

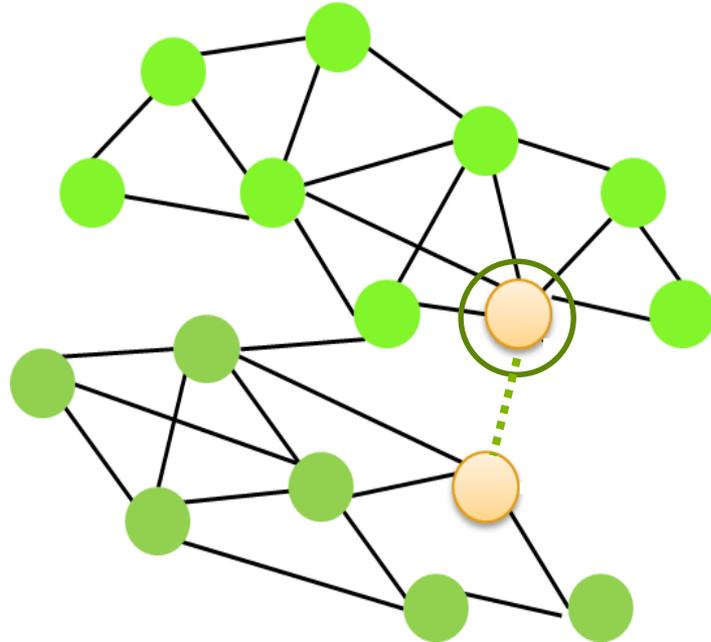
Link Prediction

Eman Badr
Mohammed Saquib Akmal Khan

11-06-2013

Link Prediction

Which pair of nodes **should** be connected?

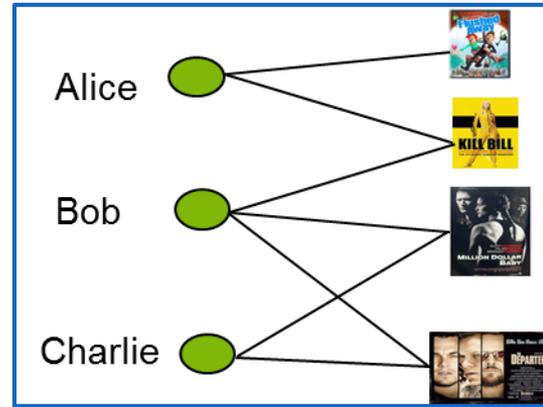


Applications

- Facebook friend suggestion
- Recommendation systems
- Monitoring and controlling computer viruses that use email as a vector
- Predict unobserved links in protein protein interaction networks in biological system

Suggestions
Add people you know as friends and connect with public profiles you like.

 Keerthi Malyala Add as friend	×	 Vijay Kamal Add as friend	×
 Rinal Chheda Add as friend	×	 Chitra Rapaka Add as friend	×
 Saloni Mehta Add as friend	×	 Coolgirl Sailu Add as friend	×
 Ashwin Zade Add as friend	×	 Megha Jain Add as friend	×



Customers Who Bought This Item Also Bought



[Garmin nüvi 360 3.5-Inch Bluetooth Portable GPS Navigator](#)
★★★★☆ (695)
[Click for price](#)



[Garmin nüvi 660 4.3-Inch Widescreen Bluetooth Portable GPS Nav...](#)
★★★★☆ (754) \$320.64



[Garmin Suction Cup Mount for Nüvi \(010-10723-031\)](#)
★★★★☆ (54) \$26.10

Methods for Link Prediction

- Link prediction problems rely on **Homophily** \Leftrightarrow similar nodes are more likely to be connected.
- All the methods assign a connection weight $score(i,j)$ to pairs of nodes i, j , based on the input graph, and then produce a ranked list in decreasing order of $score(i, j)$.
- Can be viewed as computing a measure of proximity or “similarity” between nodes i and j .

Typical Approaches

- Common neighbors

$$\text{score}(i, j) = |\Gamma(i) \cap \Gamma(j)|$$

- Jaccard coefficient

$$\text{score}(i, j) = |\Gamma(i) \cap \Gamma(j)| / |\Gamma(i) \cup \Gamma(j)|$$

- Degree Product

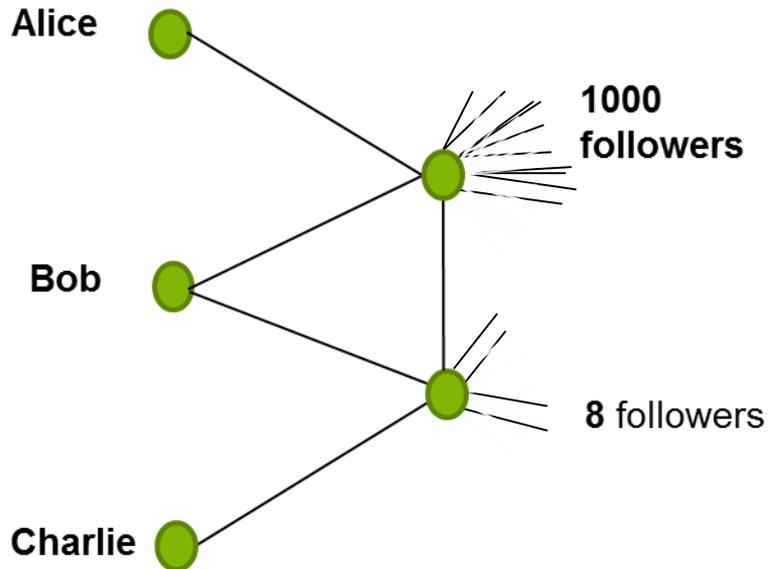
$$\text{score}(i, j) = |\Gamma(i)| |\Gamma(j)|$$

- Shortest path

- $\text{score}(i, j)$ is 1 divided by the length of the shortest path through the network from i to j (or zero for pairs that are not connected by any path).

Typical Approaches (cont.)

- Adamic/Adar link prediction heuristic
 - gives more weight to low degree common neighbors.



Hierarchical Structure and Predicting Missing Links

Consists of three parts

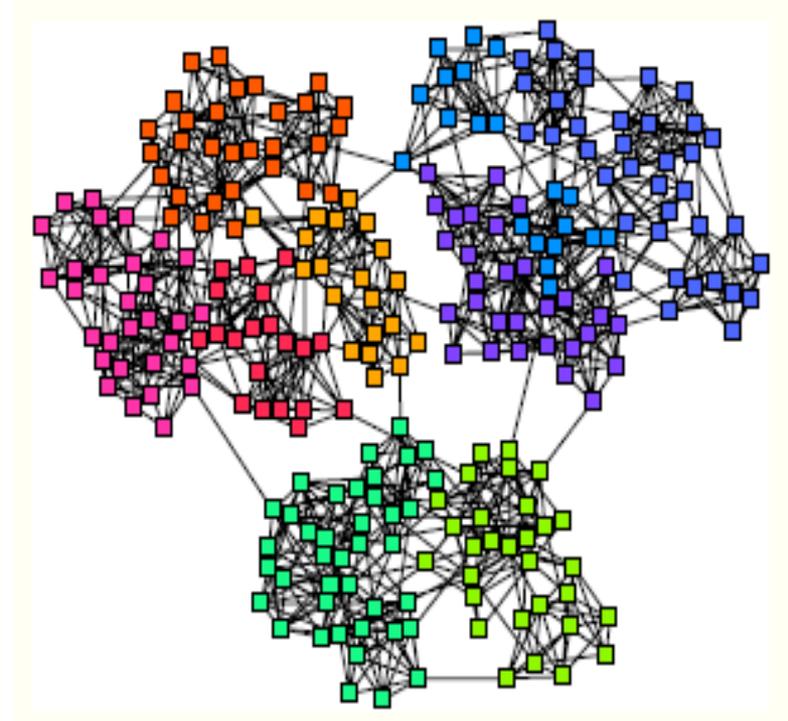
- Inferring hierarchical structure from network data
- Generating random graphs which are statistically similar to real ones
- Predicting missing links using hierarchal structure

