CentraleSupélec

ESSEC BUSINESS SCHOOL

Network Science Analytics

Option Applied Math and M.Sc. in DSBA

Lecture 5A

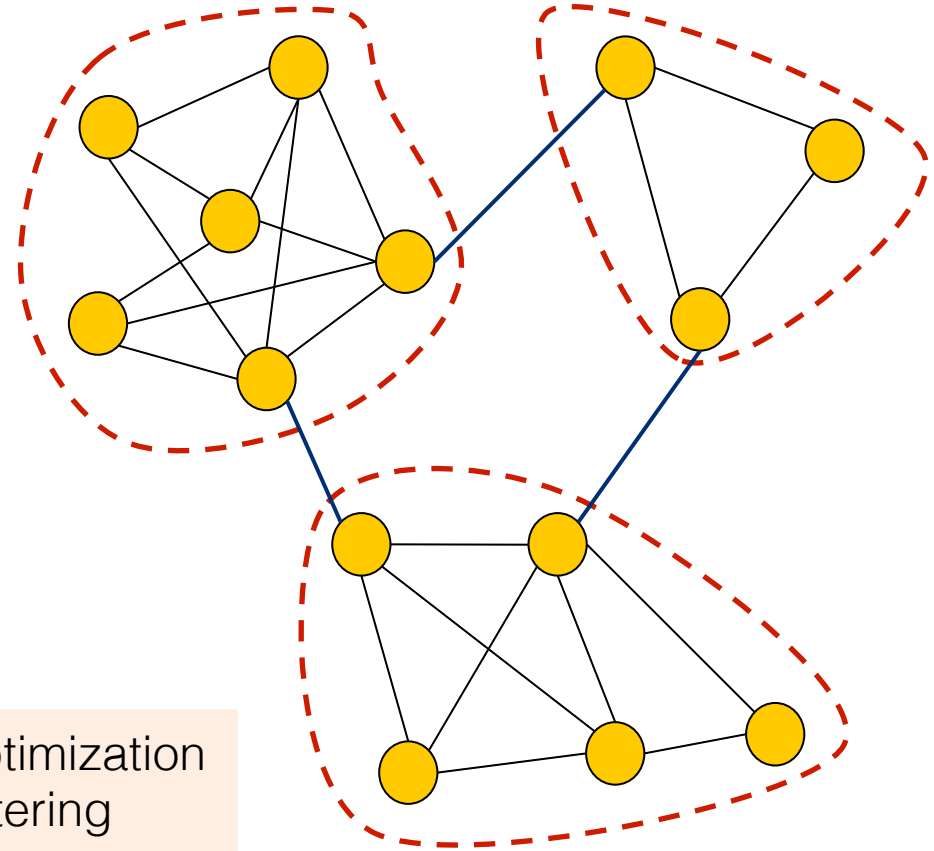
Link prediction in networks

Fragkiskos Malliaros

Friday, March 2, 2018

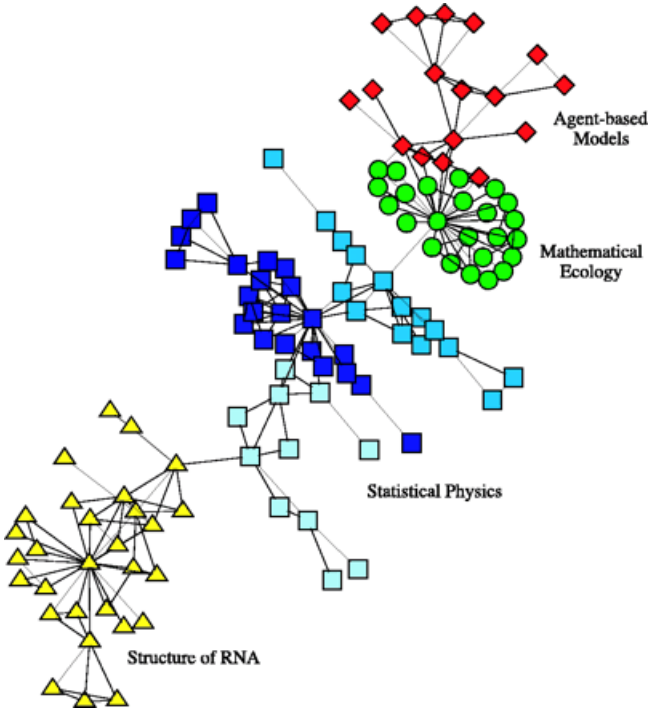
Community structure of large-scale graphs

Community Structure

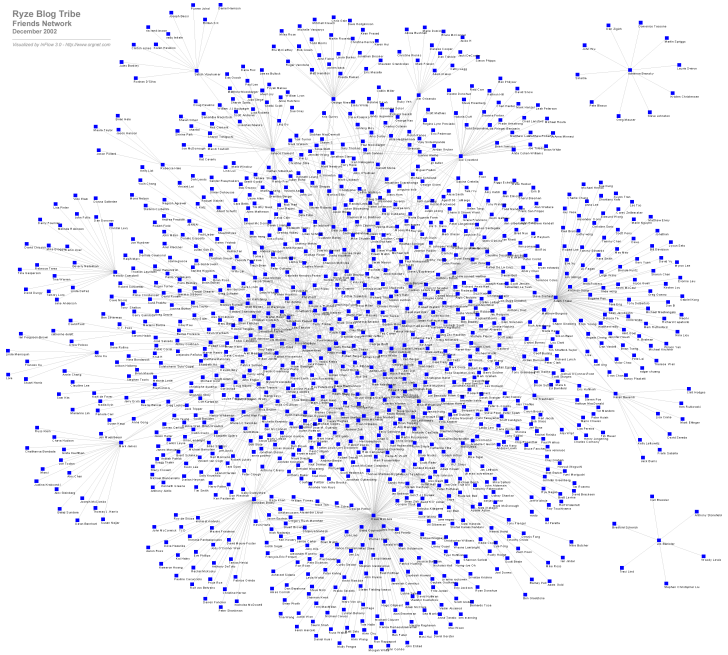


- Modularity optimization
- Spectral clustering
- …

Community Structure in Small vs. Large Graphs



Small scale collaboration network (Newman)



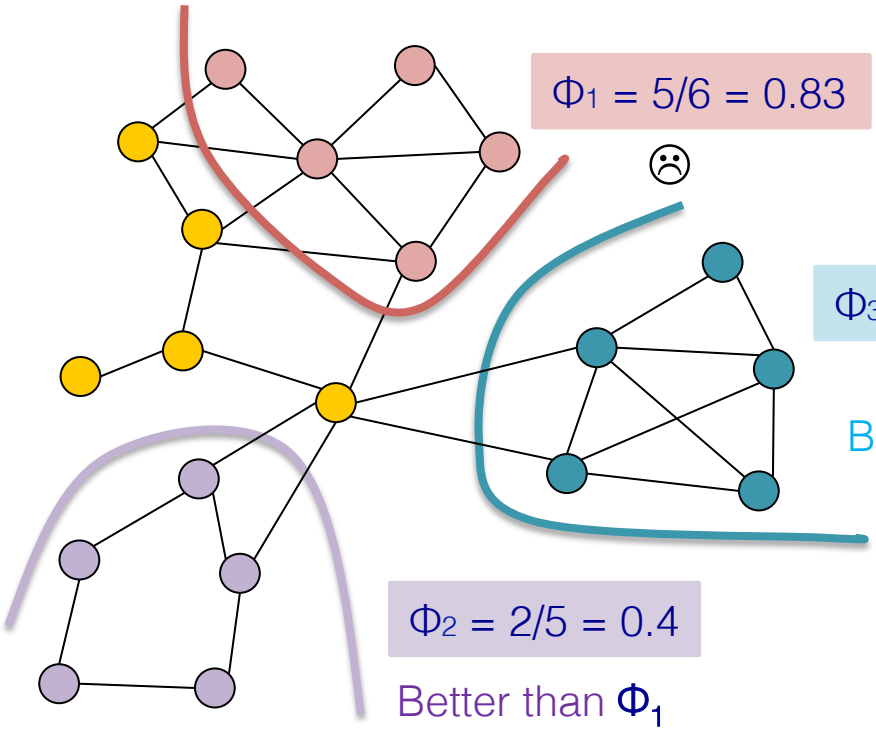
Blog network
http://www.ryze.com

Examine the Structural Differences

- Use conductance Φ(S) as a community evaluation measure
  - Smaller value for conductance implies better community-like properties [Leskovec et al. '09]

