

Link Prediction

Davide Mottin, Konstantina Lazaridou

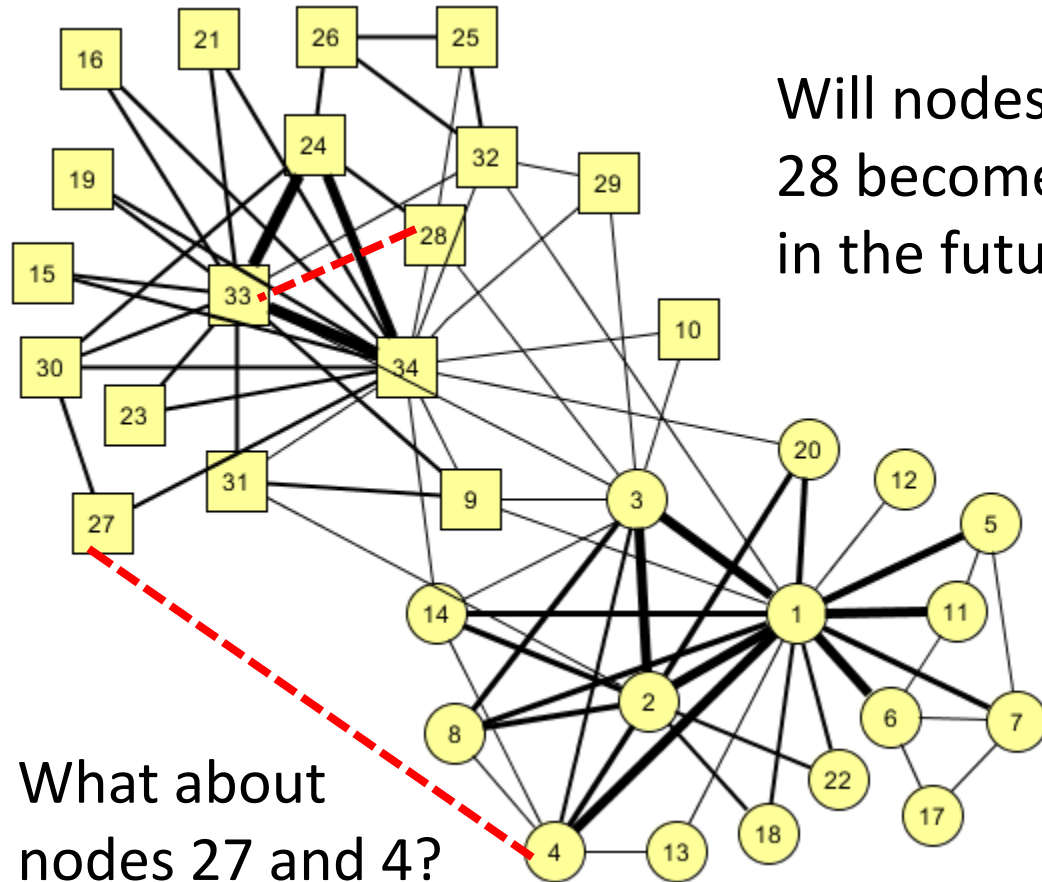
Hasso Plattner Institute

Graph Mining course Winter Semester 2016

Acknowledgements

- Most of this lecture is taken from:
<http://www.cs.uoi.gr/~tsap/teaching/cs-l14/>
- Other adapted content is from:
 - Lu, L. and Zhou, T., 2011. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, 390(6), pp.1150-1170.
 - The link prediction problem for social networks, Alexandra Chouldechova:
<http://statweb.stanford.edu/~owen/courses/319/achouldechova.pdf>

Link Prediction



Will nodes 33 and 28 become friends in the future?

Does the network structure contain enough information to predict which new links will be formed in the future?

What about nodes 27 and 4?

Why link prediction?

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

Who to follow

The image shows a screenshot of the Twitter interface. At the top, there is a navigation bar with 'Home', 'Connect', and 'Discover' tabs. A search bar is visible with the placeholder text 'Enter a #hashtag or keyword'. Below the navigation bar, there is a sidebar on the left with several menu items: 'Stories', 'Activity', 'Who to follow', 'Find friends', and 'Browse categories'. The 'Who to follow' section is highlighted, showing a list of suggested accounts. The main content area displays the 'Who to follow' header, a search bar, and three suggested accounts: David Allen (@gtdguy), Eric Perkins (@PerkatPlay), and Kevin D. Lyons (@KevinLyons). Each account entry includes a profile picture, name, handle, bio, and a 'Follow' button.

Home Connect Discover

Enter a #hashtag or keyword

Stories

Activity

Who to follow

Find friends

Browse categories

Minneapolis trends · Change

#FlyMeToLondon Promoted

#EveryoneHasThat1Friend

#MyMomWouldBeatMyAssIf


#ThoughtsWhileRunning

Who to follow

Twitter accounts suggested for you based on who you follow and more.

Search using a person's full name or @username

Search Twitter

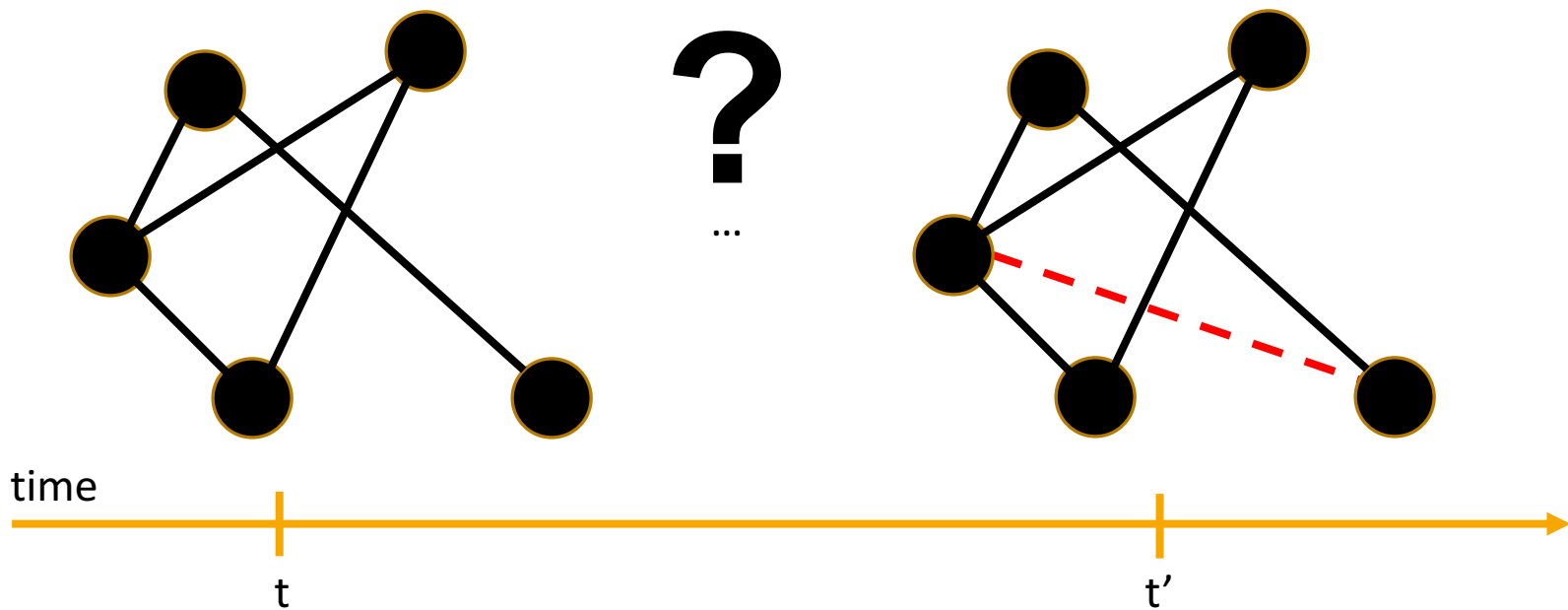
 **David Allen**  @gtdguy
Originator of GTD, founder of David Allen Co.
Followed by seth goldstein , Les McKeown and Kellie Sites .