Inferring hierarchical structure from network data

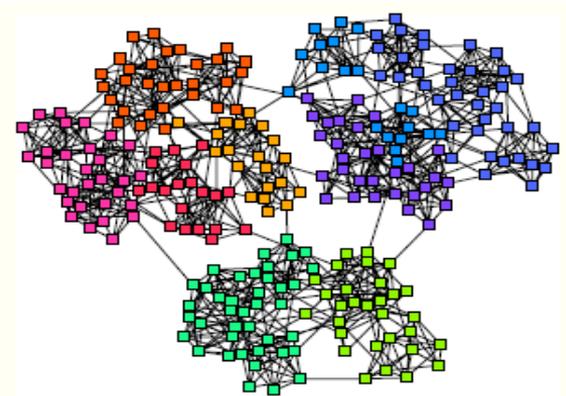
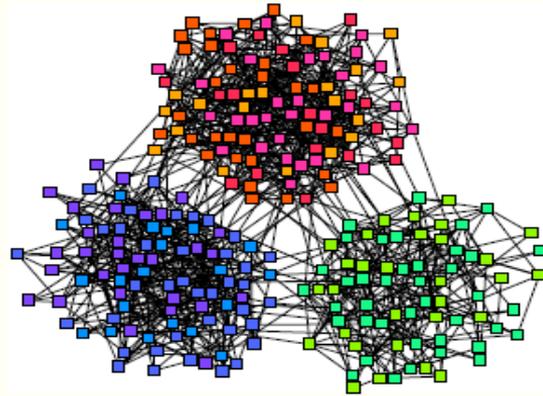
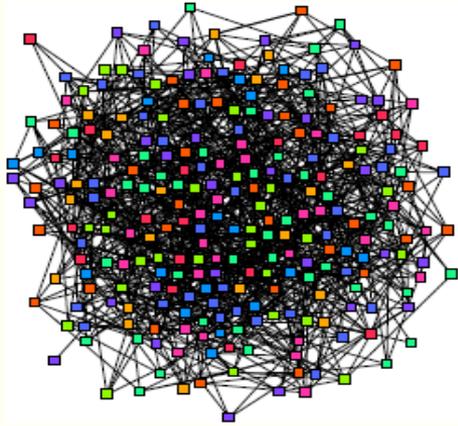
Nature 453, 98 - 101 (2008) - Aaron Clauset, Cristopher Moore, M. E. J. Newman

Hierarchical Structure

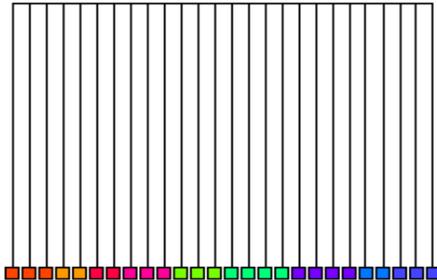
- Recent studies suggest that networks often exhibit hierarchical organization, where vertices divide into groups that further subdivide into groups of groups, and so forth over multiple scales



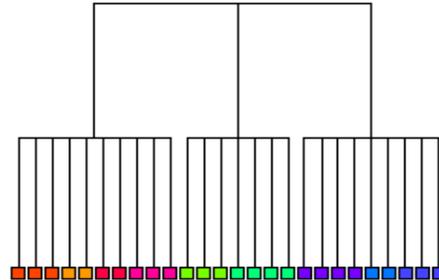
Clustering and Hierarchy



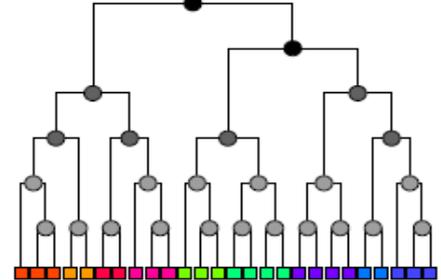
a



b

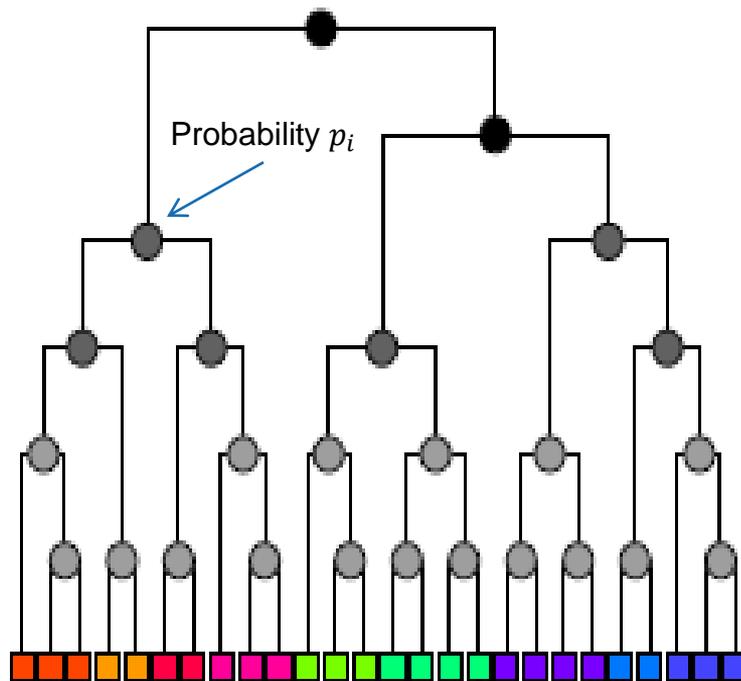


c



Hierarchical Random Graph

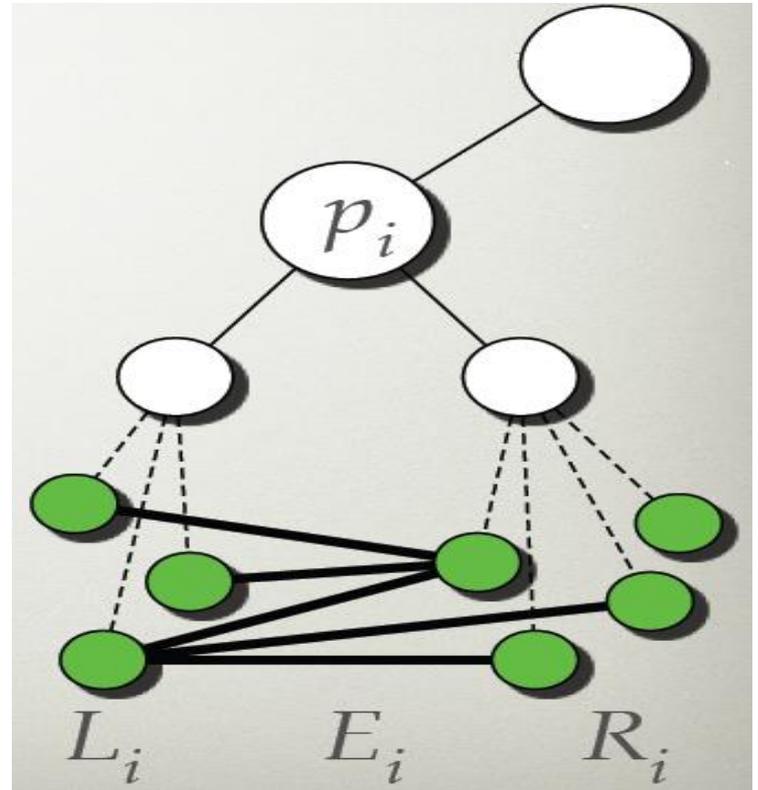
- A binary tree T : leaves are original vertices, internal nodes represent communities
- Each internal node has a probability p_i
- Two vertices are connected with probability p_i where i is their lowest common ancestor



Maximum Likelihood

- For each internal node i
 - L_i and R_i = # descendants
 - E_i = # edges between them
- Likelihood these edges exist, and not others, is

$$L_i = p^{E_i}(1 - p_i)^{L_i R_i - E_i}$$



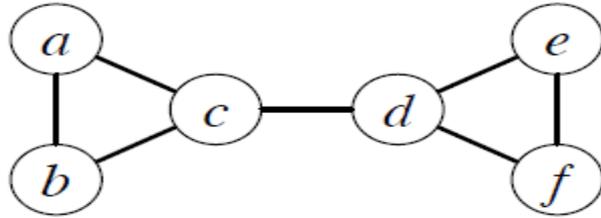
Maximum Likelihood (cont.)