Φ(S) = # outgoing edges / # edges within

lower is better



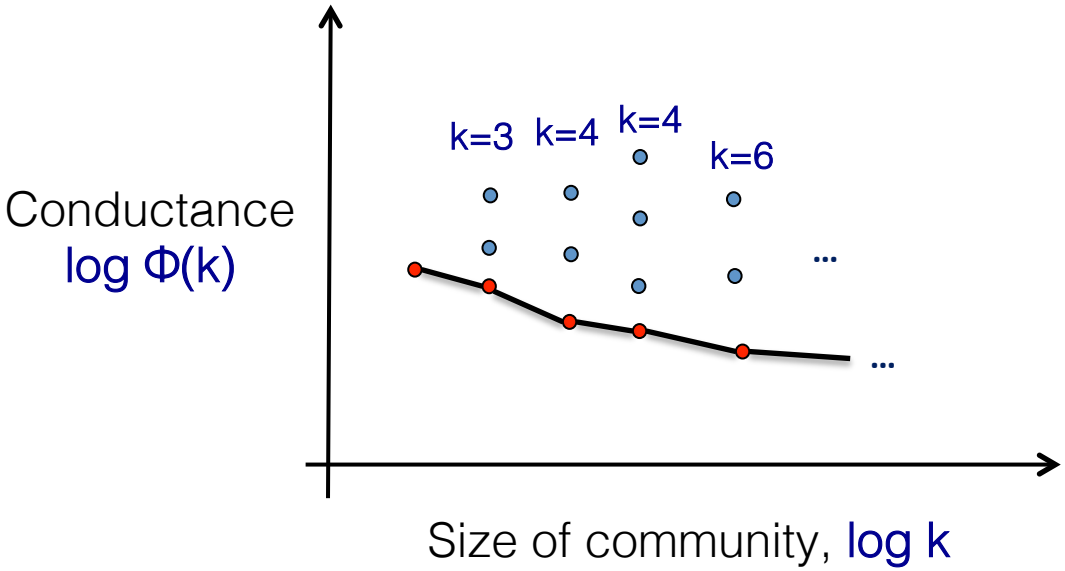
Find the best community of k = 5 nodes

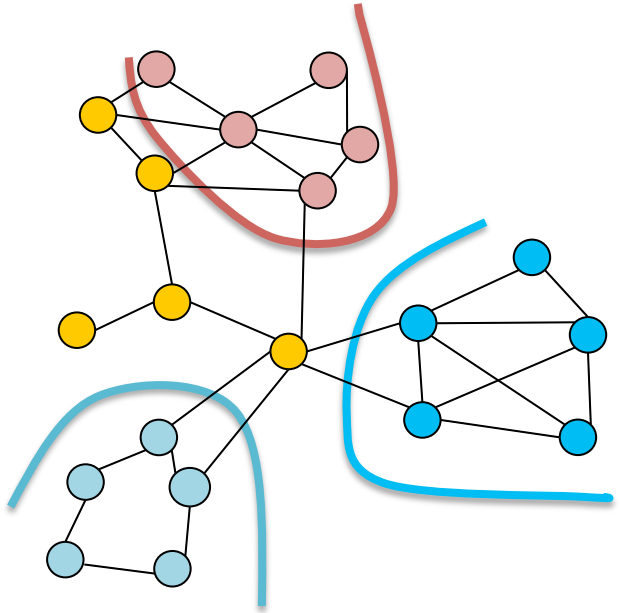
$$\Phi(k) = \min_{S \subset V, |S|=k} \Phi(S)$$

Example by J. Leskovec, ICML 2009

Network Community Profile plot

- Network Community Profile (NCP) plot [Leskovec et al. '09]
  - Plot the best conductance score (minimum) Φ(k) for each community size k



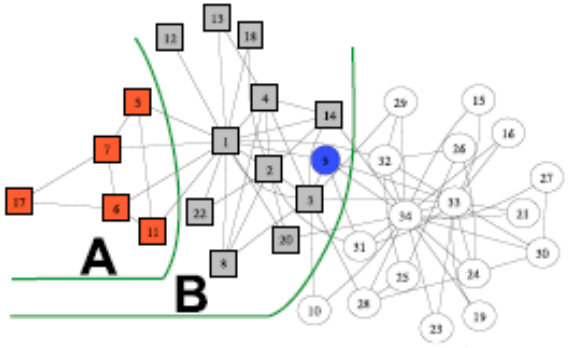


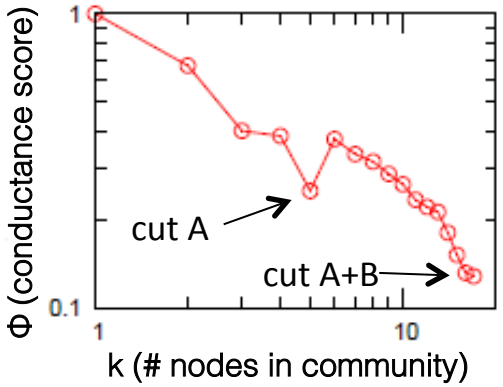
NCP plot of real graphs

?

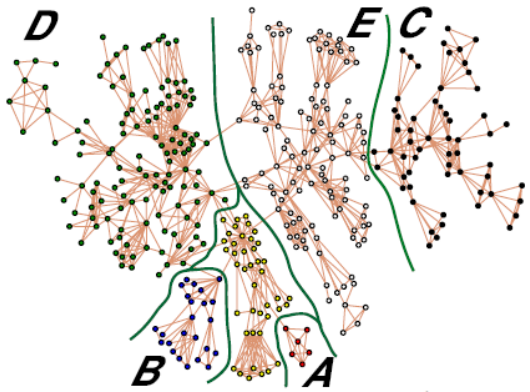
NCP Plot Examples

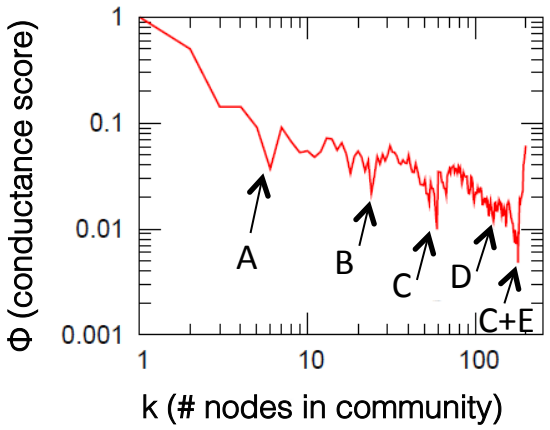
Small scale networks





Zachary's karate club social network

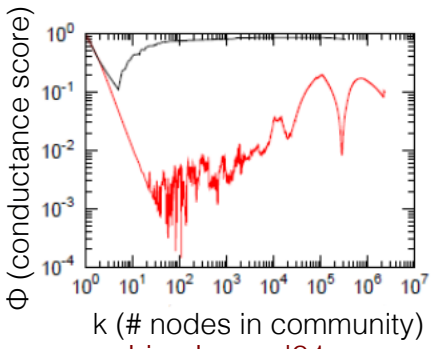




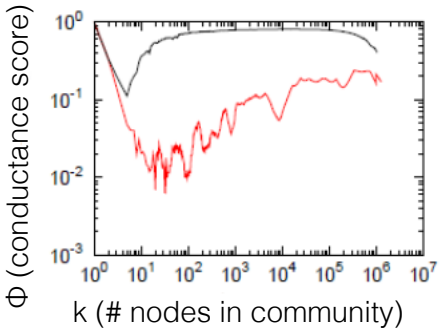
Newman's collaboration network

[Leskovec et al. '09]