 **Eric Perkins** @PerkatPlay
KARE-TV Host/News/Sports Anchor/Reporter
Followed by Alecia Puppe , Adam Proehl and Melissa Harrison .

 **Kevin D. Lyons** @KevinLyons

Understanding the network

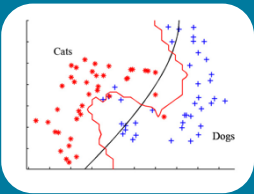
- Understanding how social networks **evolve**
- **The link prediction problem**
 - Given a snapshot of a social network at time t , we seek to accurately predict the edges that will be added to the network during the interval (t, t')



Lecture road



Unsupervised methods



Classification approaches



Who to follow

Link prediction problem

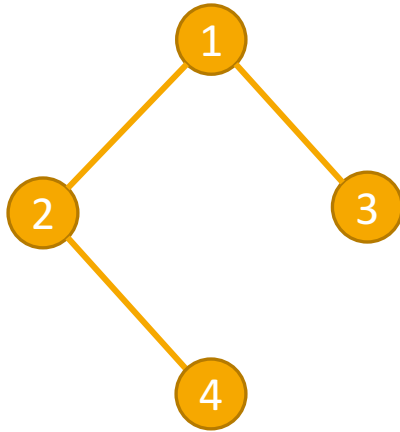
For $t < t_0$, let $G[t, t_0]$ denote the subgraph of G consisting of all edges that took place between t and t_0 . For $t_0 < t'_0 < t_1 < t'_1$, given $G[t_0, t'_0]$, we wish to output a list of edges not in $G[t_0, t'_0]$ that are predicted to appear in $G[t_1, t'_1]$

- ✓ $[t_0, t'_0]$ training interval
- ✓ $[t_1, t'_1]$ test interval

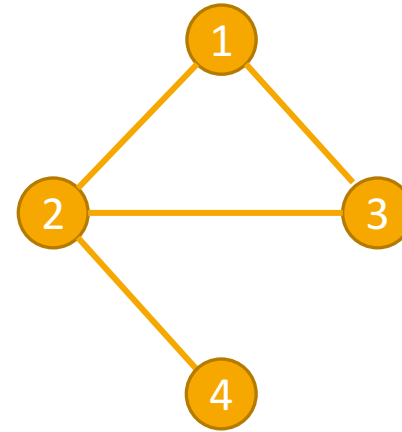
- Based solely on the *topology* of the network (social proximity) (the more general problem also considers attributes of the nodes and links)
- Different from the problem of *inferring missing* (hidden) links (there is a temporal aspect)

Link Prediction concepts

$$t_0 < t'_0 < t_1 < t'_1$$



$$G[t_0, t'_0]$$
$$E_{old} = \{(1,2), (1,3), (2,4)\}$$



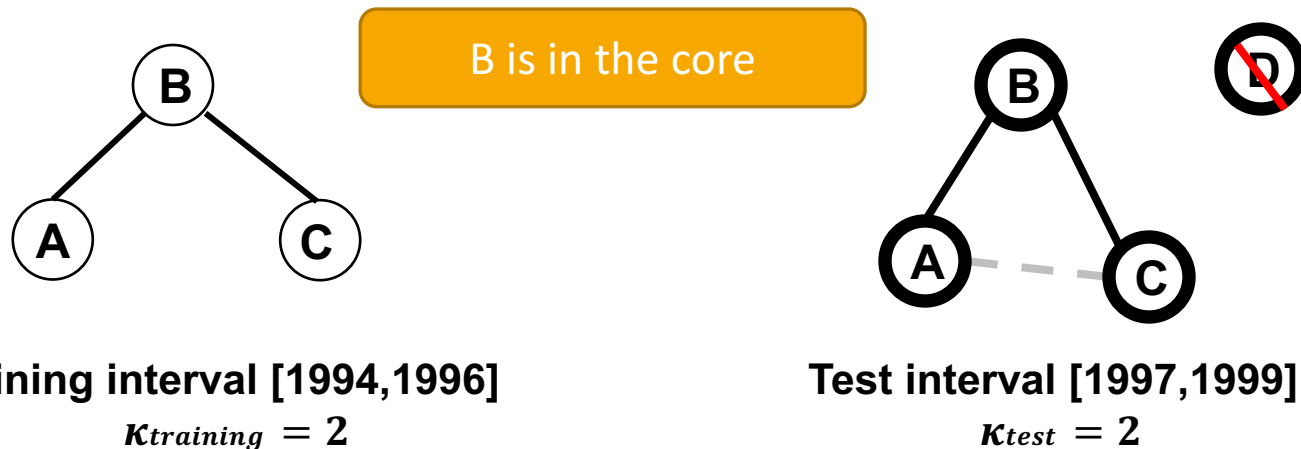
$$G[t_1, t'_1]$$
$$E_{new} = \{(2,3)\}$$

Definition [Core]

$Core \subset V$ is the set of all nodes that are incident to at least $\kappa_{training}$ edges in $G[t_0, t'_0]$ and at least κ_{test} edges in $G[t_1, t'_1]$

An example for link prediction

- Co-authorship network (G) from “author list” of the physics e-Print arXiv (www.arxiv.org)
- Took 5 such networks from 5 sections of the print



Core: set of authors who have at least 2 papers during both training and test

$$G[1994,1996] = G_{collab} = \langle A, E_{old} \rangle$$

$$E_{new} = \text{new collaborations (edges)}$$

Data

	training period			Core		
	authors	papers	edges	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	17806	1790	6654	3294
hep-th	5241	9498	15842	1438	2311	1576

Figure 1: The five sections of the arXiv from which co-authorship networks were constructed: **astro-ph** (astrophysics), **cond-mat** (condensed matter), **gr-qc** (general relativity and quantum cosmology), **hep-ph** (high energy physics—phenomenology), and **hep-th** (high energy physics—theory). The set **Core** is the subset of the authors who have written at least $\kappa_{training} = 3$ papers during the training period and $\kappa_{test} = 3$ papers during the test period. The sets E_{old} and E_{new} denote edges between **Core** authors which first appear during the training and test periods, respectively.