- Each L_i is maximized by $p_i = \frac{E_i}{L_i R_i}$
- The likelihood of the entire tree is then
 - $L(D) = \prod_i L_i$
- The log-likelihood is
 - $\ln L(D) = -\sum_i L_i R_i h\left(\frac{E_i}{L_i R_i}\right)$

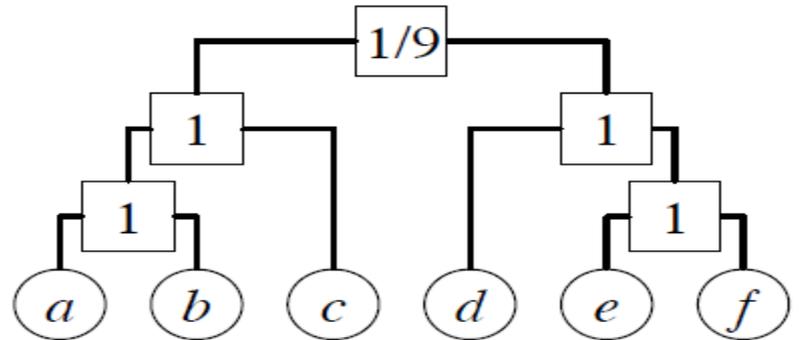
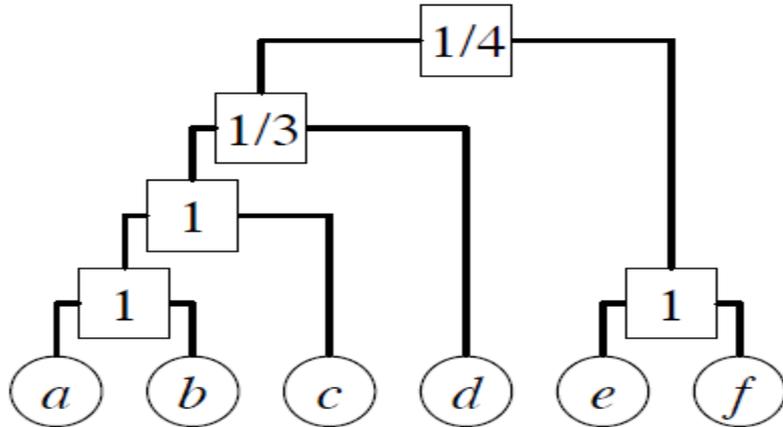
Where $h(p) = -p \ln p - (1 - p) \ln(1 - p)$

Maximum Likelihood (cont.)

$$L(T) = \left[\left(\frac{1}{3} \right) \left(\frac{2}{3} \right)^2 \right] \cdot \left[\left(\frac{2}{8} \right)^2 \left(\frac{6}{8} \right)^6 \right] = 0.0016$$

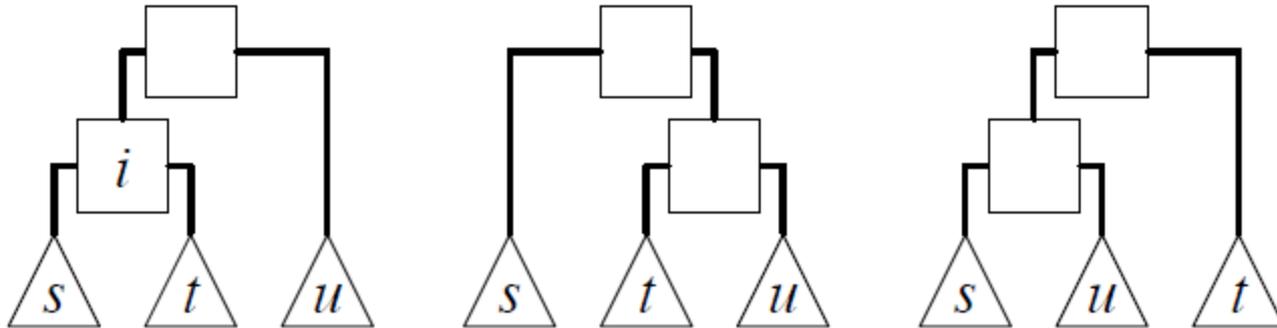


$$L(T) = \left(\frac{1}{9} \right) \left(\frac{8}{9} \right)^8 = 0.0433$$



Markov Chain

- A Markov chain Monte Carlo method to sample dendrograms D with probability proportional to their likelihood $L(D)$
- we accept the transition $D \rightarrow D'$ if $\log L = \log L(D') - \log L(D)$ is nonnegative, so that D' is at least as likely as D



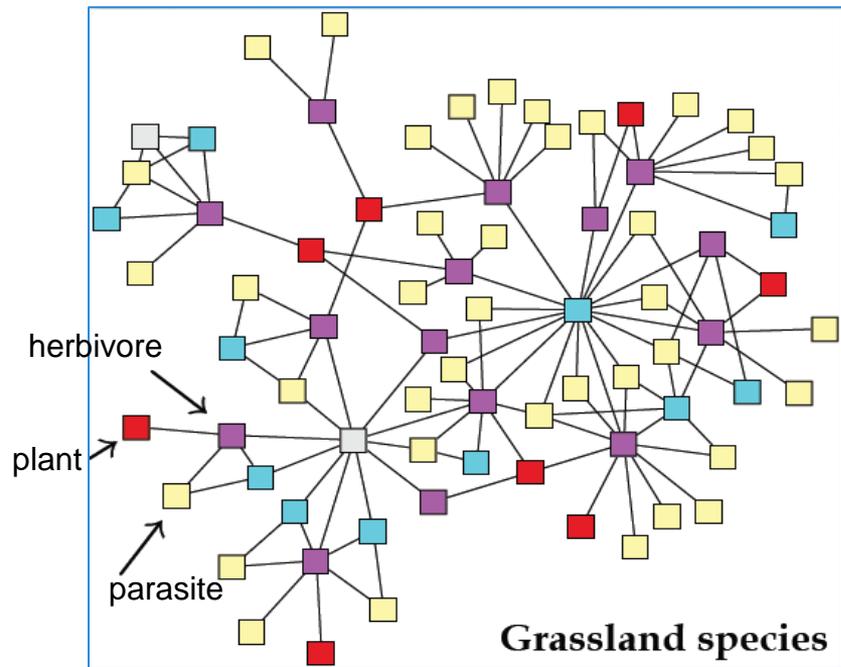
**Generating random graphs which
are statistically similar to real ones**

Resampling from the hierarchical random graph

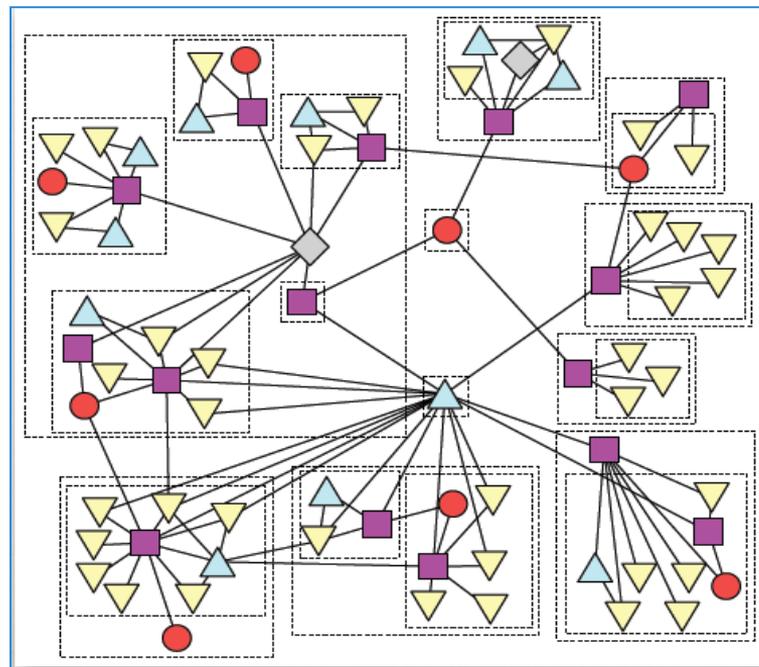
1. Initialize the Markov chain by choosing a random starting dendrogram.
2. Run the Monte Carlo algorithm until equilibrium is reached.
3. Sample dendrograms at regular intervals from those generated by the Markov chain.
4. For each sampled dendrogram D , create a resampled graph G' with n vertices by placing an edge between each of the $n(n - 1)/2$ vertex pairs (i, j) with independent probability p_r , where r is the lowest common ancestor of i and j in D

Resampling from the hierarchical random graph (cont.)