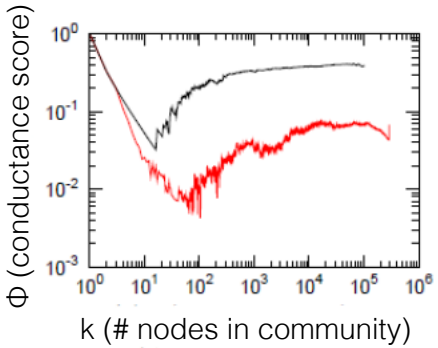
NCP Plot of Large Real Graphs



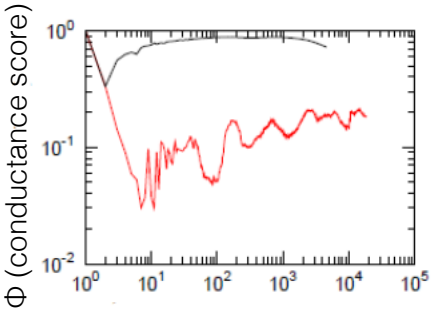
LiveJournal01



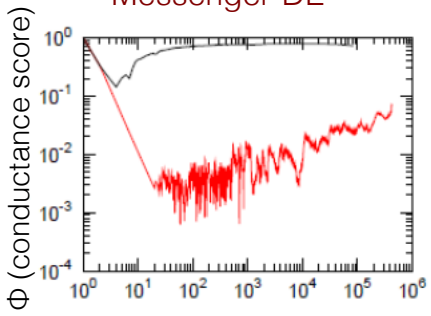
Messenger-DE



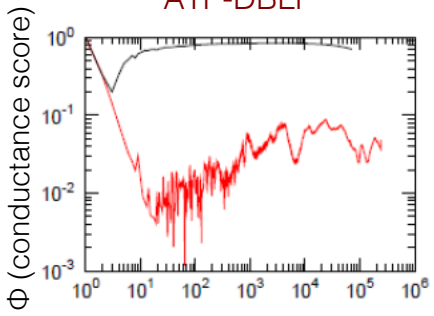
ATP-DBLP



Cit-Hep-Th



Web-Google



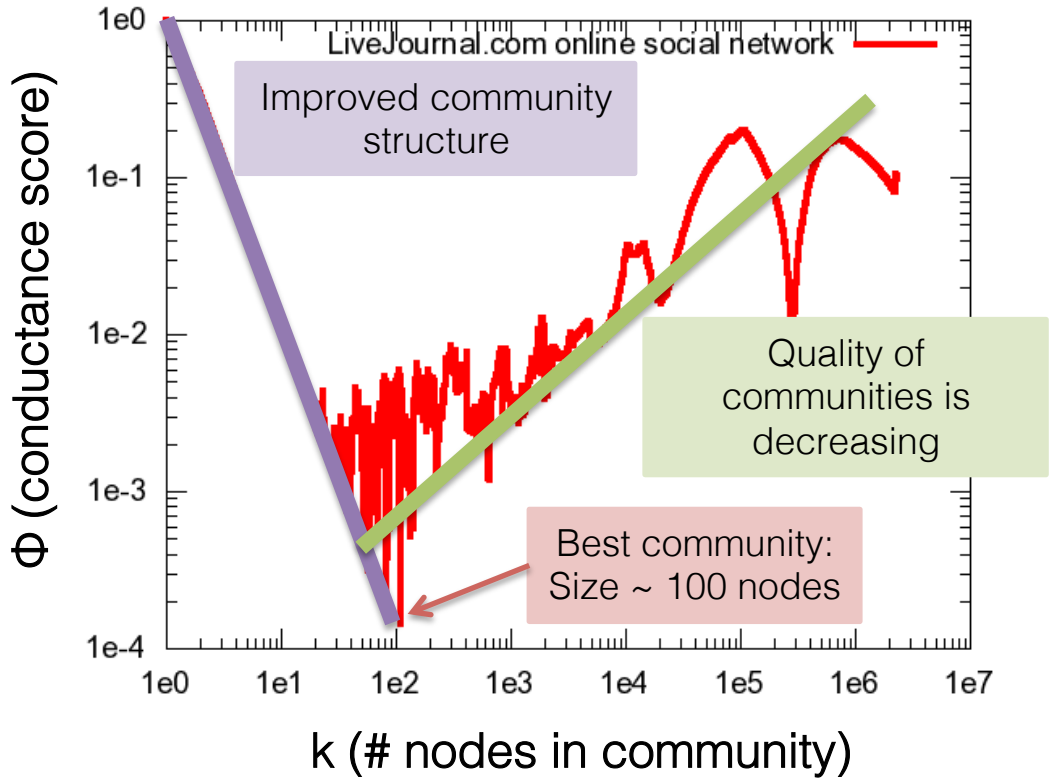
Amazon-All

Large scale networks

Any common property?

[Leskovec et al. '09]

Observation in Large Graphs

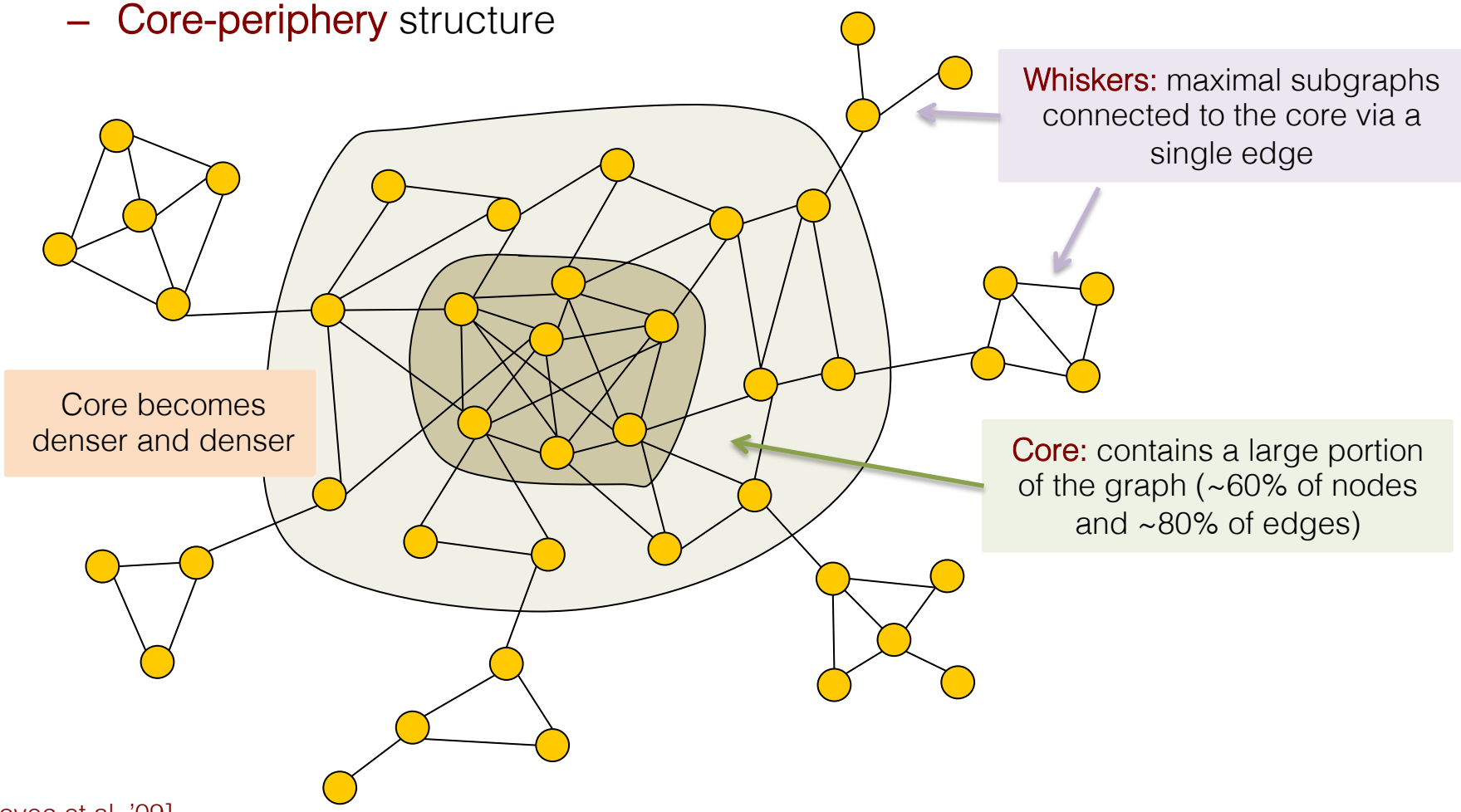


LiveJournal social network
|V| = 5M, |E| = 42M

Figure: http://snap.stanford.edu/ncp/
Slide by J. Leskovec, ICML 2009

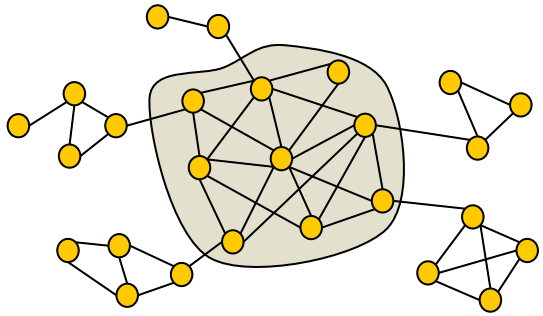
Explanation: Core-Periphery Structure

- How can we explain the observed structure of large graphs?
  - Core-periphery structure



[Leskovec et al. '09]

Core-Periphery Structure



- Core-periphery structure
  - Core
  - Whiskers
- Whiskers
  - Non-trivial structure → more than random (shape and size)
- Q: What is happening if we remove whiskers (periphery) from graphs?
  - Almost nothing. The whiskers are replaced by 2-whiskers (subgraphs connected to the core with 2 edges)
- The core itself has core-periphery structure
- Important point: Whiskers are also responsible for the best communities in large graphs (lowest point of NCP plot)

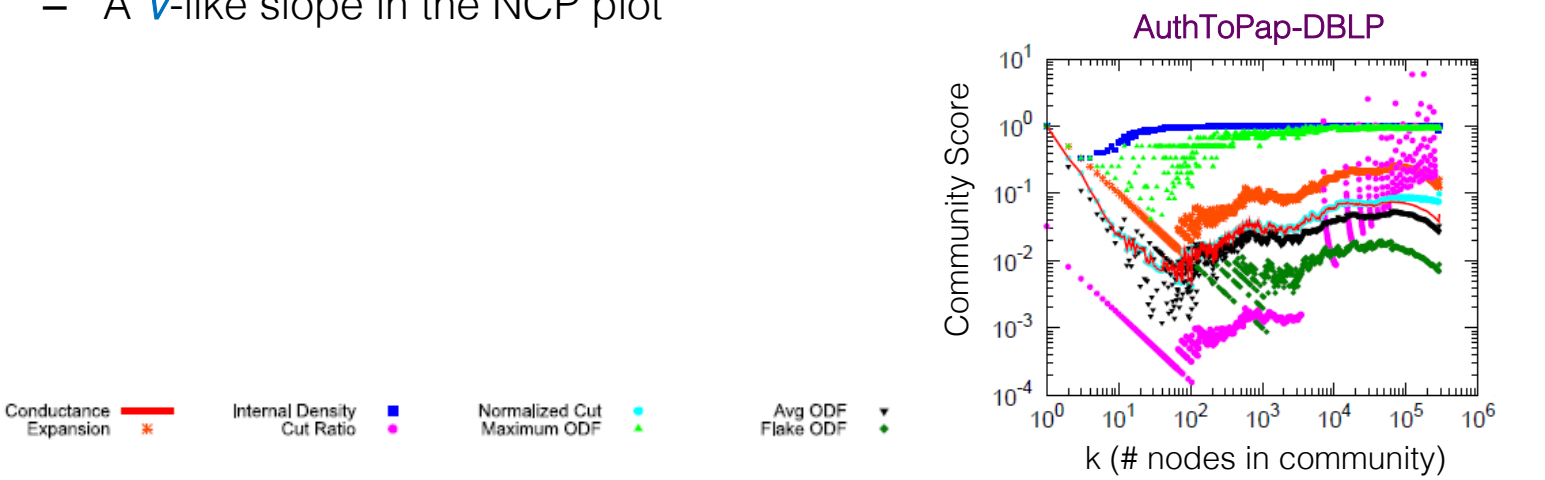
Match the observations made using the properties of the Kronecker graphs