Example Dataset: co-authorship

	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

$t_0 = 1994, t'_0 = 1996$: **training interval** -> [1994, 1996]

$t_1 = 1997, t'_1 = 1999$: **test interval** -> [1997, 1999]

- $G_{collab} = \langle V, E_{old} \rangle = G[1994, 1996]$
- E_{new} : authors in V that co-author a paper during the test interval **but not** during the training interval
- $\kappa_{training} = 3, \kappa_{test} = 3$: **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

Predict E_{new}

Methods for link prediction

- Assign a connection weight score(x, y) to each pair of nodes (x, y) based on the input graph
- Produce a ranked list of decreasing order of score
- We can consider all links *incident to a specific node x* , and recommend to x the top ones
- If we focus to a specific x , the score can be seen as a **centrality measure** for x

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

✓ Some form of **similarity** or **node proximity**

Lecture road



Unsupervised methods



Classification approaches



Who to follow

Summary of unsupervised methods

- Neighborhood based approaches
 - Common neighbors, Adamic, Jaccard, ...
- Path based approaches
 - Shortest path, Katz
- Low-rank approximation
- Clustering and mixed approaches

LP Methods: Neighborhood-based

Intuition

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

Let $\Gamma(x)$ denote the set of nodes adjacent to x , i.e, $\Gamma(x) = \{y | (x, y) \in E\}$

- **Common neighbors:** how many neighbors are in common between x and y

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

- **Jaccard coefficient:** how likely a neighbor of x is also a neighbor of y

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

- **Adamic/Adar:** large weight to common neighbors with low degree (the lower the degree the higher the relevance)

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

Adamic

- Neighbors who are linked with **2** nodes are assigned weight = $1/\log(2) = \mathbf{1.4}$
- Neighbors who are linked with **5** nodes are assigned weight = $1/\log(5) = \mathbf{0.62}$

LP Methods: Preferential attachment

Intuition

The more popular a node is the more probable it will form a link with popular nodes

- Let $\Gamma(x)$ denote the set of nodes adjacent to x , i.e, $\Gamma(x) = \{y | (x, y) \in E\}$
$$score(x, y) = |\Gamma(x)| |\Gamma(y)|$$
- Inspired to scale-free network formation
- Researchers found empirical evidence to suggest that co-authorship is correlated with the product of the neighborhood sizes

This depends *on the degrees* of the nodes not on their neighbors per se

Other neighborhood based methods

- Salton index: $score(x, y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{\sqrt{|\Gamma(x)| |\Gamma(y)|}}$
- Sørensen index: $score(x, y) = \frac{2|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x)| + |\Gamma(y)|}$
- Hub Promoted Index: $score(x, y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{\min\{|\Gamma(x)|, |\Gamma(y)|\}}$
- Hub Depressed Index: $score(x, y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{\max\{|\Gamma(x)|, |\Gamma(y)|\}}$
- Leicht-Holme-Newman Index: $score(x, y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x)| |\Gamma(y)|}$
- Resource allocation: $score(x, y) = \sum_{z \in |\Gamma(x) \cap \Gamma(y)|} \frac{1}{|\Gamma(z)|}$

Methods for Link Prediction: Path based

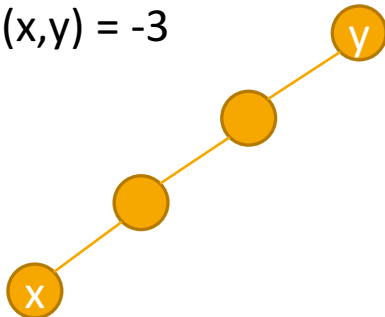
Intuition

Use the (shortest) distance between two nodes as a link prediction measure

- For $(x, y) \in V \times V - E_{old}$

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

$score(x, y) = -3$



Very basic approach, it does not consider connections among (x, y) but only the distance

LP Methods: Path based

- Katz index

$$score(x, y) = \sum_{\ell=1}^{\infty} \beta^{\ell} |paths_{xy}^{(\ell)}| = \beta A_{xy} + \beta^2 A_{xy}^2 + \dots$$

Element (x,y) in the
Adjacency matrix

- Sum over ALL paths of length ℓ
- $0 < \beta < 1$ is a parameter of the predictor, exponentially damped to count short paths more heavily
- *Small β = predictions much like common neighbors*
- Two forms:
 - **Unweighted**: 1 if two authors collaborated, 0 otherwise
 - **Weighted**: strength of the collaboration

Closed form for the entire score matrix:

$$(I - \beta A)^{-1} - I$$

LP Methods: Path based

- Consider a random walk on G_{old} that starts at x and iteratively moves to a neighbor of x chosen uniformly random from $\Gamma(x)$
- The **Hitting Time** $H_{x,y}$ from x to y is the expected number of steps it takes for the random walk starting at x to reach y .

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

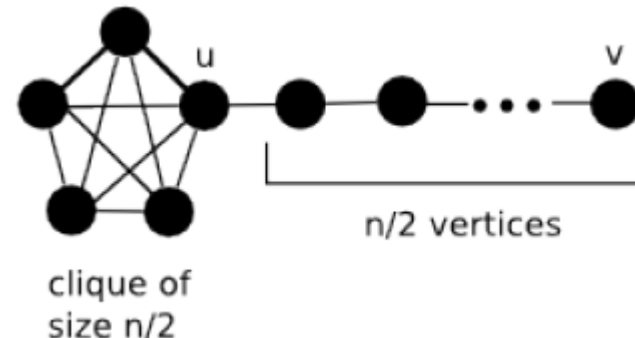
- The **Commute Time** from x to y is the expected number of steps to travel from x to y and from y to x

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

Not symmetric, can be shown

$$h_{vu} = \Theta(n^2)$$

$$h_{uv} = \Theta(n^3)$$



LP Methods: Path based

- The hitting time and commute time measures are sensitive to parts of the graph far away from x and y -> periodically **jump back to x**
- Random walk on G_{old} that starts at x and has a probability c of returning to x at each step
- Random walk with restart: Starts from x , with probability $(1 - c)$ moves to a random neighbor and with probability c returns to x

$$s = (1 - c)(I - cD^{-1}A)^{-1}e_x$$

where s is a similarity vector between x and all the other nodes in the graph and e_x is the vector that has all 0, but a 1 in position x

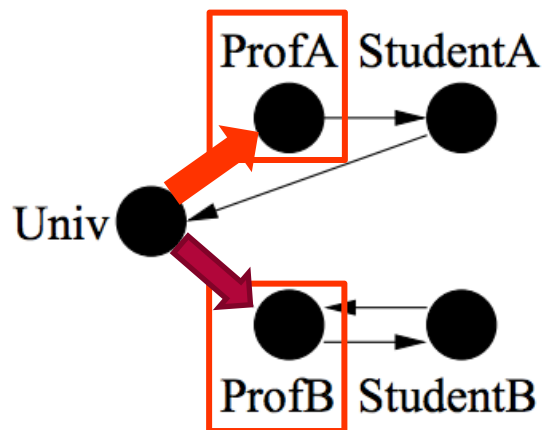
$$\text{score}(x, y) = s_y$$

Path based: SimRank approaches

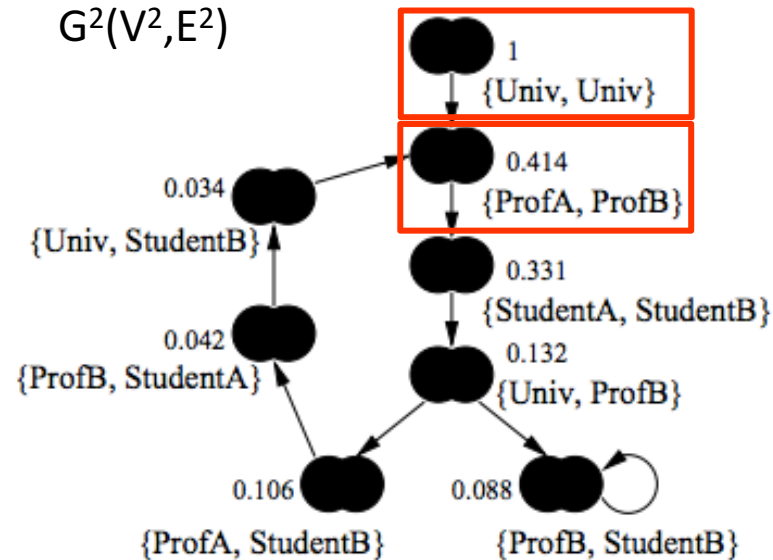
Intuition:

two objects are similar if they are referenced by similar objects

$G(V,E)$



$G^2(V^2,E^2)$

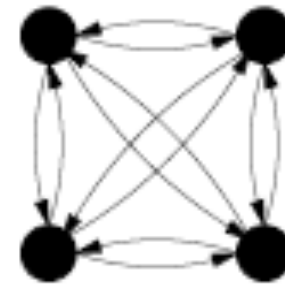
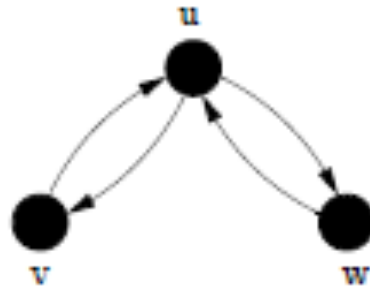
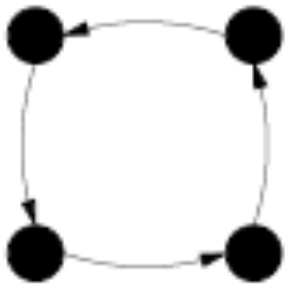