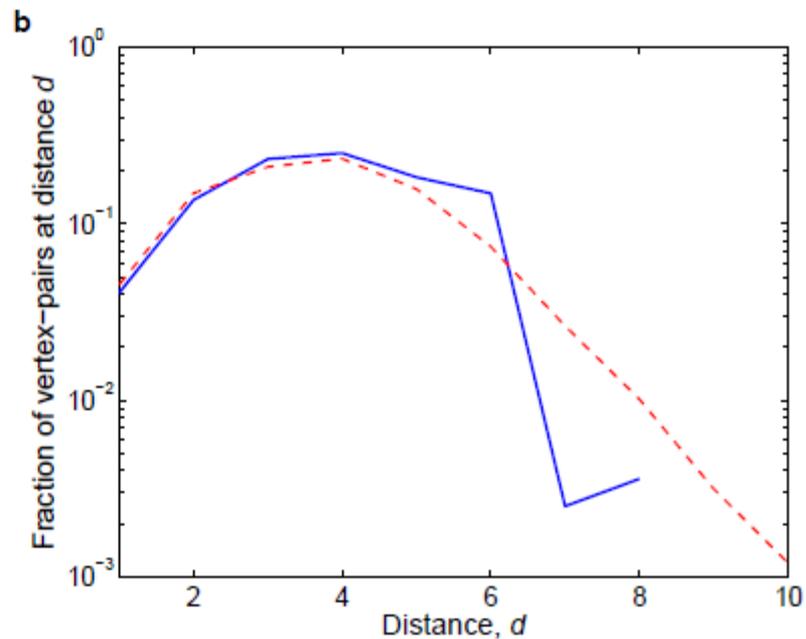
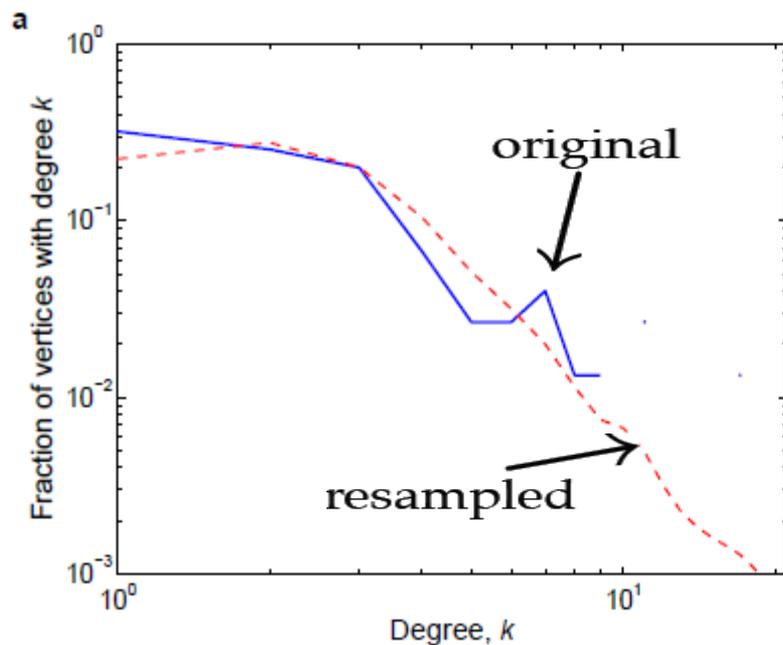
A Food Web



An Inferred Hierarchy

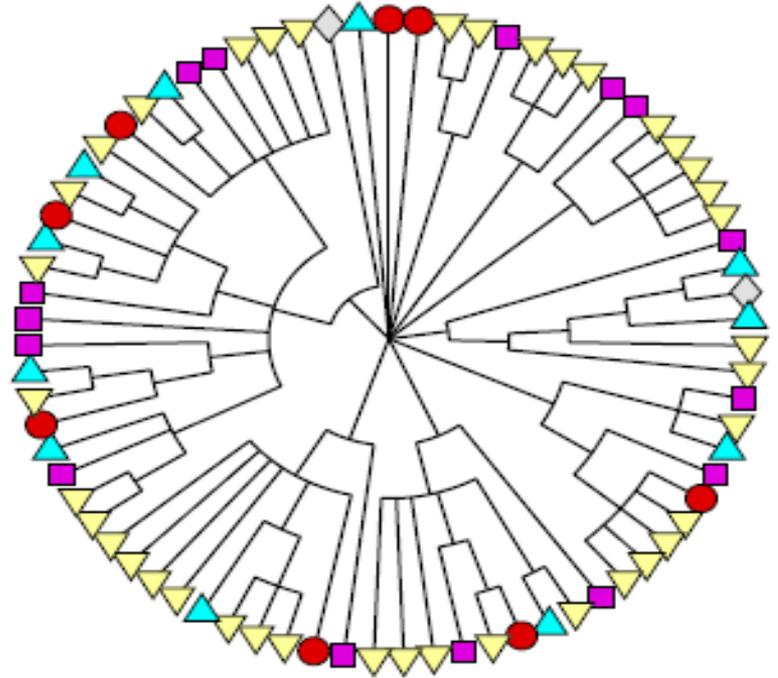


Network Statistics



Consensus Dendrograms

- Sample many trees instead of one.
- From phylogeny construction: combine these into a *consensus* dendrogram.

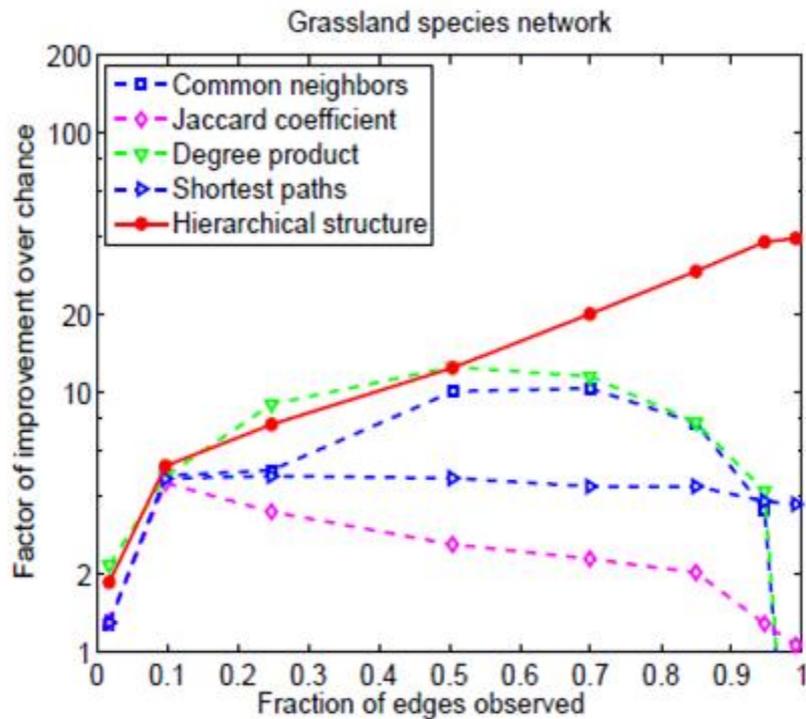


Link Prediction

Predicting missing connection

1. Initialize the Markov chain by choosing a random starting dendrogram.
2. Run the Monte Carlo algorithm until equilibrium is reached.
3. Sample dendrograms at regular intervals thereafter from those generated by the Markov chain.
4. For each pair of vertices i, j for which there is not already a known connection, calculate the mean probability p_{ij} that they are connected by averaging over the corresponding probabilities p_{ij} in each of the sampled dendrograms D .
5. Sort these pairs i, j in decreasing order of p_{ij} and predict that the highest ranked ones have missing connections.