Similar Structural Observations

- Jellyfish model for the Internet topology [Tauro et al. '01]
- Min-cut plots [Chakrabarty et al. '04]
  - Perform min-cut recursively
  - Plot the relative size of the minimum cut
- Robustness of large scale social networks [Malliaros, Megalooikonomou, Faloutsos '12, '15]
  - Robustness estimation based on the expansion properties of graphs
  - Social networks are expected to have low robustness due to the existence of communities → the (small number of) inter-community edges will act as bottlenecks
  - Large scale social graphs tend to be extremely robust
  - Structural differences (in terms of robustness and community structure) between different scale graphs

Clustering Algorithms and Objective Criteria

- Question 1: Is the observed property an effect of the used community detection algorithm (Metis + flow based method)?
  - A: No. The qualitative shape of the NCP plot is the same, regardless of the community detection algorithm [Leskovec et al. '09]
- Question 2: Is the observed property an effect of the conductance community evaluation measure?
  - A: No. All the objective criteria that based on both internal and external connectivity, show an almost similar qualitatively behavior [Leskovec et al. '10]
  - A *V*-like slope in the NCP plot



Conclusions

- Large scale real-world graphs
  - Core-periphery structure
  - No large, well defined communities
  - Structural differences between different scale graphs

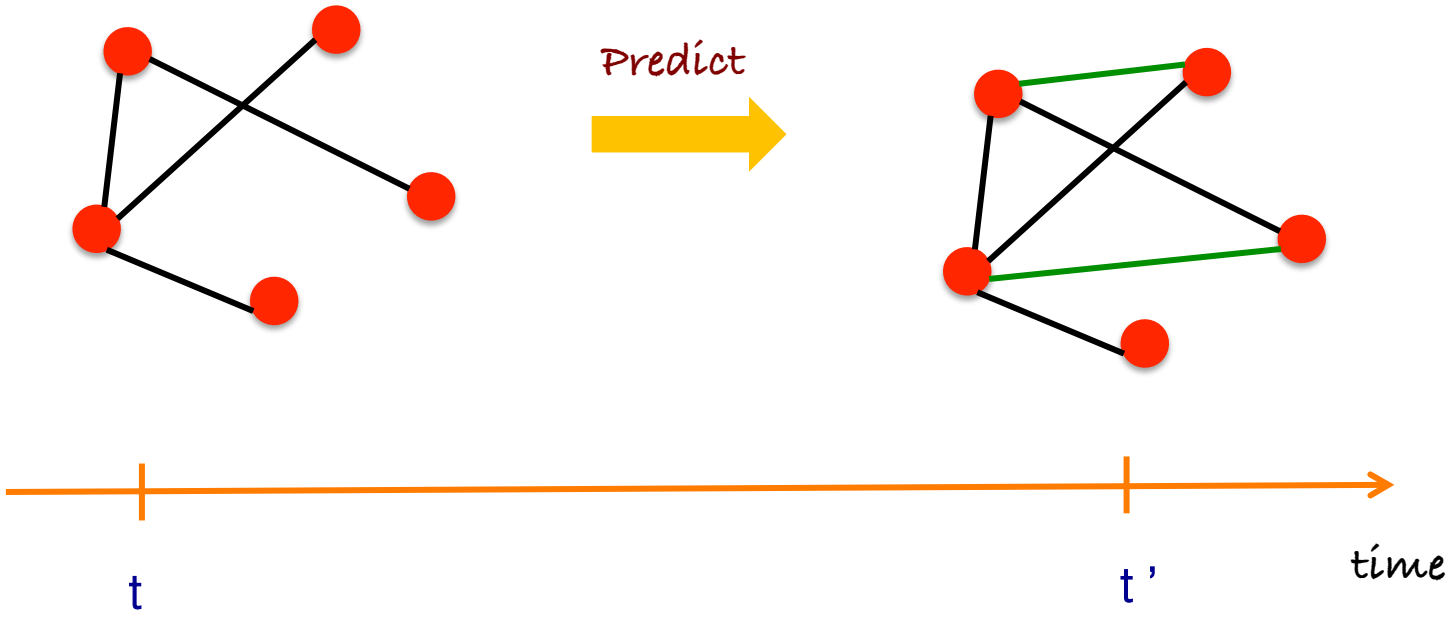
- Community detection algorithms should take into account these structural observation
  - Whiskers correspond to the best (conductance-based) communities
  - Need larger high-quality clusters?
  - Bag of whiskers: union of disjoint (disconnected) whiskers are mainly responsible for the best high-quality clusters of larger size (above 100)

Survey Articles

- S. Fortunato. Community detection in graphs. Physics Reports 486, 75-174, 2010
- S. E. Schaeffer. Graph clustering. Computer Science Review, 1(1): 27–64, 2007
- F. D. Malliaros and M. Vazirgiannis. Clustering and community detection in directed networks: A survey. Physics Reports 533, 95-142, 2013
- S. Fortunato and D. Hric. Community detection in networks: a user guide. Physics Reports 659, 1-44, 2016
- S. Papadopoulos, Y. Kompatsiaris, A. Vakali, and P. Spyridonos. Community detection in Social Media: Performance and application considerations. Data Min Knowl Disc, 24:515–554, 2012

Link prediction

Link Prediction



In this lecture

- Unsupervised link prediction
- Supervised link prediction
- Learning to predict links based on supervised random walks

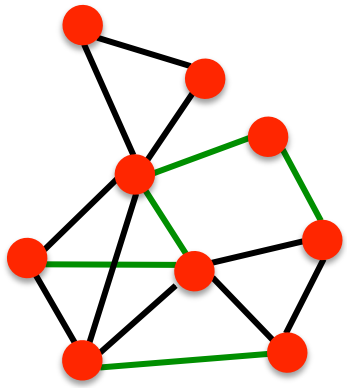
Motivation

- Recommending new friends in online social networks
- Suggesting interactions between the members of a company/ organization that are external to the hierarchical structure of the organization itself
- Suggesting collaborations between researchers based on co-authorship
- Predicting connections between members of terrorist organizations who have not been directly observed to work together
- Overcoming the data-sparsity problem in recommender systems using collaborative filtering

Link Prediction in Networks (1/2)

- The link prediction task
  - Given $G[t_0, t_0']$ a graph on edges up to time t_0', output a ranked list L of links (not in $G[t_0, t_0']$) that are predicted to appear in $G[t_1, t_1']$

- Evaluation
  - $n=|E_{new}|$: # of news edges that appear during the test period $[t_1, t_1']$
  - Take top n elements of list L and count the correct edges



$G[t_0, t_0']$
$G[t_1, t_1']$

[Liben-Nowell and Kleinberg '03]

Link Prediction in Networks (2/2)

- Prediction for a subset of nodes
- Two parameters: $k_{training}$ and k_{test}
- Core: all nodes that are incident to at least $k_{training}$ edges in $G[t_0, t'_0]$, and at least k_{test} edges in $G[t_1, t'_1]$

- Networks evolve over time
- Predict only edges whose endpoints appear in both the training and test intervals

- Predict new edges between the nodes in Core

Problem Formulation (2/2)

	training period			Core		
	authors	papers	collaborations[1]	authors	$\|E_{old}\|$	$\|E_{new}\|$
astro-ph	5343	5816	41852	1561	6178	5751
cond-mat	5469	6700	19881	1253	1899	1150
gr-qc	2122	3287	5724	486	519	400
hep-ph	5414	10254	47806	1790	6654	3294
hep-th	5241	9498	15842	1438	2311	1576

Example: Predict links in an evolving collaboration network

- t_0 = 1994, t'_0 = 1996: training interval -> [1994, 1996]
- t_1 = 1997, t'_1 = 1999: test interval -> [1997, 1999]

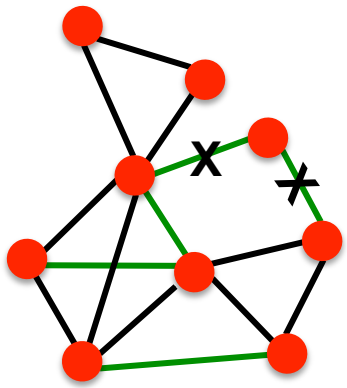
- $G_{collab} = (V, E_{old})$ = G[1994, 1996]
- E_{new}: authors in V that co-author a paper during the test interval but not during the training interval

$k_{training} = 3$, $k_{test} = 3$: Core consists of all authors who have written at least 3 papers during the training period and at least 3 papers during the test period

Goal: predict E_{new}

Link Prediction - Evaluation

- For each pair of nodes (x, y), compute score c(x, y)
  - For example, c(x,y) could be the # of common neighbors of x and y
- Sort pairs (x, y) by the decreasing score c(x, y)
  - Only consider/predict edges where both endpoints are in the core
- Predict the top n pairs of new links
- See which of those links actually appear in $G[t_1, t'_1]$



Methods of Link Prediction

How to assign score c(x, y) between two nodes x and y?