Structural context

Glen Jeh and Jennifer Widom. **SimRank: a measure of structural-context similarity**. SIGKDD, 2002

Path based: SimRank approaches

Expected Meeting Distance (EMD): how soon two random surfers are expected to meet at the same node if they started at nodes x and y and randomly walked (in lock step) the graph backwards



- $score(\cdot, \cdot) = \infty$
=> no node will meet

- $score(u, v) = score(u, w) = \infty$
- $score(v, w) = 1$
=> v and w are much more similar than u is to v or w .

- $score(\cdot, \cdot) = 3$
=> any two node will meet in expectedly 3 steps, the similarity is lower than the previous for v, w

Path based: SimRank approaches

- Let us consider G^2
- A node (a, b) as a state of the tour in G : if a moves to c , b moves to d in G , then (a, b) moves to (c, d) in G^2

A tour in G^2 of length n represents a pair of tours in G where each has length n

- What are the states in G^2 that correspond to “meeting” points?

Singleton nodes (common neighbors)

- The EMD $m(a, b)$ is just the expected distance (hitting time) in G^2 between (a, b) and any singleton node
- The sum is taken over all walks that start from (a, b) and end at a singleton node

LP Methods: Low Rank Approximations

- Assume that a small number of latent factors describe the social and attribute link strength
- Take the adjacency matrix A and a parameter r
- Extract these r latent factors using a low rank matrix approximations
- Apply SVD to find a factorization of A
- Take the r that best approximates A

Singular Value Decomposition

Diagonal matrix

$$A = U \Sigma V^T = \begin{bmatrix} \vec{u}_1 & \vec{u}_2 & \dots & \vec{u}_r \end{bmatrix} \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_r \end{bmatrix} \begin{bmatrix} \vec{v}_1 \\ \vec{v}_2 \\ \vdots \\ \vec{v}_r \end{bmatrix}$$

$[n \times r]$ $[r \times r]$ $[r \times n]$

Orthonormal matrix

Orthonormal matrix

- r : rank of matrix A
- $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r$: singular values (square roots of eig-vals AA^T , $A^T A$)
- u_1, u_2, \dots, u_r : left singular vectors (eig-vectors of AA^T)
- v_1, v_2, \dots, v_r : right singular vectors (eig-vectors of $A^T A$)

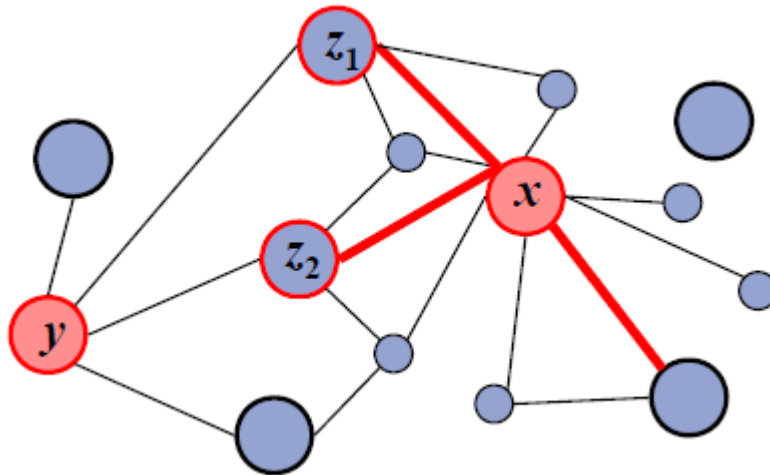
$$A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \dots + \sigma_r u_r v_r^T$$

LP Methods: Unseen bigrams

Intuition

To compute the $\text{score}(x, y)$ use top-k nodes S_x^k that are similar to x using any of the previous scores and intersect the neighbors of y

- $\text{score}_{unweighted}^*(x, y) = |\{z: z \in \Gamma(y) \cap S_x^k\}|$
- $\text{score}_{weighted}^*(x, y) = \sum_{z \in \Gamma(y) \cap S_x^k} \text{score}(x, z)$



z_1, z_2 are similar to x and in the neighborhood of y

LP Methods: Clustering

Intuition

Improve the score deleting the "weakest" edges

- Compute $score(x, y)$ for all edges in E_{old}
- Given a user defined parameter p delete $(1 - p)$ fraction of the edges whose score is the lowest
- Recompute $score(x, y)$ for all pairs in the subgraph

Evaluation of Link Prediction

- Each link predictor p outputs a ranked list L_p of pairs in $V \times V \setminus E_{old}$: predicted new collaborations in decreasing order of confidence

- If you have defined a core then consider

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

Evaluation method: *Size of the intersection of*

- the first n edge predictions from L_p that are in $Core \times Core$, and
- the set E_{new}^*

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

Evaluation of LP: baseline

- Baseline: **random predictor**
- Randomly select pairs of nodes who are not connected in the training interval
- Probability that a random prediction is correct