Comparison with other link prediction algorithms



Theoretical Justification of Popular Link Prediction Heuristics

IJCAI'11 Proceedings of the Twenty-Second international joint conference on Artificial
Intelligence Vol. 3 - Purnamrita Sarkar, Deepayan Chakrabarti, Andrew W. Moore

Theoretical Justification of Popular Link Prediction Heuristics

- Some of the most popular heuristics are: most common neighbors, Adamic/Adar, and short paths
- These heuristic proved its very good performance in link prediction problem, **WHY?**

Approach (Latent Space Model)

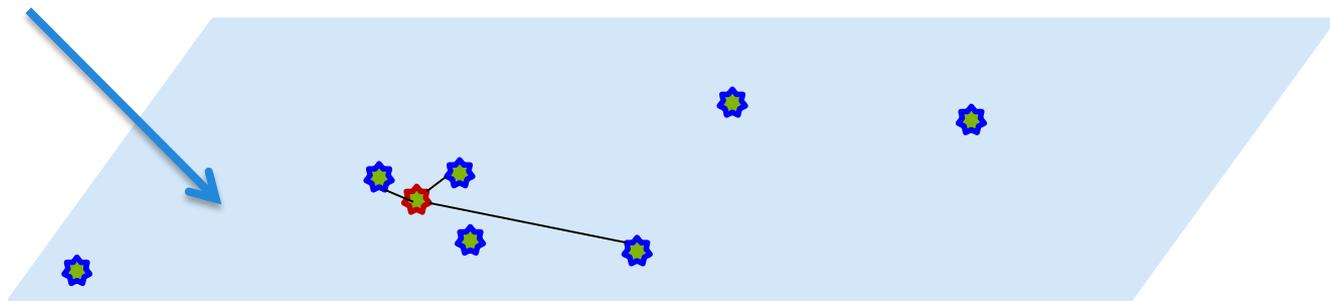
Raftery et al.'s Model:

Points close in this space are more likely to be connected.

The problem of link prediction is to find the nearest neighbor who is not currently linked to the node.

➤ Equivalent to inferring distances in the latent space

Unit volume universe



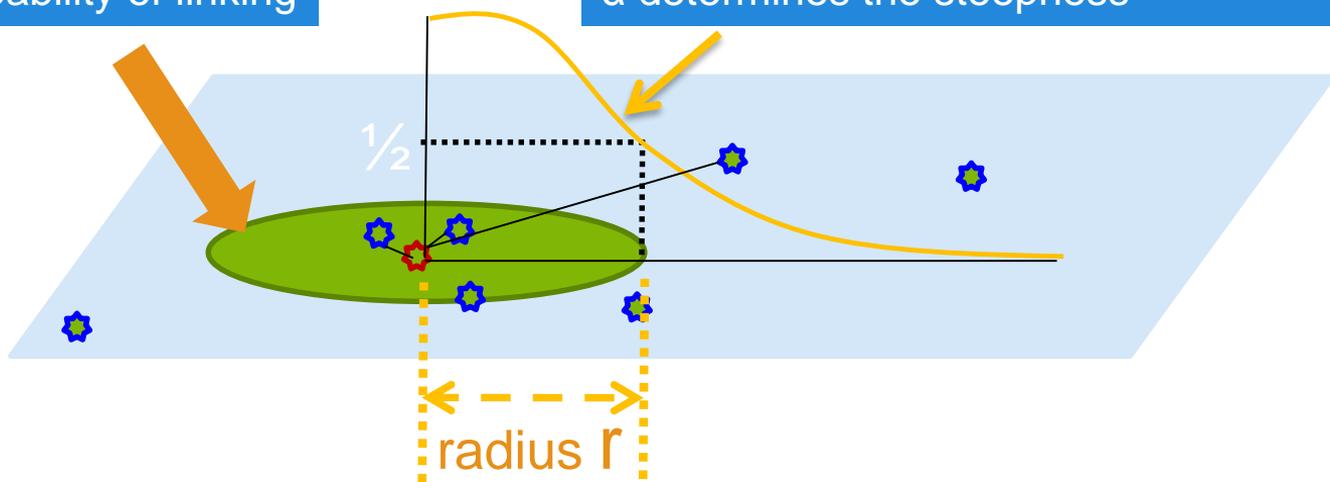
Nodes are uniformly distributed in a latent space

Link Prediction – Generative Model

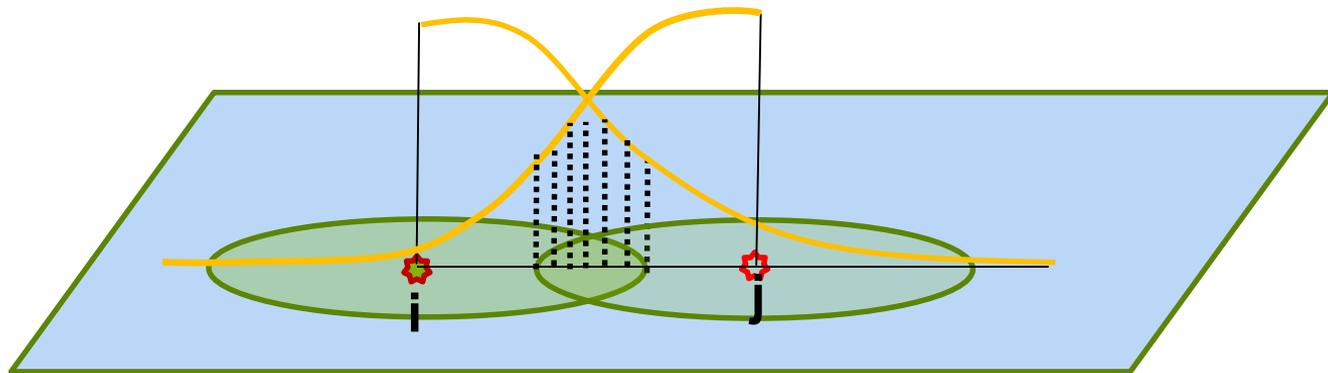
$$P(i \sim j | d_{ij}) = \frac{1}{1 + e^{\alpha(d_{ij} - r)}}$$

Higher probability of linking

α determines the steepness



Common Neighbors

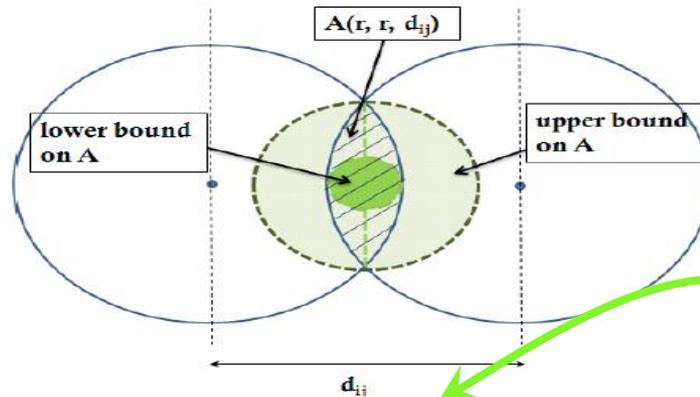


$$\Pr(i, j) = \Pr(\text{common neighbors} | d_{ij}) = \Pr(i \sim k \sim j | d_{ij})$$

$$\Pr_2(i, j) = \int \Pr(i \sim k | d_{ik}) \Pr(j \sim k | d_{jk}) P(d_{ik}, d_{jk} | d_{ij}) \partial d_{ik} \partial d_{jk}$$

Product of two logistic probabilities, integrated over a volume determined by d_{ij}

Common Neighbors



η = Number of common neighbors

Empirical
Bernstein



$$P\left[\frac{\eta}{N} - \varepsilon \leq A(r, r, d_{ij}) \leq \frac{\eta}{N} + \varepsilon\right] = 1 - 2\delta$$

Bounds on
distance



$$2r \left[1 - \left(\frac{\eta/N + \varepsilon}{V(r)} \right)^{1/D} \right] \leq d_{ij} \leq 2r \sqrt{1 - \left(\frac{\eta/N - \varepsilon}{V(r)} \right)^{2/D}}$$

$V(r)$ = volume of radius r in D dims

Common Neighbors contd..