Some form of similarity or node proximity of x and y

- Many different ways to formalize this
- Use only graph features

Methods

- Neighborhood-based (# of shared neighbors)
- Network proximity-based methods (paths between x and y)

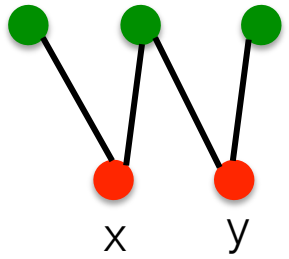
Neighborhood-based Methods (1/4)

- Let $\Gamma(x)$ be the set of neighbors of x in G_{old}
- Methods
  - Common neighbors overlap
  - Jaccard coefficient
  - Adamic/Adar index
  - Preferential attachment

Intuition: The larger the overlap of the neighbors of two nodes, the more likely the nodes to be linked in the future

Neighborhood-based Methods (2/4)

Let $\Gamma(x)$ be the set of neighbors of x in G_{old}



Common neighbors:

$$c(x, y) = |\Gamma(x) \cap \Gamma(y)|$$

A : adjacency matrix

$A_{x,y}^2$: Number of different paths of length 2

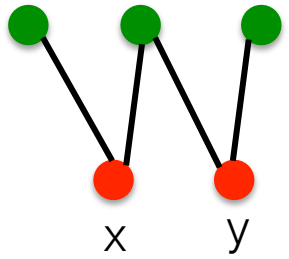
Jaccard coefficient:

$$c(x, y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}$$

The probability that both x and y have common neighbors

Neighborhood-based Methods (3/4)

Let Γ(x) be the set of neighbors of x in G_{old}



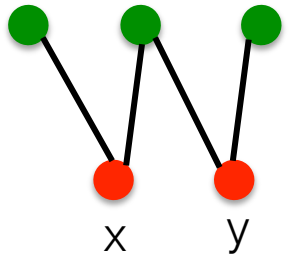
Adamic/Adar:

$$c(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log |\Gamma(z)|}$$

Assigns large weights to common neighbors z of x and y which themselves have few neighbors (weight rare features more heavily)

Neighborhood-based Methods (4/4)

Let $\Gamma(x)$ be the set of neighbors of x in G_{old}



Preferential attachment:

$$c(x, y) = |\Gamma(x)| \cdot |\Gamma(y)|$$

# of neighbors = degree

Based on the intuition that the probability that a new edge has node x as its endpoint is proportional to $|\Gamma(x)|$, i.e., nodes prefer to form ties with 'popular' nodes

Proximity-based Methods (1/4)

Intuition: The "closer" two nodes are in the network, the **more likely** is to be linked in the future

- Methods
  - Based on shortest path length between x and y
  - Based on all paths between x and y
    - Katz_β measure
    - Random walk-based
      - Hitting time
      - Commute time
      - Rooted PageRank
      - SimRank

Shortest Path-Based

For nodes x, y

c(x, y) = (negated) length of shortest path between x and y

Some further normalization may needed

If there are more than n pairs of nodes at the shortest path length ℓ, order them at random

Ensemble of All Paths (1/2)

Katz$_\beta$ measure:

$$c(x,y) = \sum_{\ell=1}^{\infty} \beta^{\ell} \cdot |\text{paths}_{x,y}^{\langle \ell \rangle}|$$

Set of all paths of length ℓ from x to y

- $0 < \beta < 1$ is a parameter of the predictor, exponentially damped to count short paths more heavily
- Small β yields predictions much like common neighbors

Ensemble of All Paths (2/2)

Katz$_\beta$ measure:

$$c(x,y) = \sum_{\ell=1}^{\infty} \beta^{\ell} \cdot |\text{paths}_{x,y}^{\langle \ell \rangle}|$$

Set of all paths of length ℓ from x to y

$$\sum_{\ell=1}^{\infty} \beta^{\ell} \cdot |\text{paths}_{x,y}^{\langle \ell \rangle}| = \beta A_{xy} + \beta^2 (A^2)_{xy} + \beta^3 (A^3)_{xy} + \cdots$$

Closed form: $(I - \beta A)^{-1} - I$ Matrix of scores

- Unweighted version: $\text{path}_{x,y}^{(1)} = 1$ (edge with 'weight = 1')
- Weighted version: $\text{path}_{x,y}^{(1)}$ = # times x and y have collaborated

Random Walk-based Methods (1/2)

Consider a random walk on G_{old} that starts at x and iteratively moves to a neighbor of x chosen uniformly at random from $\Gamma(x)$

- Hitting time $H_{x,y}$ (from x to y): the expected number of steps it takes for the random walk starting at x to reach y

$$c(x, y) = -H_{x,y}$$

Not symmetric, in general

- Commute time $Comm(x, y)$ (from x to y): the expected number of steps to travel from x to y and from y to x

$$c(x, y) = -(H_{x,y} + H_{y,x})$$

Random Walk-based Methods (2/2)

- The hitting time and commute time measures are sensitive to parts of the graph far away from x and y
  - Periodically reset the walk
- Random walk on G_{old} that starts at x and has a probability α of returning to x at each step

- Rooted (Personalized) PageRank
  - Starts from x
  - With probability (1 – α) moves to a random neighbor
  - With probability α returns to x

c(x, y) = stationary probability of y in a rooted PageRank

SimRank

- Intuition: Two objects are similar, if they are related to similar objects

- Two objects x and y are similar, if they are related to objects a and b respectively and a and b are themselves similar

$$\text{sim}(x, y) = \frac{\gamma}{|\Gamma(x)| \cdot |\Gamma(y)|} \sum_{a \in \Gamma(x)} \sum_{b \in \Gamma(y)} \text{sim}(a, b)$$

Recursive definition

computation

$$\text{sim}_0(a, b) = \begin{cases} 1, & \text{if } a = b \\ 0, & \text{if } a \neq b \end{cases}$$

Repeat iteration until convergence

Expresses the average similarity between neighbors of x and neighbors of y

$$c(x, y) = \text{sim}(x, y)$$

[Jeh and Widom '02]

Evaluation and results

How to Evaluate the Prediction?

- Each link predictor p outputs a ranked list L_p of pairs in $V \times V - E_{old}$: predicted new edges in decreasing order of confidence
- Here we focus on Core of the graph (degree $k>3$), thus

$$E^*_{new} = E_{new} \cap (Core \times Core)$$

- Evaluation method: size of intersection of
  - The first n edge predictions from L_p that are in $Core \times Core$ and
  - The actual set E_{new}

How many of the (relevant) top-n predictions are correct (precision?)

Evaluation: Baseline Predictor

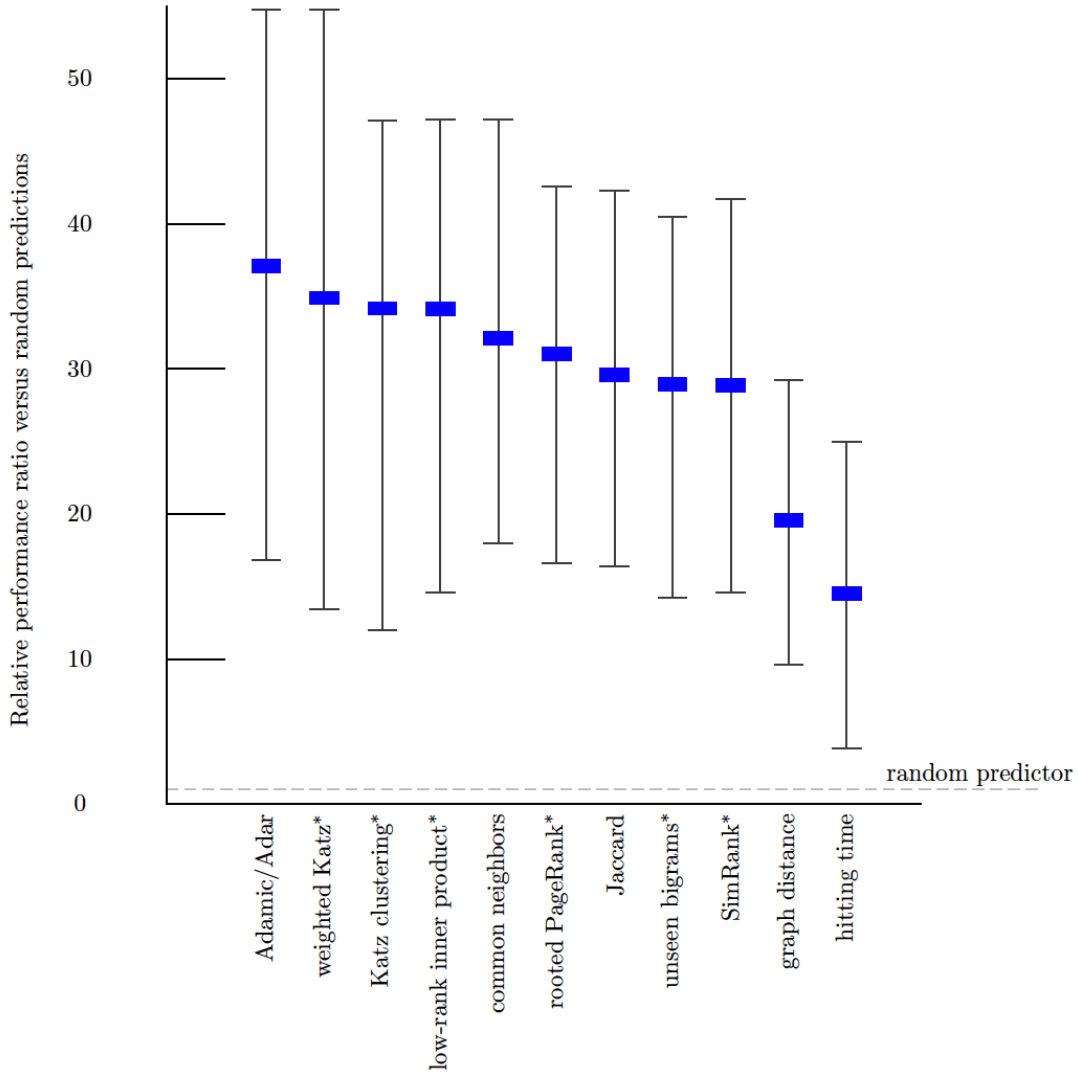
- Prediction accuracy will be measured in terms of relative improvement over a random predictor
  - Why? First application was in co-authorship networks
  - Many collaborations form (or fail to form) for reasons outside the scope of the networks
  - The raw accuracy (in the unsupervised case) is very low