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

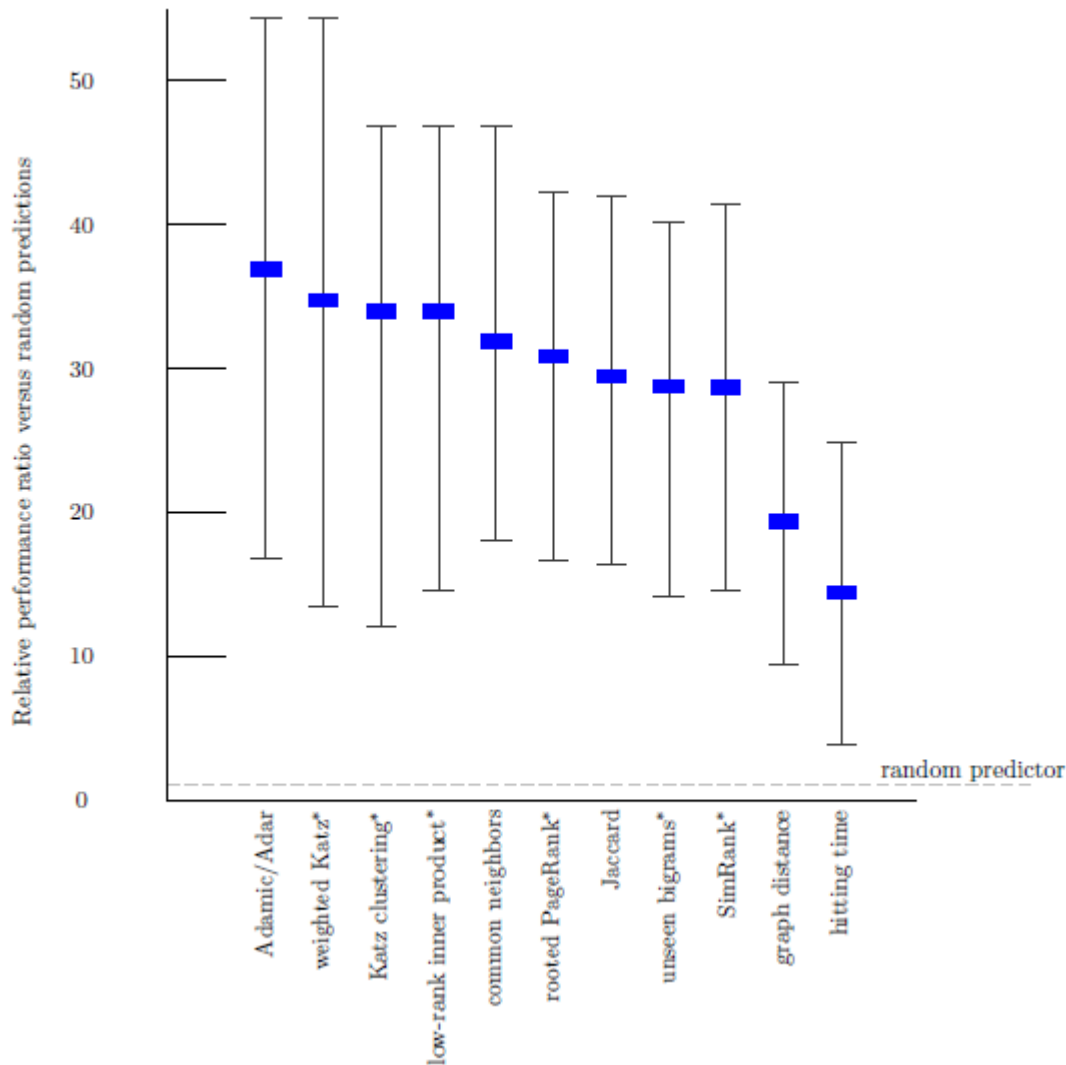
Evaluation: improvement over random

predictor		astro-ph	cond-mat	gr-qc	hep-ph	hep-th
probability that a random prediction is correct		0.475%	0.147%	0.341%	0.207%	0.153%
graph distance (all distance-two pairs)		9.4	25.1	21.3	12.0	29.0
common neighbors		18.0	40.8	27.1	26.9	46.9
preferential attachment		4.7	6.0	7.5	15.2	7.4
Adamic/Adar		16.8	54.4	30.1	33.2	50.2
Jaccard		16.4	42.0	19.8	27.6	41.5
SimRank	$\gamma = 0.8$	14.5	39.0	22.7	26.0	41.5
hitting time		6.4	23.7	24.9	3.8	13.3
hitting time—normed by stationary distribution		5.3	23.7	11.0	11.3	21.2
commute time		5.2	15.4	33.0	17.0	23.2
commute time—normed by stationary distribution		5.3	16.0	11.0	11.3	16.2
rooted PageRank	$\alpha = 0.01$	10.8	27.8	33.0	18.7	29.1
	$\alpha = 0.05$	13.8	39.6	35.2	24.5	41.1
	$\alpha = 0.15$	16.6	40.8	27.1	27.5	42.3
	$\alpha = 0.30$	17.1	42.0	24.9	29.8	46.5
	$\alpha = 0.50$	16.8	40.8	24.2	30.6	46.5
Katz (weighted)	$\beta = 0.05$	3.0	21.3	19.8	2.4	12.9
	$\beta = 0.005$	13.4	54.4	30.1	24.0	51.9
	$\beta = 0.0005$	14.5	53.8	30.1	32.5	51.5
Katz (unweighted)	$\beta = 0.05$	10.9	41.4	37.4	18.7	47.7
	$\beta = 0.005$	16.8	41.4	37.4	24.1	49.4
	$\beta = 0.0005$	16.7	41.4	37.4	24.8	49.4

Evaluation: improvement over random

predictor		astro-ph	cond-mat	gr-qc	hep-ph	hep-th
probability that a random prediction is correct		0.475%	0.147%	0.341%	0.207%	0.153%
graph distance (all distance-two pairs)		9.4	25.1	21.3	12.0	29.0
common neighbors		18.0	40.8	27.1	26.9	46.9
Low-rank approximation:	rank = 1024	15.2	53.8	29.3	34.8	49.8
Inner product	rank = 256	14.6	46.7	29.3	32.3	46.9
	rank = 64	13.0	44.4	27.1	30.7	47.3
	rank = 16	10.0	21.3	31.5	27.8	35.3
	rank = 4	8.8	15.4	42.5	19.5	22.8
	rank = 1	6.9	5.9	44.7	17.6	14.5
Low-rank approximation:	rank = 1024	8.2	16.6	6.6	18.5	21.6
Matrix entry	rank = 256	15.4	36.1	8.1	26.2	37.4
	rank = 64	13.7	46.1	16.9	28.1	40.7
	rank = 16	9.1	21.3	26.4	23.1	34.0
	rank = 4	8.8	15.4	39.6	20.0	22.4
	rank = 1	6.9	5.9	44.7	17.6	14.5
Low-rank approximation:	rank = 1024	11.4	27.2	30.1	27.0	32.0
Katz ($\beta = 0.005$)	rank = 256	15.4	42.0	11.0	34.2	38.6
	rank = 64	13.1	45.0	19.1	32.2	41.1
	rank = 16	9.2	21.3	27.1	24.8	34.9
	rank = 4	7.0	15.4	41.1	19.7	22.8
	rank = 1	0.4	5.9	44.7	17.6	14.5
unseen bigrams (weighted)	common neighbors, $\delta = 8$	13.5	36.7	30.1	15.6	46.9
	common neighbors, $\delta = 16$	13.4	39.6	38.9	18.5	48.6
	Katz ($\beta = 0.005$), $\delta = 8$	16.8	37.9	24.9	24.1	51.1
	Katz ($\beta = 0.005$), $\delta = 16$	16.5	39.6	35.2	24.7	50.6
unseen bigrams (unweighted)	common neighbors, $\delta = 8$	14.1	40.2	27.9	22.2	39.4
	common neighbors, $\delta = 16$	15.3	39.0	42.5	22.0	42.3
	Katz ($\beta = 0.005$), $\delta = 8$	13.1	36.7	32.3	21.6	37.8
	Katz ($\beta = 0.005$), $\delta = 16$	10.3	29.6	41.8	12.2	37.8
clustering:	$\rho = 0.10$	7.4	37.3	46.9	32.9	37.8
Katz ($\beta_1 = 0.001, \beta_2 = 0.1$)	$\rho = 0.15$	12.0	46.1	46.9	21.0	44.0
	$\rho = 0.20$	4.6	34.3	19.8	21.2	35.7
	$\rho = 0.25$	3.3	27.2	20.5	19.4	17.4

Evaluation: Average relevance performance



- average ratio over the five datasets of the given predictor's performance *versus a baseline* predictor's performance.
- the error bars indicate the minimum and maximum of this ratio over the five datasets.
- the parameters for the starred predictors are: (1) for weighted Katz, $\beta = 0.005$; (2) for Katz clustering, $\beta_1 = 0.001$; $\rho = 0.15$; $\beta_2 = 0.1$; (3) for low-rank inner product, rank = 256; (4) for rooted PageRank, $\alpha = 0.15$; (5) for unseen bigrams, unweighted, common neighbors with $\delta = 8$; and (6) for SimRank, $C(\gamma) = 0.8$.