OPT = node closest to i

MAX = node with max common neighbors with i

Theorem:

$$d_{\text{OPT}} \leq d_{\text{MAX}} \leq d_{\text{OPT}} + 2[\varepsilon/V(\mathbf{1})]^{1/D}$$

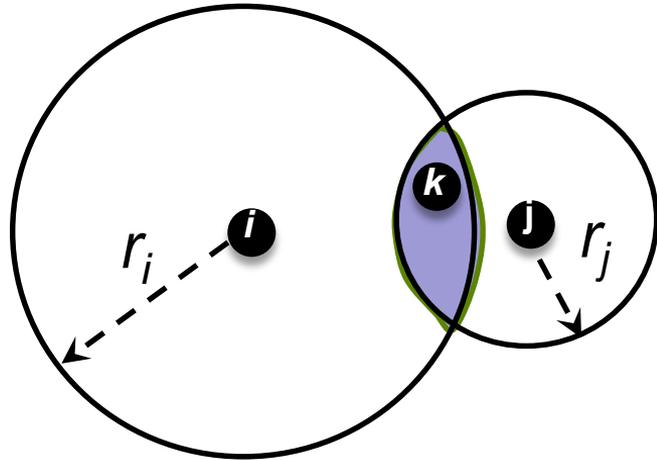
$$\varepsilon = c_1 (\text{var}_N/N)^{1/2} + c_2/(N-1)$$

Deterministic Model with Distinct Radii

- Consider nodes i , j and k with radii r_i , r_j and r_k respectively.
- Connectivity Model: $i \rightarrow j$ iff $d_{ij} \leq r_j$
(Directed graph)

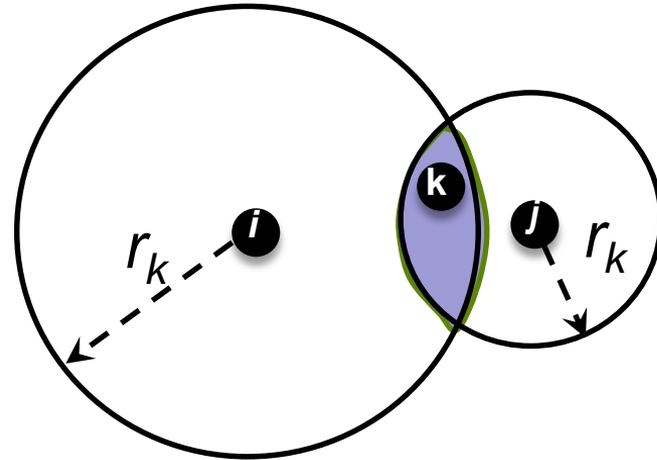
Common Neighbors – Distinct Radii

Type 1: $i \leftarrow k \rightarrow j$



$$A(r_i, r_j, d_{ij})$$

Type 2: $i \rightarrow k \leftarrow j$



$$A(r_k, r_k, d_{ij})$$

Type 2 – Common neighbors

- Constraint: $d_{ij} \leq d_{ik} + d_{jk} \leq 2r_k$
- Intuition – weighting common neighbors differently, depending on their radii.

Type 2 – Common neighbors

Example

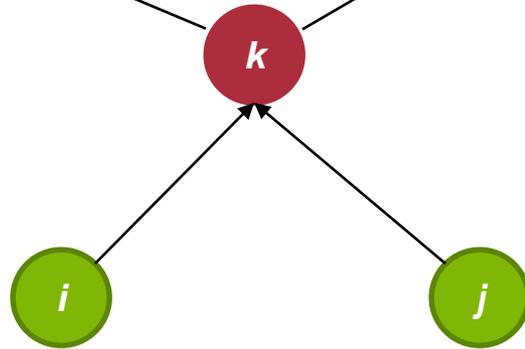
- Toy network : N_1 nodes of radius r_1 and N_2 nodes of radius r_2
- η_1 and η_2 , *common neighbors of N_1 and N_2*
- $r_1 < r_2$
- Mixture : $A(r_1, r_1, d_{ij})$, $A(r_2, r_2, d_{ij})$

Type 2 – Common neighbors

Example contd..

$$\eta_1 \sim \text{Bin}(N_1, A(r_1, r_1, d_{ij}))$$

$$\eta_2 \sim \text{Bin}(N_2, A(r_2, r_2, d_{ij}))$$

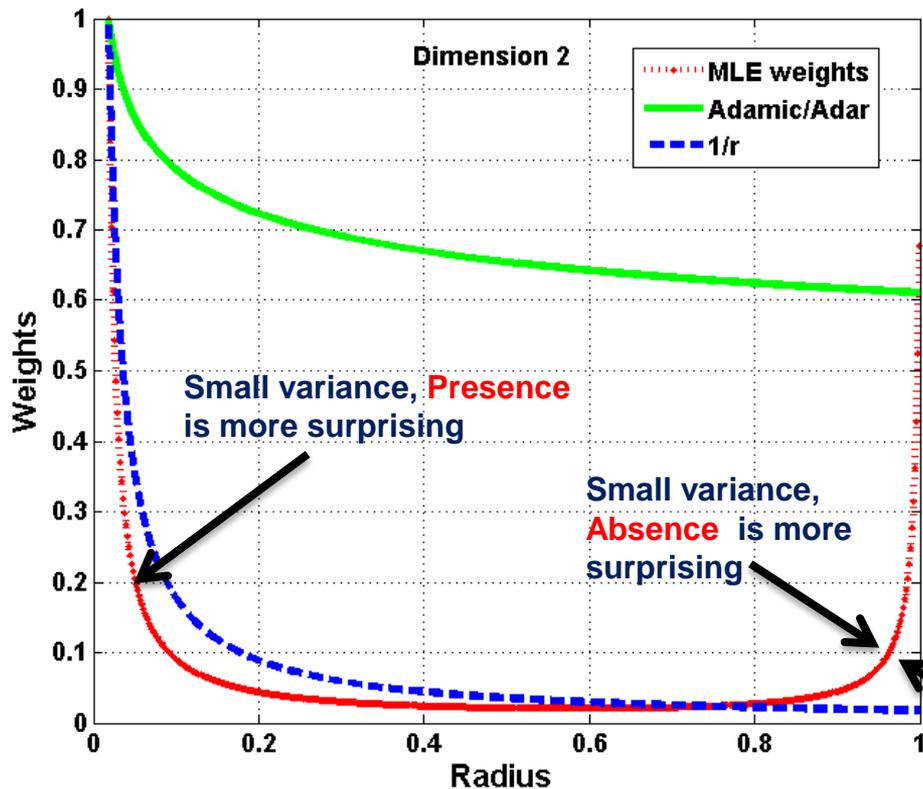


Maximize $P(\eta_1, \eta_2 \mid d_{ij}) =$ Product of two binomials
 $w(r_1) E(\eta_1 \mid d^*) + w(r_2) E(\eta_2 \mid d^*) = w(r_1)\eta_1 + w(r_2)\eta_2$

$RHS \uparrow \rightarrow LHS \uparrow \rightarrow d^* \downarrow$

Type 2 – Common neighbors

Example contd..



$$w(r) \approx \frac{\text{Constant}}{r} \approx \frac{\text{Constant}}{\text{deg}(i)^{1/D}}$$

$$\text{Adamic/Adar} = \sum_{k \in N(i) \cap N(j)} \frac{1}{\log(\text{deg}(k))}$$

r is close to max radius

Summary

- Simple heuristic of counting common neighbors often outperforms more complicated heuristics
- Weighted counts of common neighbors outperforms the unweighted count.