- Random Predictor: randomly selects pairs of authors who did not collaborate (i.e., there is no edge) in the training interval
  - Probability that a random prediction is correct:

$$\frac{|E_{new}|}{\binom{|\text{Core}|}{2} - |E_{old}|}$$

Random predictions are correct with prob. between 0.15% (cond-mat) to 0.48% (astro-ph)

Relative Average Performance



- Relative avg. performance of various predictors vs. the random predictor
- Avg. ratio over 5 datasets
- The Adamic/Adar and Katz measures perform very well

[Liben-Nowell and Kleinberg '03]

Discussion and Extensions

- Improve performance. Even the best (Katz clustering on gr-qc) is correct on only about 16% of its prediction
- Improve efficiency on very large networks (approximation of distances
- Treat more recent links (e.g., collaborations) as more important
- Additional information (paper titles, author institutions, etc.) latently present in the graph
- Why not using those measures as features in a model that performs the predictions?

Supervised Link Prediction

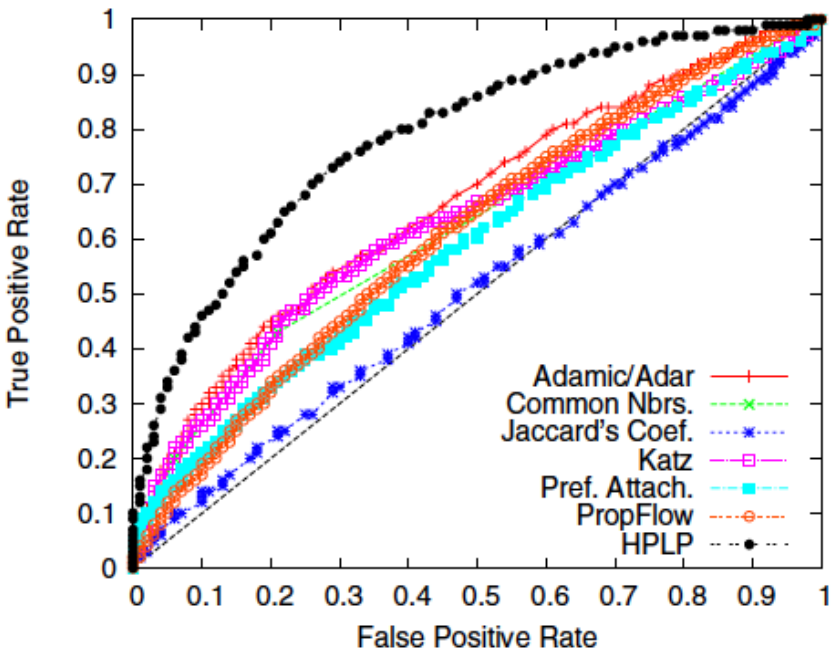
- The same "similarity" measures can be used as features for supervised link prediction
  - Training and test datasets
  - Similar to a binary classification task: 'class 1' (edge exists); 'class 0' (edge doesn't exist)
  - Use ROC curve (false positive rate vs. true positive rate) to evaluate the predictions
  - Weak point: have to deal with class imbalance

Features

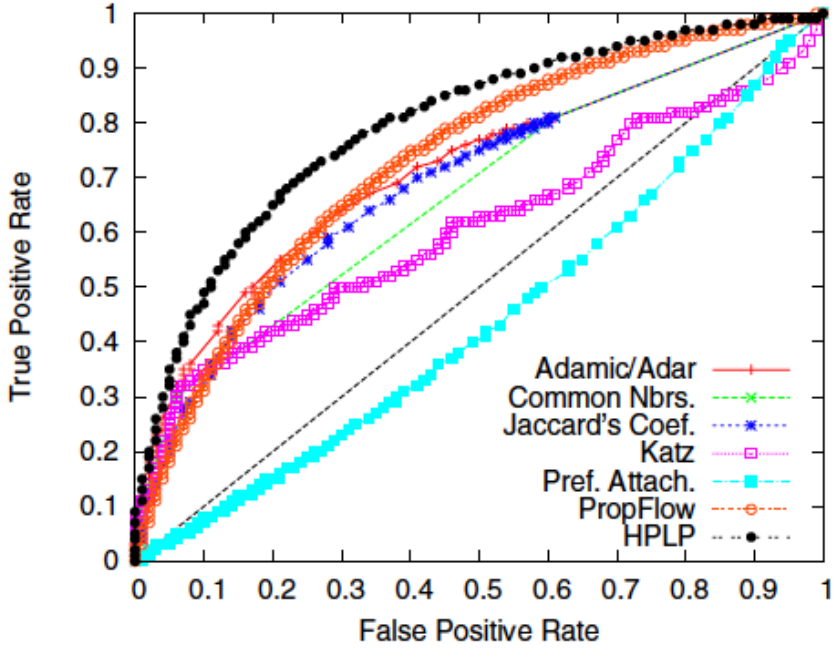
Name	Parameters	HPLP	HPLP+
In-Degree(i)	-	✓	✓
In-Volume(i)	-	✓	✓
In-Degree(j)	-	✓	✓
In-Volume(j)	-	✓	✓
Out-Degree(i)	-	✓	✓
Out-Volume(i)	-	✓	✓
Out-Degree(j)	-	✓	✓
Out-Volume(j)	-	✓	✓
Common Nbrs(i,j)	-	✓	✓
Max. Flow(i,j)	$l = 5$	✓	✓
Shortest Paths(i,j)	$l = 5$	✓	✓
PropFlow(i,j)	$l = 5$	✓	✓
Adamic/Adar(i,j)	-		✓
Jaccard's Coef(i,j)	-		✓
Katz(i,j)	$l = 5$, $\beta = 0.005$		✓
Pref Attach(i,j)	-		✓

[Lichtenwalter et al. '10]

Experimental Results



Cond-Mat



Phone

- Random forests classifier
- HPLP: method that combines multiple features

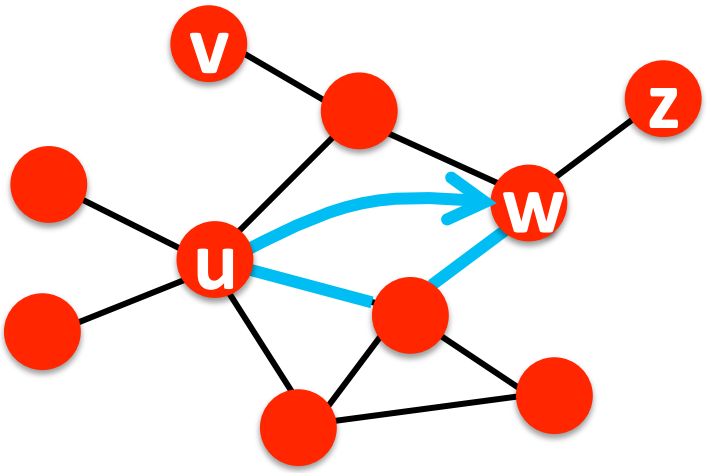
[Lichtenwalter et al. '10]

LPmade: Link Prediction Made Easy
Source: https://www3.nd.edu/~dial/software/

Supervised random walks for link prediction

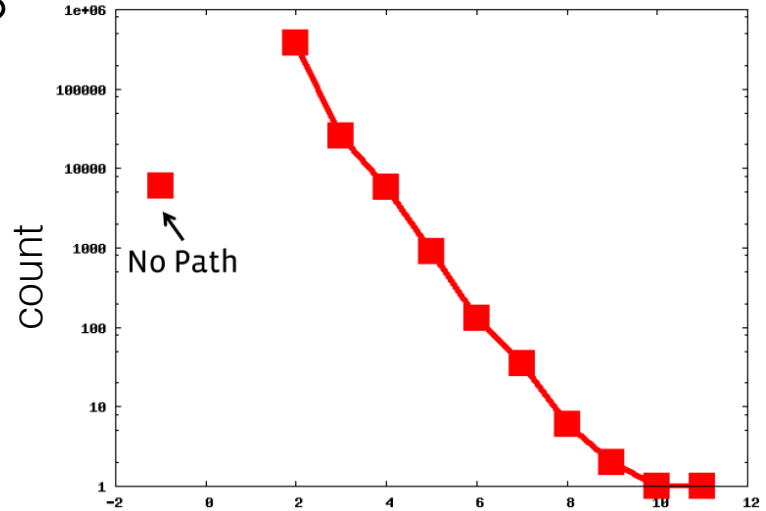
Supervised Link Prediction - Motivation

- Can we learn to predict new friends?
  - Facebook's People You May Know
  - Let's look at the FB data
    - 92% of new friendships on FB are friend-of-a-friend
    - More mutual friends help