Evaluation: prediction overlap

How similar are the predictions made by the different methods? Why?

	Adamic/Adar	Katz clustering	common neighbors	hitting time	Jaccard's coefficient	weighted Katz	low-rank inner product	rooted Pagerank	SimRank	unseen bigrams
Adamic/Adar	1150	638	520	193	442	1011	905	528	372	486
Katz clustering		1150	411	182	285	630	623	347	245	389
common neighbors			1150	135	506	494	467	305	332	489
hitting time				1150	87	191	192	247	130	156
Jaccard's coefficient					1150	414	382	504	845	458
weighted Katz						1150	1013	488	344	474
low-rank inner product							1150	453	320	448
rooted Pagerank								1150	678	461
SimRank									1150	423
unseen bigrams										1150

Number of common correct predictions

	Adamic/Adar	Katz clustering	common neighbors	hitting time	Jaccard's coefficient	weighted Katz	low-rank inner product	rooted Pagerank	SimRank	unseen bigrams
Adamic/Adar	92	65	53	22	43	87	72	44	36	49
Katz clustering		78	41	20	29	66	60	31	22	37
common neighbors			69	13	43	52	43	27	26	40
hitting time				40	8	22	19	17	9	15
Jaccard's coefficient					71	41	32	39	51	43
weighted Katz						92	75	44	32	51
low-rank inner product							79	39	26	46
rooted Pagerank								69	48	39
SimRank									66	34
unseen bigrams										68

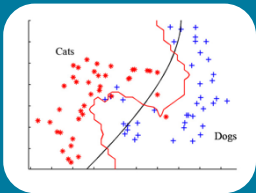
Unsupervised Link Prediction Challenges

- Shortest paths suffer of the small-world effect
- Improve performance. Even the best (Katz clustering on gr-qc) correct on **only about 16%** of its prediction
- Improve **efficiency** on very large networks (approximation of distances)
- Consider **time effect**: most recent links are more important
- Exploit additional information (attributes, text, ...)

Lecture road



Unsupervised methods



Classification approaches



Who to follow

Classification for link prediction

Intuition

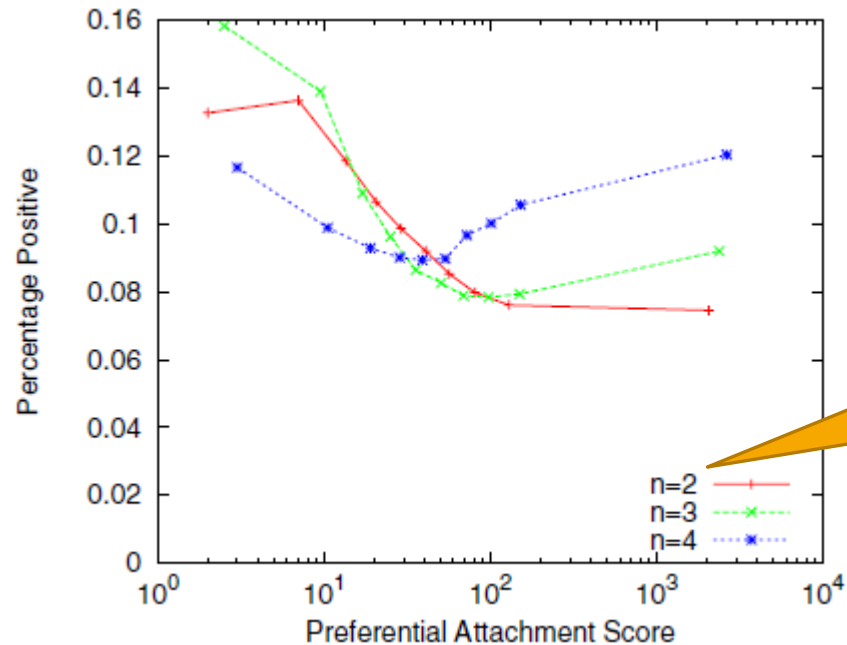
Use any supervised classifier to predict if a link exists (=1) or no (=0)

As features use weak node to node link predictors (e.g., common neighbors)

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)	-		✓

PropFlow: special random walk stopping at size l or when cycle

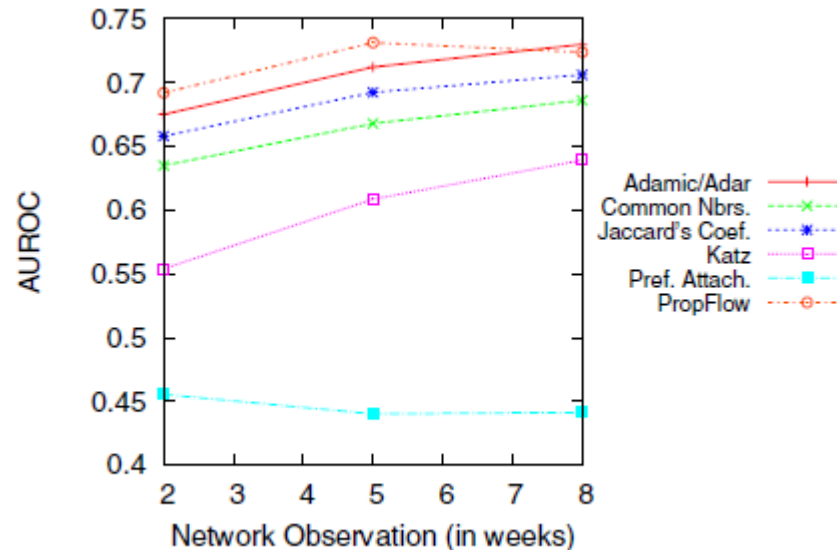
Why Supervised learning for LP?



Restricted to n -neighbors
(just look at the links
from one node to
nodes at distance n)

- Unsupervised methods like those that we have seen so far might work well with some network but do not generalize to others
- Features are dependent

How to get training data?



- τ_x length (in time) of computing features
- τ_y length of determining the class attribute
- Large $\tau_x \Rightarrow$ better quality of features as the network reaches saturation
- Increasing $\tau_y \Rightarrow$ increases labeled data and final performance

Metrics for Performance Evaluation

- **Confusion Matrix:** contains the number of
 - True positive: correctly predicted links that are actually links
 - True negative: number of correctly predicted non-links
 - False positive: number of predicted links that are not links
 - False negative: number of non-predicted links that are actually links

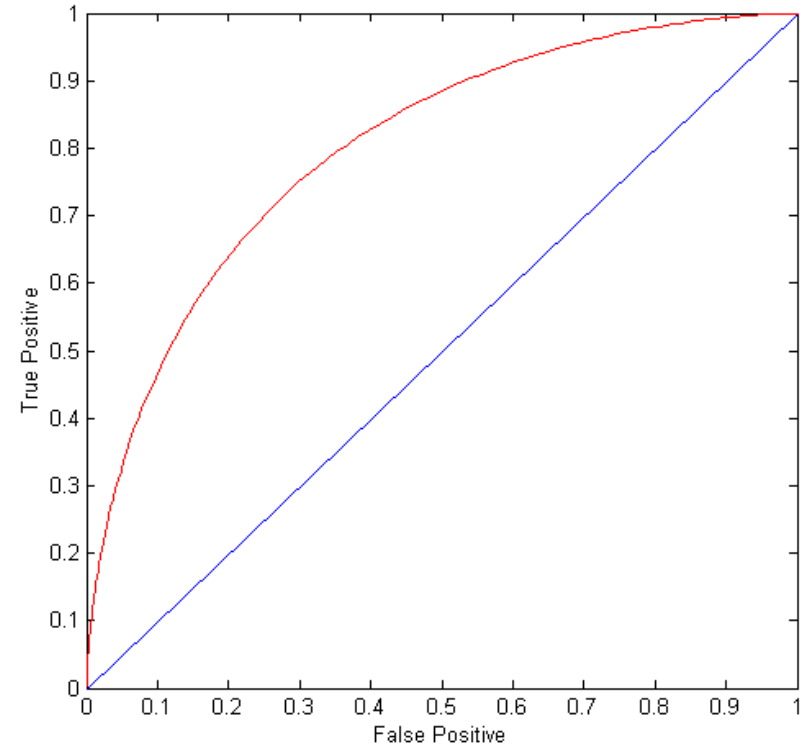
	PREDICTED CLASS		
	Class=Yes	Class=No	
ACTUAL CLASS	Class=Yes	TP	FN
	Class=No	FP	TN