Supervised Random Walks

WSDM '11 - Proceedings of the Fourth ACM International Conference on
Web Search and Data Mining - Lars Backstrom, Jure Leskovec

Problem

- Friendship is important on social networks
- How to predict the future interaction?
- How to recommend potential friends to new user?



People You May Know



Ankita Banerjee

Narula Institute Of Technology
5 mutual friends

+1 Add Friend



Monalisha Roy

Medinipur, West Bengal, India
3 mutual friends

+1 Add Friend



Imtiaz Ali

Islamia High School, Kolkata (Calcutta)
4 mutual friends

+1 Add Friend



Shamail Shafique

St augustine day school
3 mutual friends

+1 Add Friend

Link Prediction / Link Recommendation

Motivation

- Predicting future interaction brings direct business consequences – Facebook, Myspace
- Possible collaborations, JV
- Beyond social networks: predicting coauthor/collaborations/protein-protein interactions/ security

Challenges

- Real networks are extremely sparse
- Modeling social networks using intrinsic features of the network (node and edge level attributes)

General approaches

- Classification
- Ranking nodes – PageRank, Personalized PageRank, Random Walk with Restarts

Our approach

- Combination – classification, node ranking
- Based on the Supervised Random Walks
 - Combines the network structure with the characteristics of nodes and edges
- Develop an algorithm to estimate the edge strength
 - bias a PageRank-like random walk to visits given nodes more often

Random Walk with Restarts (RWR)

- Measures the proximity of nodes in a graph (Pan et al, 2004)
- Start a random walk to a node s and compute the proximity of each other node to node s
- Restart probability : α

Problem formulation

- Given $G(V, E)$
- A start point s , learning candidate $C = \{c_i\}$
- Destination nodes $D = \{d_1, \dots, d_k\}$, no-link nodes $L = \{l_1, \dots, l_n\}$, $C = D \cup L$
- For edge (u, v) we compute the strength $a_{uv} = f_w(\psi_{uv})$

Optimization

- $$\min_w F(w) = \|w\|^2$$
 such that
$$\forall d \in D, l \in L : p_l < p_d$$
 (1)
- $$\min_w F(w) = \|w\|^2 + \lambda \sum_{d \in D, l \in L} h(p_l - p_d)$$
 (2)
- Eq(1) is *hard* version and Eq(2) is *soft* version of the optimization problem.

Algorithm

Initialize PageRank scores p and partial derivatives $\frac{\partial p_u}{\partial w_k}$:

foreach $u \in V$ **do** $p_u^{(0)} = \frac{1}{|V|}$

foreach $u \in V, k = 1, \dots, |w|$ **do** $\frac{\partial p_u}{\partial w_k}^{(0)} = 0$

$t = 1$

while not converged do

foreach $u \in V$ **do**

$p_u^{(t)} = \sum_j p_j^{(t-1)} Q_{ju}$

$t = t + 1$

$t = 1$

foreach $k = 1, \dots, |w|$ **do**

while not converged do

foreach $u \in V$ **do**

$\frac{\partial p_u}{\partial w_k}^{(t)} = \sum_j Q_{ju} \frac{\partial p_j}{\partial w_k}^{(t-1)} + p_j^{(t-1)} \frac{\partial Q_{ju}}{\partial w_k}$

$t = t + 1$

return $\frac{\partial p_u}{\partial w}^{(t-1)}$

$$Q_{uv} = (1 - \alpha)Q'_{uv} + \alpha \mathbf{1}(v = s)$$

$$Q'_{uv} = \begin{cases} \frac{a_{uv}}{\sum_w a_{uw}} & \text{if } (u, v) \in E \\ 0 & \text{otherwise} \end{cases}$$

Algorithm 1: Iterative power-iterator like computation of PageRank vector p and its derivative $\frac{\partial p_u}{\partial w}$.

Experiments on Synthetic data

- A scale-free graph G with 10,000 nodes
- Evaluated by classification accuracy
- Strength function :

$$a_{uv} = \exp(\psi_{uv1} - \psi_{uv2})$$

Experiments on Synthetic data

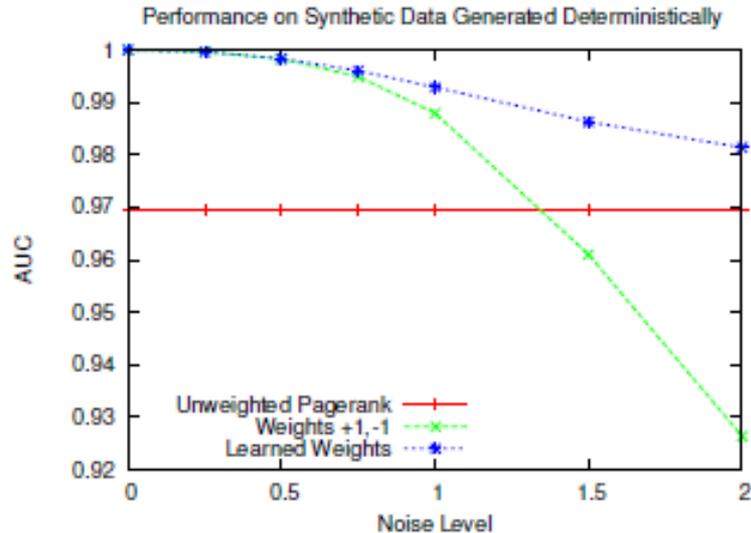


Figure 1: Experiments on synthetic data. Deterministic D .

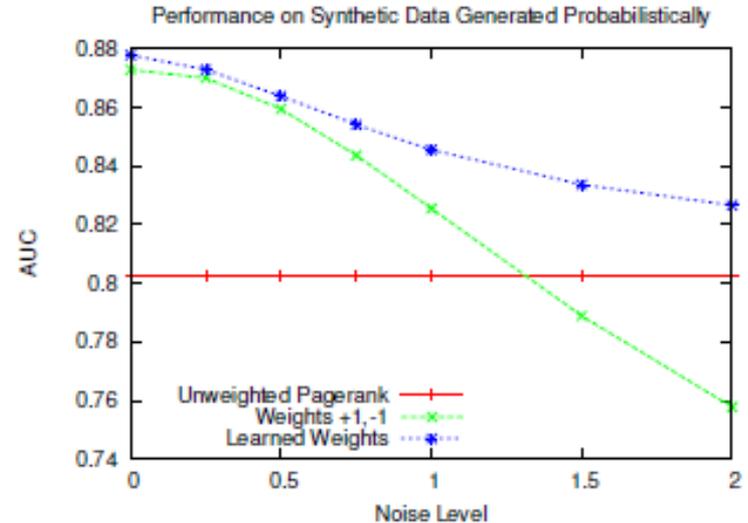


Figure 2: Experiments on synthetic data. Probabilistic D .

Note: AUC – Area under the ROC curve. 1.0 means perfect classification and 0.5 means random guessing.