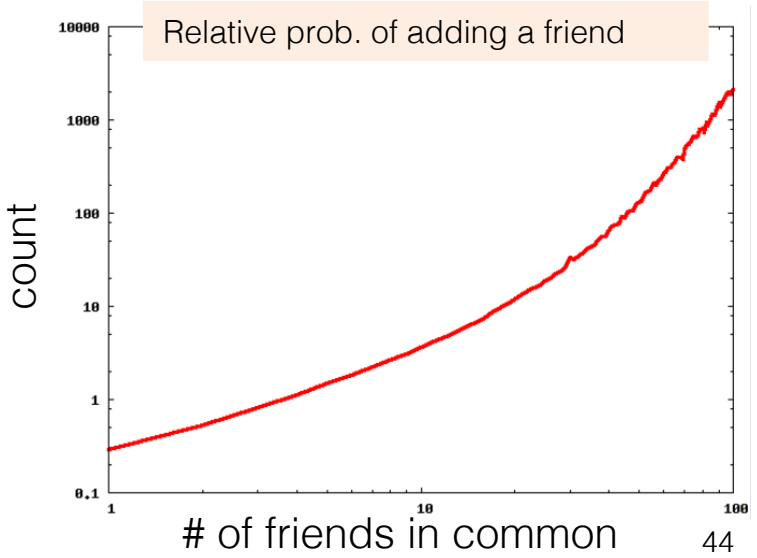


[Backstrom and Leskovec '11]

Friendship by # of hops



Relative prob. of adding a friend

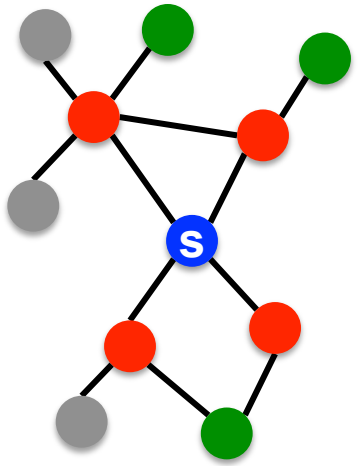


Supervised Link Prediction - Overview

- Proximity measure + classification approach [previous part]
  - Class imbalance
  - Which proximity measures to use
- Rank the nodes: higher score to nodes that s will create link to
  - PageRank, Random walk with restarts, etc.
  - Node/edge features are not taken into account
- In this part: use the idea of Supervised Random Walks
  - Combine information about the network structure and node/edge attributes (e.g., age, gender, home town, school – profile information)
  - Use this information to guide the random walk
  - Learn a function that assigns strengths (weights) to the edges …
    - i.e., random walk transition probabilities
  - … such that a random walker is more likely to visit the nodes to which new links will be created in the furture

Supervised Link Prediction (1/2)

- Goal: Recommend a list of possible friends
- Supervised ML setting
  - Labeled training examples:
    - For every user s, we have a list of other users she will create links to in the future: $\{d_1, \ldots, d_k\}$
    - Use FB data from May 2012 and $\{d_1, \ldots, d_k\}$ are the new friendships you created since then
    - These are the positive training examples
    - Use all other users as negative examples
  - Task: For a given node s, score nodes $\{d_1, \ldots, d_k\}$ higher than any other nodes in the network

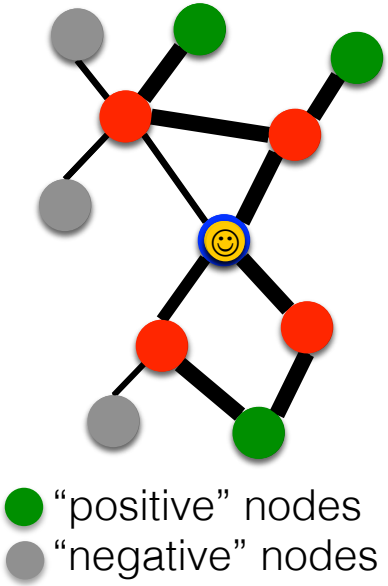


Green nodes are the nodes to which s creates links in the future

[Backstrom and Leskovec '11]

Supervised Link Prediction (2/2)

- How to combine node/edge features and the network structure?
  - Estimate strength of each friendship (u, v) using:
    - Profile of user u, profile of user v
    - Interaction history of users u and v
  - This creates a weighted graph
  - Do Personalized PageRank from s and measure the "proximity" (the visiting prob.) of any other node w from s
  - Sort nodes w by decreasing proximity



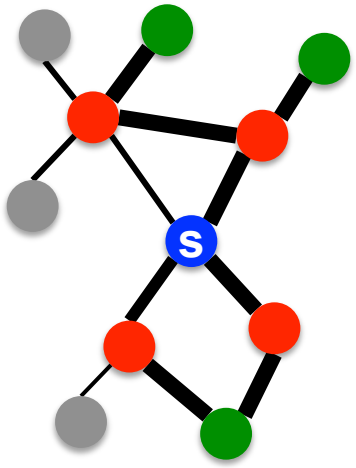
Supervised Random Walks

- Let s be the starting node
- Let $f_\beta(u, v)$ be a function that assigns strength α_{uv} to edge (u ,v)

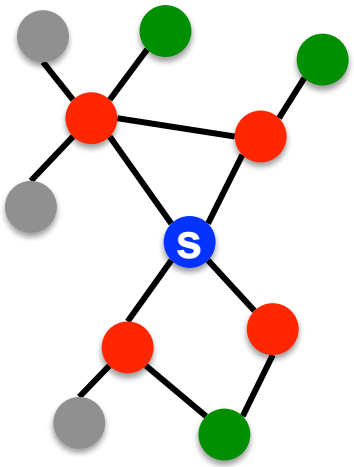
$$\alpha_{uv} = f_\beta(u, v) = \exp\left(-\sum_i \beta_i \cdot x_{uv}(i)\right)$$

  - x_{uv} is feature vector of (u, v)
    - Features of node u
    - Features of node v
    - Features of edge (u, v)
  - *[Note: β is the weight vector that we will estimate later]*

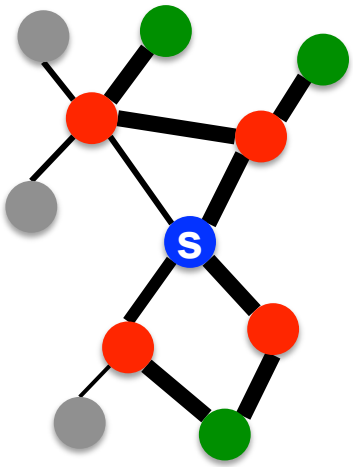
- Do random walk with restarts from s where the transitions are made based on the edge strengths α_{uv}



SRW: Prediction



Network



Set edge strengths
$\alpha_{uv} = f_\beta(u, v)$

Random Walk with Restarts on the weighted graph. Each node **w** has a PageRank proximity p_w

Sort nodes **w** by the decreasing PageRank score p_w

Recommend top **k** nodes with the highest proximity p_w to node **s**

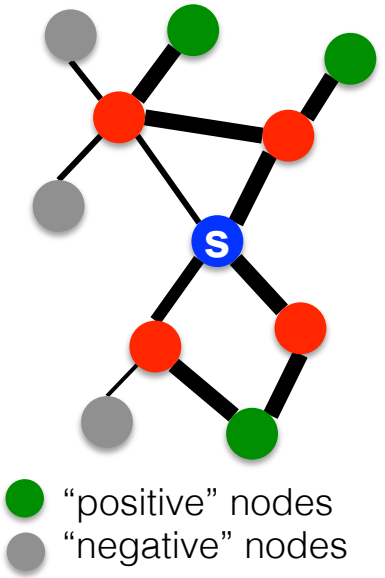
- How do we estimate edge strengths?
  - How to estimate parameters β in $f_\beta(u, v)$
- Idea: Set β such that it (correctly) predicts the known future links

Personalized PageRank

- α_{uv} is the strength of edge (u, v)
- Random walk transition matrix

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

Not a uniform probability for each outgoing edge



- PageRank transition matrix

$$Q_{ij} = (1-\gamma)Q'_{ij} + \gamma \mathbf{1}(j = s)$$

γ: restart probability of PageRank

- Compute PageRank vector $\mathbf{p} = \mathbf{p}^T Q$
- Rank nodes w by decreasing p_w

The Optimization Problem

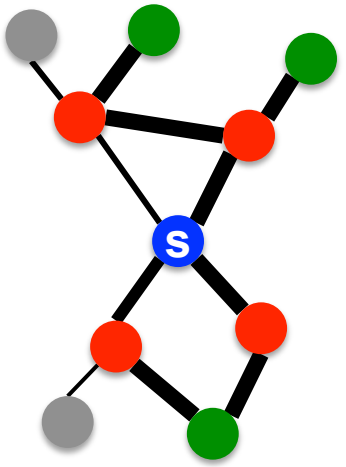
- Positive examples $D = \{d_1, \ldots, d_k\}$
- Negative examples $L = \{\text{other nodes}\}$
- What do we want?

$$\min_\beta F(\beta) = ||\beta||^2$$

Prefer small weights β to prevent overfitting

$$\text{such that: } \forall d \in D, l \in L : p_l < p_d$$

Every positive example (list D) has to have higher PageRank score than every negative example (list L)



– Note:
  - Exact solution to this problem may not exist
  - So we make the constraints "soft" (i.e., optional) - introduce a loss function

[Backstrom and Leskovec '11]

Making Constraints "Soft"

