*** = authors/venues/terms)

Query's related (linked) objects of type-***X***

Attribute object's group membership (*to learn*)

$$r^{(k)}(q, p) = \sum_{l=1}^{L} w_k^{(l)} \boxed{\phi^{(l)}(q, p)}$$

meta path-based relevance score (*l*-th feature)

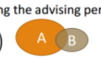
$$s(q, p) = \sum_{k=1}^{K} \theta_q^{(k)} \cdot \left\{ r^{(k)}(q, p) + \boxed{f_{\mathcal{P}}^{(k)}(p)} \right\}$$

Paper relative authority: A paper may have quite different visibility/authority among different groups, even it is overall highly cited

- Propagation of simple, commonly accepted constraints in Time-Constrained Probabilistic Factor Graph (TPFG)
  - *"Advisor has more publications and longer history than advisee at the time of advising"*
  - *"Once an advisee becomes advisor, s/he will not become advisee again"*

- Right heuristics: Advisee (B) tends to coauthor with advisor (A) during the advising period
- Kulczinski measure :

$$kulc(A, B) = \frac{|A \cap B|}{2} \left(\frac{1}{|A|} + \frac{1}{|B|} \right)$$

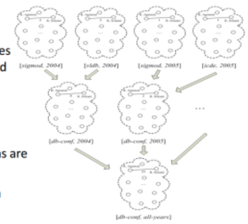


  - high overlap between their publications – Kulczinski measure
  - It is more often to see advisor in a publication but not advisee: Imbalance Ratio measure:

$$IR(A, B) = \frac{|A| - |B|}{|A \cup B|}$$

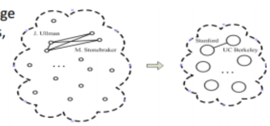
  - start time ≈ the year they start to collaborate
  - end time ≈ the year Kulczinski measure dropped significantly

Informational OLAP



- In the DBLP network, study the collaboration patterns among researchers
- Dimensions come from informational attributes attached at the whole snapshot level, so-called *Info-Dims*
- I-OLAP Characteristics:
  - Overlay multiple pieces of information
  - No change on the objects whose interactions are being examined
    - In the underlying snapshots, each node is a researcher
    - In the summarized view, each node is still a researcher

Topological OLAP



- Dimensions come from the node/edge attributes inside individual networks, so-called *Topo-Dims*
- T-OLAP Characteristics
  - Zoom in/Zoom out
  - Network topology changed: "generalized" nodes and "generalized" edges
    - In the underlying network, each node is a researcher
    - In the summarized view, each node becomes an institute that comprises multiple researchers

The DISTINCT Methodology

- Measure similarity between references
  - Link-based similarity: Linkages between references
    - References to the same object are more likely to be connected (Using random walk probability)
  - Neighborhood similarity
    - Neighbor tuples of each reference can indicate similarity between their contexts
- Self-boosting: Training using the "same" bulky data set
- Reference-based clustering
  - Group references according to their similarities

- Build a training set automatically
  - Select distinct names, e.g., Johannes Gehrke
  - The collaboration behavior within the same community share some similarity
  - Training parameters using a typical and large set of "unambiguous" examples
- Use SVM to learn a model for combining different join paths
  - Each join path is used as two attributes (with link-based similarity and neighborhood similarity)
  - The model is a weighted sum of all attributes
- Single-link (highest similarity between points in two clusters) ?
  - No, because references to different objects can be connected.
- Complete-link (minimum similarity between them)?
  - No, because references to the same object may be weakly connected.
- Average-link (average similarity between points in two clusters)?
  - A better measure
  - *Refinement: Average neighborhood similarity* and *collective random walk probability*