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

ROC Curve

ROC stands for Receiver Operating Characteristic

- Show the performance of a binary classifier
 - $\text{TPR (sensitivity)} = \text{TP} / (\text{TP} + \text{FN})$ (how many data points are correctly classified among those that are actually positive)
 - $\text{FPR} = \text{FP} / (\text{TN} + \text{FP})$ (percentage of negative classified as positive)
- (0,0): declare everything to be negative class
- (1,1): declare everything to be positive class
- (0,1): ideal
- **Diagonal line: Random guessing**
- Below diagonal line: prediction is worse than random



AUC: area under the ROC curve

Drawing a ROC curve

Pairs of nodes ordered by score (parameter k)

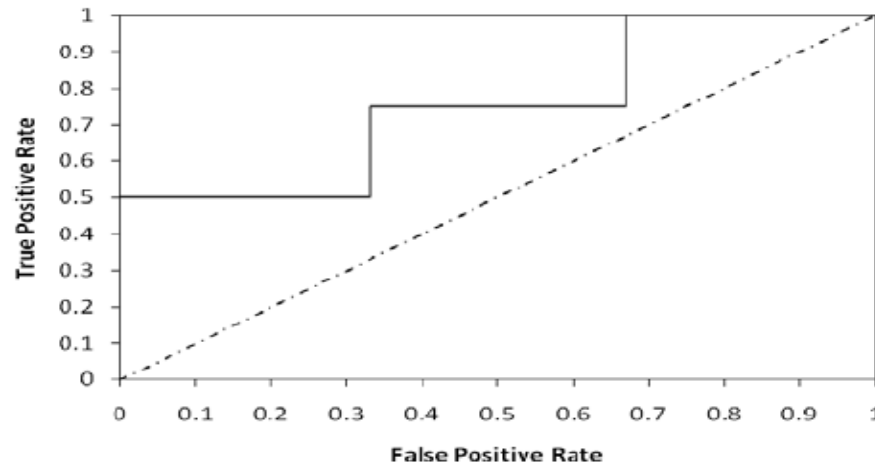
Number of correctly detected links if only considered the first k pairs

Number of links that are not link recognized in the top-k

Number of non-links correctly detected

Number of links not detected

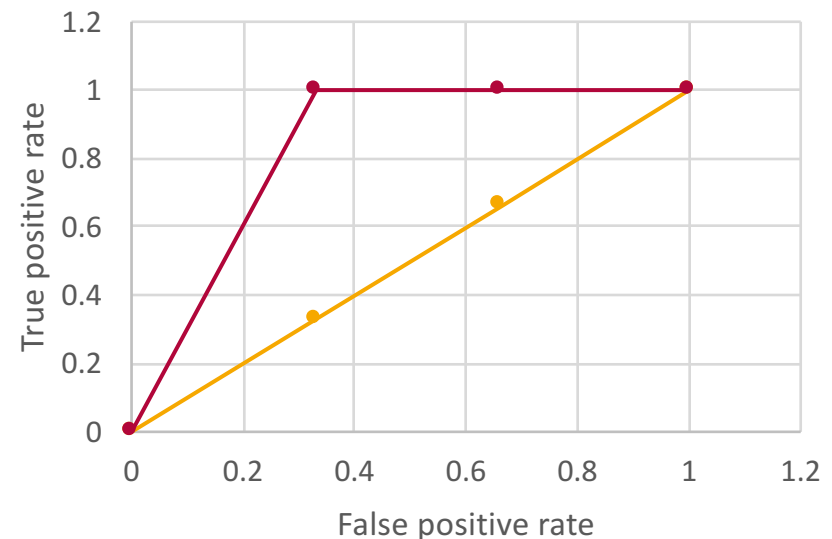
Rank		1	2	3	4	5	6	7	8	9	10
Actual class		+	+	-	-	+	-	-	+	-	-
TP	0	1	2	2	2	3	3	3	4	4	4
FP	0	0	0	1	2	2	3	4	4	5	6
TN	6	6	6	5	4	4	3	2	2	1	0
FN	4	3	2	2	2	1	1	1	0	0	0
TPR	0	0.25	0.5	0.5	0.5	0.75	0.75	0.75	1	1	1
FPR	0	0	0	0.17	0.33	0.33	0.50	0.67	0.67	0.83	1



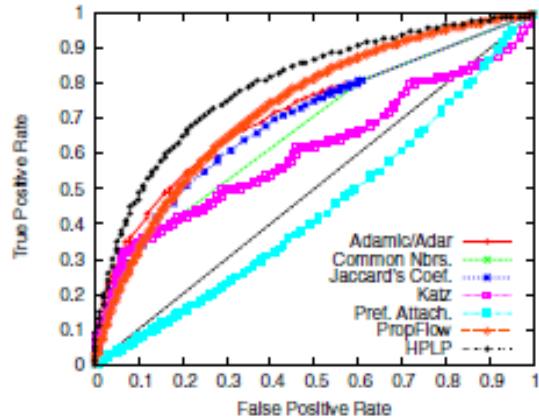
ROC curve: an example

- Assume that you have 4 nodes => 6 pairs
- Order the pairs by decreasing score
- Mark if the predicted link at that threshold is actually a link
- Compute TP, TN, FP, FN, TPR, FPR

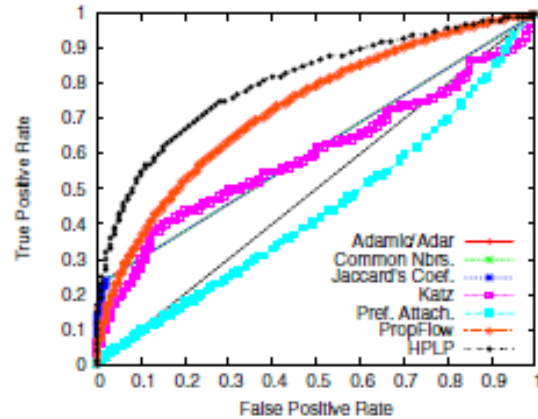
	(1,2)	(2,3)	(3,4)	(1,4)	(2,4)	(1,3)
Score	0.9	0.75	0.6	0.5	0.4	0.2
Actual Link	Yes	Yes	No	Yes	No	No
TP	1	2	2	3	3	3
TN	3	3	2	2	1	0
FP	0	0	1	1	2	3
FN	2	1	1	0	0	0
TPR	1/3	2/3	2/3	1	1	1
FPR	0	0	1/3	1/3	2/3	1