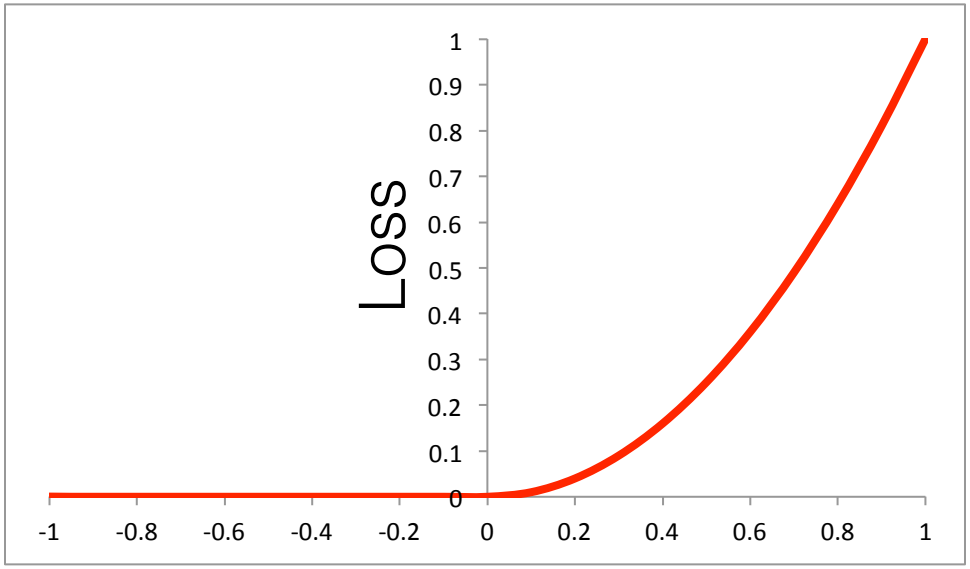
- Want to minimize

$$\min_{\beta} F(\beta) = \sum_{d \in D, l \in L} h(p_l - p_d) + \lambda ||\beta||^2$$

Regularization parameter

Penalty for violating the constraint that $p_d > p_l$

  - Loss function: h(x) = 0 if x < 0, else $h(x) = x^2$



$p_l < p_d$ $p_l = p_d$ $p_l > p_d$

Solving the Problem: Intuition

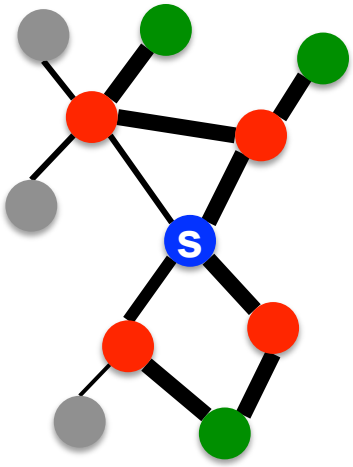
- How to minimize F ?

$$\min_\beta F(\beta) = \sum_{d \in D, l \in L} h(p_l - p_d) + \lambda ||\beta||^2$$

- Both p_l and p_d depend on β
  - Given β assign edge weights $\alpha_{uv} = f_\beta(u, v)$
  - Using $Q=[\alpha_{uv}]$ (edge strengths), compute PageRank scores p
  - Rank nodes by decreasing score

- Goal: want to find β such that $p_l < p_d$

Solving the Problem: Intuition



- How to minimize F(β)?

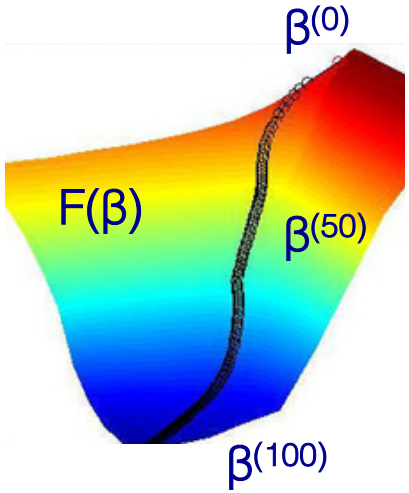
$$\min_\beta F(\beta) = \sum_{d \in D, l \in L} h(p_l - p_d) + \lambda \|\beta\|^2$$

- Idea
  - Start with some random $\beta^{(0)}$
  - Evaluate the derivative of F(β) and do a small step in the opposite direction

$$\beta^{(t+1)} = \beta^{(t)} - \eta \frac{\partial F(\beta^{(t)})}{\partial \beta}$$

  - Repeat until convergence

gradient descent



Data: Facebook

- Facebook Iceland network
  - 174,000 nodes (55% of population)
  - Avg. degree 168
  - Avg. person added 26 friends/month
- For every node s:
  - Positive examples
    - D = {New friendships created in Nov '09}
  - Negative examples
    - L = {Other nodes s that did not create new links to}
  - Limit to friends of friends (FoFs) – 2-hop neighborhood
    - On avg. there are 20,000 FoFs (maximum is 2 million)

Experimental Setting

- Node and Edge features for learning
  - Node: Age, Gender, Degree
  - Edge: Age of an edge, Communication, Profile visits, Co-tagged photos

- Evaluation
  - Precision at top 20
    - We produce a list of 20 candidates
      - By taking top 20 nodes x with highest PageRank score p_x
    - Measure to what fraction of these nodes s actually links to

[Backstrom and Leskovec '11]

Experimental Results – Facebook (1/2)

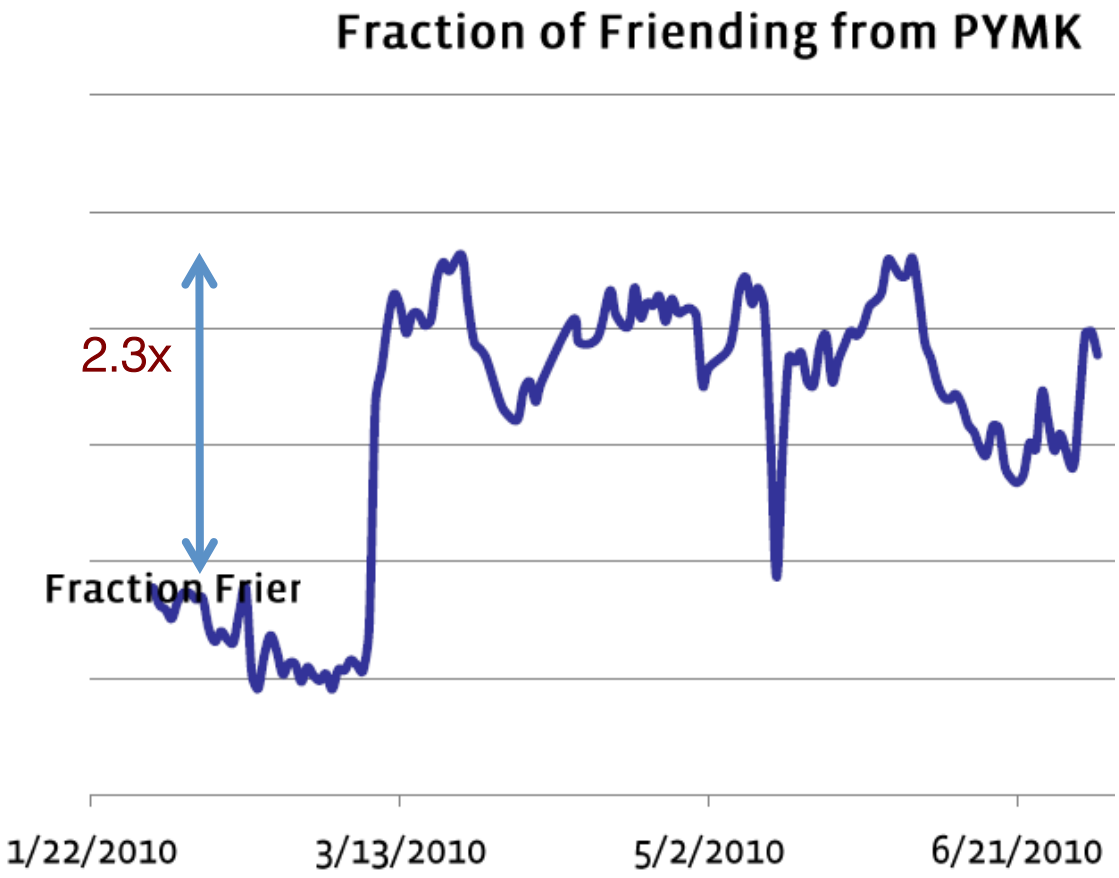
- Facebook: Predict future friends
  - Adamic/Adar index already works great
  - Supervised Random Walks (SRW) gives slight improvement

Learning Method	Prec@Top20
Random Walk with Restart	6.80
Adamic-Adar	7.35
Common Friends	7.35
Degree	3.25
SRW: one edge type	6.87
SRW: multiple edge types	**7.57**

Recommend 20 potential friendships in FB

Experimental Results – Facebook (2/2)

- 2.3x improvement over previous FB-PYMK (People You May Know)



Experimental Results - Co-Authorship

- Arxiv Hep-Ph collaboration network
  - Poor performance of unsupervised methods
  - SRW gives a boost of 25%

Learning Method	Prec@Top20
Random Walk with Restart	3.41
Adamic-Adar	3.13
Common Friends	3.11
Degree	3.05
SRW: one edge type	4.24
SRW: multiple edge types	**4.25**

Link Prediction - Summary

- Node proximity measures
  - Unsupervised and supervised
- Supervised random walks
  - Experiments on Facebook data
- WTF: The “who to follow” service at Twitter
  - Combination of random walks with the SALSA algorithm (a variant of HITS)

Thank You!

DiscoverGreece.com