Experimental setup

- Real Datasets:
 - Co-authorship networks : arXive database dataset
 - Facebook network : Iceland

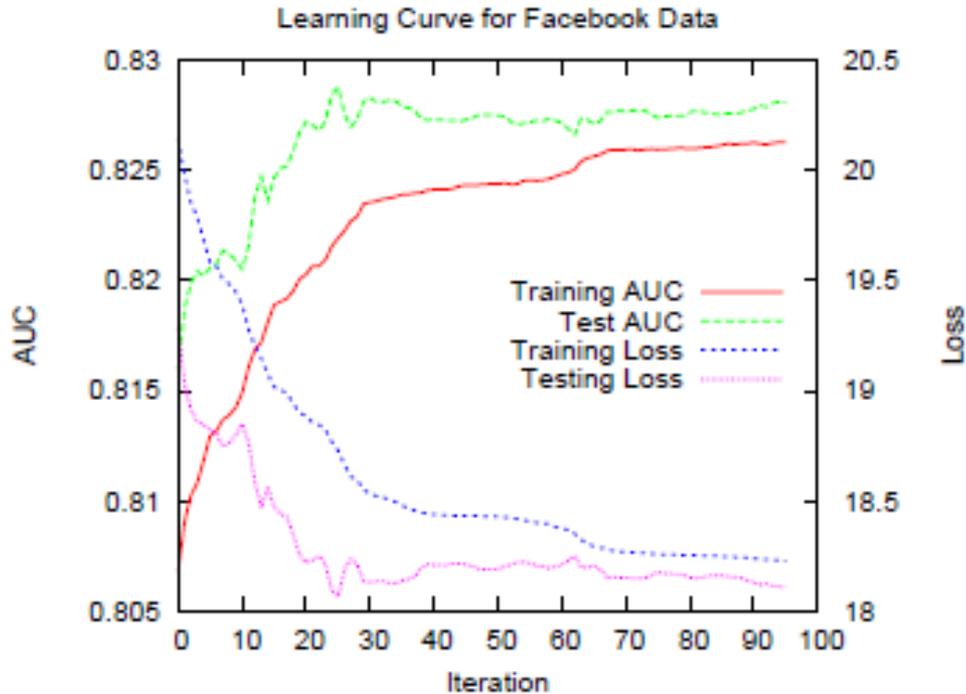
Experiments on Real datasets

- Four co-authorship networks and the Facebook network of Iceland

	N	E	S	\bar{D}	\bar{C}	\bar{D}/\bar{C}
Astro-Ph	19,144	198,110	1,123	18.0	775.6	0.023
Cond-Mat	23,608	94,492	140	9.1	335.5	0.027
Hep-Ph	12,527	118,515	340	29.2	345.3	0.084
Hep-Th	10,700	25,997	55	6.3	110.5	0.057
Facebook	174,000	29M	200	43.6	1987	0.022

Table 1: Dataset statistics. N, E : number of nodes and edges in the full network, S : number of sources, \bar{C} : avg. number of candidates per source, \bar{D} : avg. number of destination nodes.

Learning curve for Facebook data



Performance of Supervised Random Walks as a function of the number of steps of parameter estimation procedure.

Results

Learning Method	AUC	Prec@20
Random Walk with Restart	0.63831	3.41
Adamic-Adar	0.60570	3.13
Common Friends	0.59370	3.11
Degree	0.56522	3.05
DT: Node features	0.60961	3.54
DT: Network features	0.59302	3.69
DT: Node+Network	0.63711	3.95
DT: Path features	0.56213	1.72
DT: All features	0.61820	3.77
LR: Node features	0.64754	3.19
LR: Network features	0.58732	3.27
LR: Node+Network	0.64644	3.81
LR: Path features	0.67237	2.78
LR: All features	0.67426	3.82
SRW: one edge type	0.69996	4.24
SRW: multiple edge types	0.71238	4.25

Table1 : Hep-Ph co-authorship network

Learning Method	AUC	Prec@20
Random Walk with Restart	0.81725	6.80
Degree	0.58535	3.25
DT: Node features	0.59248	2.38
DT: Path features	0.62836	2.46
DT: All features	0.72986	5.34
LR: Node features	0.54134	1.38
LR: Path features	0.51418	0.74
LR: All features	0.81681	7.52
SRW: one edge type	0.82502	6.87
SRW: multiple edge types	0.82799	7.57

Table 2: Results for Facebook datasets

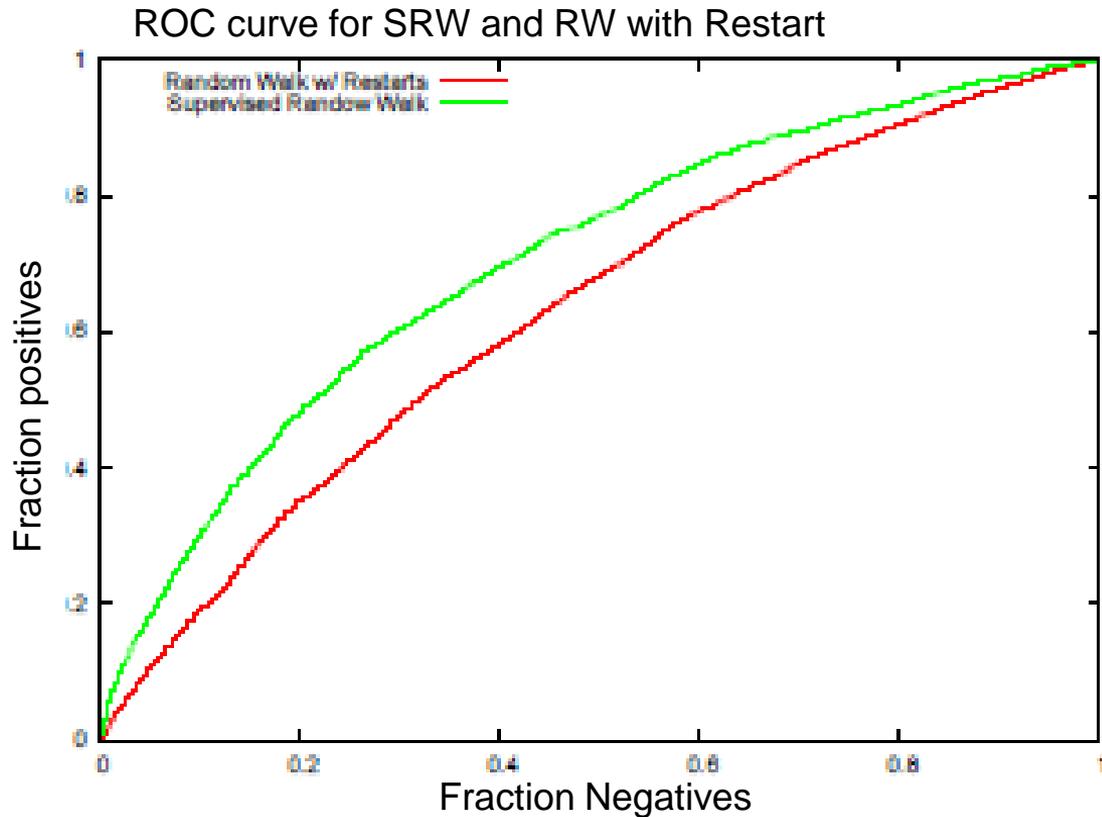
Supervised Random Walk : AUC (0.7 – 0.8), Precision at top 20 (4.2 – 7.6)

Results contd.

Dataset	AUC		Prec@20	
	SRW	LR	SRW	LR
Co-authorship Astro-Ph	0.70548	0.67639	2.55	2.15
Co-authorship Cond-Mat	0.74173	0.71672	2.54	2.61
Co-authorship Hep-Ph	0.71238	0.67426	4.18	3.82
Co-authorship Hep-Th	0.72505	0.69428	2.59	2.61
Facebook (Iceland)	0.82799	0.81681	7.57	7.52

Table 3: Results all datasets

ROC Curve of Astro-Ph test data.



Conclusion

- Supervised Random Walks has great improvement over Random Walks.
- Outperforms supervised machine learning techniques
- Combines rich node and edge features with the structure of the network
- Apply to: recommendations, anomaly detection, missing link, expertise search and ranking

Questions



References

1. A. Clauset, C. Moore, M.E.J. Newman. Hierarchical structure and the prediction of missing links in networks. Nature, 2008.
<http://www-personal.umich.edu/~mejn/papers/cmn08.pdf>
2. Theoretical Justification of Popular Link Prediction Heuristics, by P. Sarkar, D. Chakrabarti, and A. W. Moore, invited to IJCAI 2011
<http://www.cs.cmu.edu/~deepay/mywww/papers/ijcai11-linkpred.pdf>
3. L. Backstrom, J. Leskovec. Supervised Random Walks: Predicting and Recommending Links in Social Networks. In Proc. WSDM, 2011.
<http://cs.stanford.edu/people/jure/pubs/linkpred-wsdm11.pdf>

Thank You