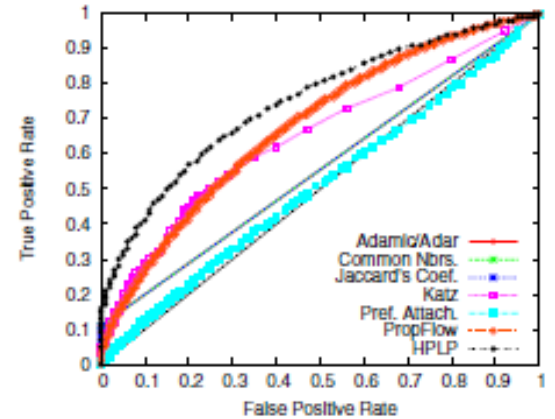
Results



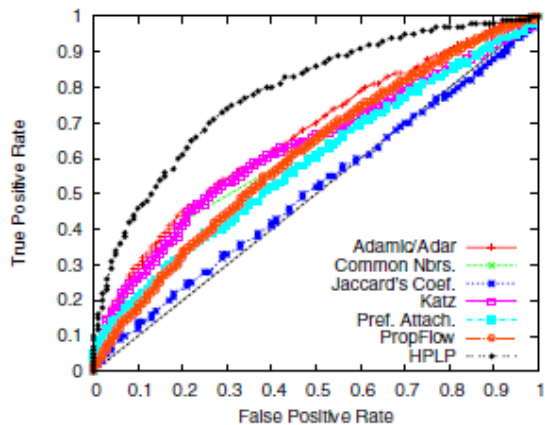
(a) phone $n = 2$



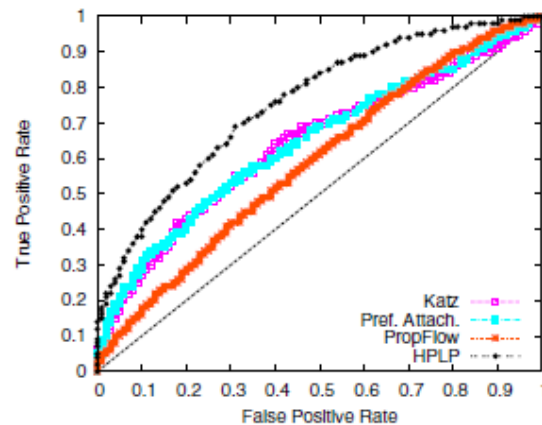
(b) phone $n = 3$



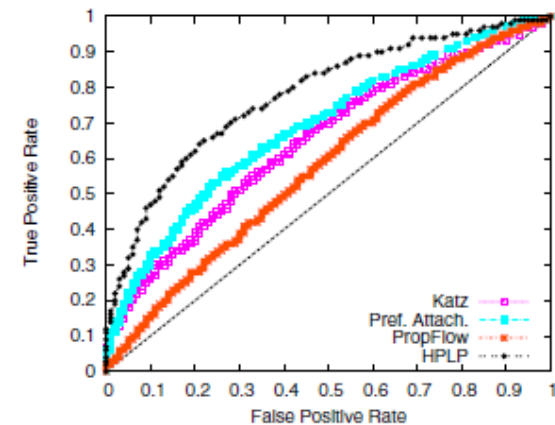
(c) phone $n = 4$



(d) condmat $n = 2$



(e) condmat $n = 3$



(f) condmat $n = 4$

Lecture road



Unsupervised methods



Classification approaches



Who to follow

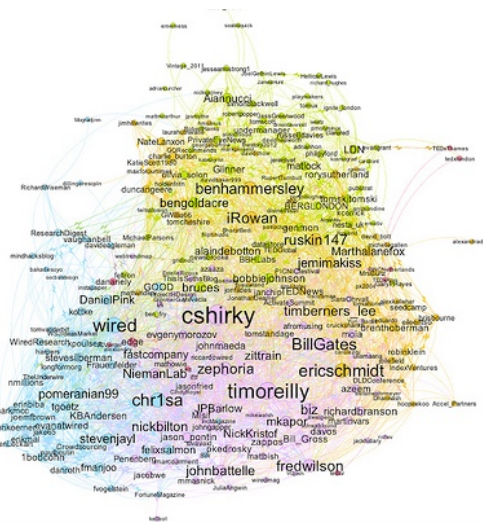
Who to Follow

- TwitWtf ("Who to Follow"): *the Twitter user recommendation service*
- 317 million users, 500 million tweets every day (2016)
<http://www.internetlivestats.com/twitter-statistics/>
- Twitter needs to help existing and new users to discover connections to sustain and grow
- Also used for search relevance, discovery, promoted products, etc.



The Twitter graph

- Node: user (directed) edge: follows
- Statistics (August 2012)
 - over *20 billion edges* (only active users)
 - *power law* distributions of in-degrees and out-degrees.
 - over 1000 with more than 1 million followers,
 - 25 users with more than 10 million followers.



<http://blog.ouseful.info/2011/07/07/visualising-twitter-friend-connections-using-gephi-an-example-using-wireduk-friends-network/>

Algorithms: Circle of trust

Circle of trust: the result of an egocentric random walk (similar to personalized (rooted) PageRank)

- Computed in an online fashion (from scratch each time) given a set of parameters (# of random walk steps, reset probability, pruning settings to discard low probability vertices, parameters to control sampling of outgoing edges at vertices with large out-degrees, etc.)
- Used in a variety of Twitter products, e.g., in search and discovery, content from users in one's circle of trust upweighted

Algorithms

- **Asymmetric nature** of the follow relationship (other social networks e.g., Facebook or LinkedIn require the consent of both participating members)
- Directed edge case is similar to the **user-item recommendations** problem where the “item” is also a user.

Algorithms: SALSA

SALSA (Stochastic Approach for Link-Structure Analysis)

a variation of HITS

As in HITS

hubs

authorities

HITS

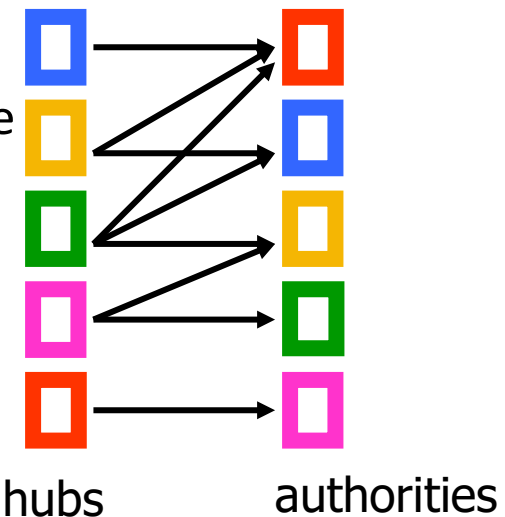
- Good hubs point to good authorities
- Good authorities are pointed to by good hubs

hub weight = sum of the authority weights of the authorities pointed to by the hub

$$h_i = \sum_{j:i \rightarrow j} a_j$$

authority weight = sum of the hub weights that point to this authority.

$$a_i = \sum_{j:j \rightarrow i} h_j$$



In the next episode ...

Student presentations

Community detection

And much more ...

ANY
QUESTIONS
?

References

- Lichtenwalter, R.N., Lussier, J.T. and Chawla, N.V.. **New perspectives and methods in link prediction.** KDD, 2010.
- Lü, L. and Zhou, T., 2011. **Link prediction in complex networks: A survey.** *Physica A: Statistical Mechanics and its Applications*, 390(6), pp.1150-1170.
- Liben-Nowell, D. and Kleinberg, J., 2007. **The link-prediction problem for social networks.** *Journal of the American society for information science and technology*, 58(7), pp.1019-1031.
- Gupta, P., Goel, A., Lin, J., Sharma, A., Wang, D. and Zadeh, R. **Wtf: The who to follow service at twitter.** WWW, 2013.