EECS 510 Social Media Mining
Lecture 3: Link Prediction for Social Networks

Spring 2016

Rosanne Liu

rosanne.liu@northwestern.edu

Electrical Engineering and Computer Science
Northwestern University

April 7, 2016

Administrative

Two main assignments of the class: paper review and term project.

- For each, form groups of 2.

Paper review:

- Each group chooses one topic to present on a selected lecture date (starting May 17).
- Groups other than the presenting one are to post paper reviews before the lecture (starting May 17).
- Presenters present both (or 2 out of 3) papers; reviewers write for 1.

Term project:

- Pick from suggested ideas.
- Come to me with your own ideas.
- Submit progress reports and eventually, the code.

Administrative

Start early!

- Before you notice you'll have to read 3 papers a day.
- View paper discussion as engagements other than assignments.
- Make your presentation interesting.
- Major ingredients of your project: data, problem, method.

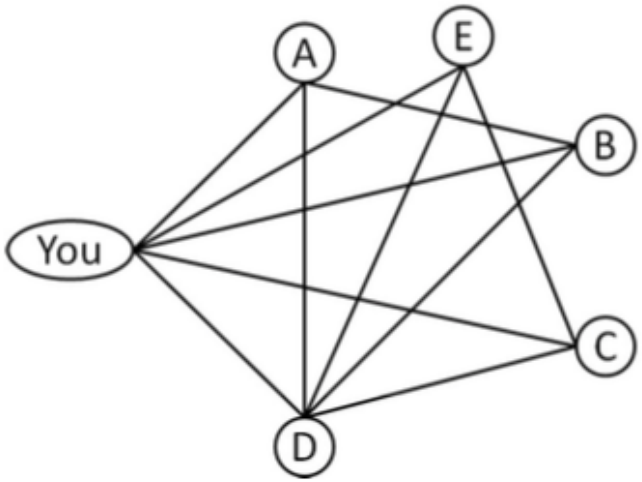
Lecture Outline

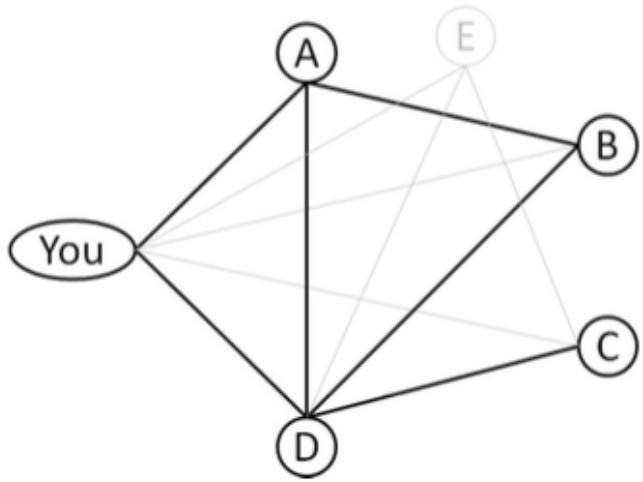
1. Graph Basics, Link Prediction Problem
2. Unsupervised Metrics
3. Supervised Learning
4. Other Methods

Outline

A quick recap of graph basics

The whole network and the observed network, $G(V, E)$





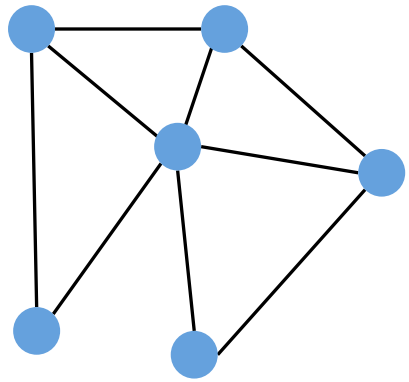
Type of network	Type of interaction predicted
Social	Friendships Collaborations Collusion
Biological	Protein-protein interactions in biological processes Food webs - how different organisms interact with each other and their environment
Information Systems	User-item interactions - recommender systems

Motivation

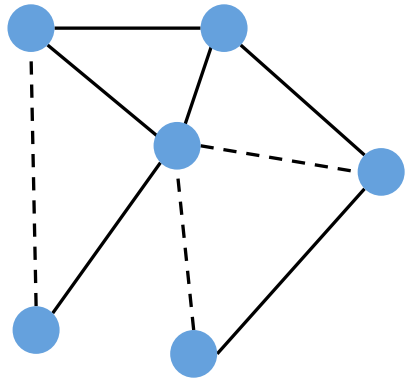
- Recommending new friends in online social networks, a.k.a., who-follows-who problem.
- Suggesting new interactions in online social networks, a.k.a., who-retweets-who problem.
- Predicting connections between members of terrorist organizations who have or have not been directly observed to work together.
- Suggesting collaborations between researchers based on co-authorship.

Link prediction

It is a canonical network topology inference problem.
Q: If a portion of the graph is unobserved, can we infer it from data?



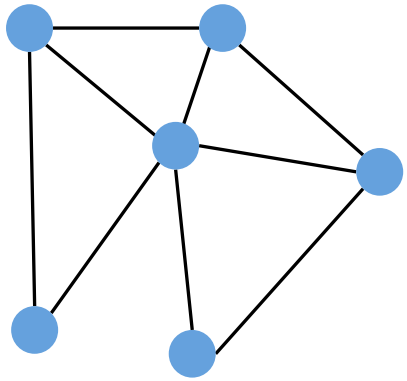
Original graph



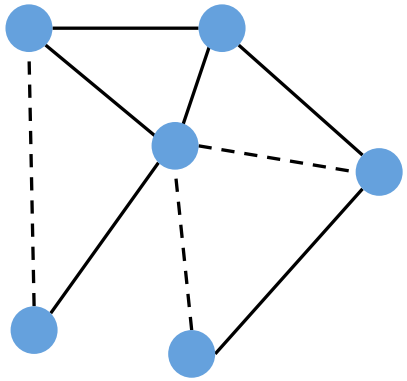
Observed graph

Suppose we observe a vertex set V in social network $G(V, E)$
For all vertex pairs $\{i, j\} \in V^{(2)}$, edges are only observed for some subset $V_{obs}^{(2)} \subset V^{(2)}$.
Goal: predict edge status for all other pairs, i.e., $V_{miss}^{(2)} = V^{(2)} \setminus V_{obs}^{(2)}$.

Link prediction



Original graph

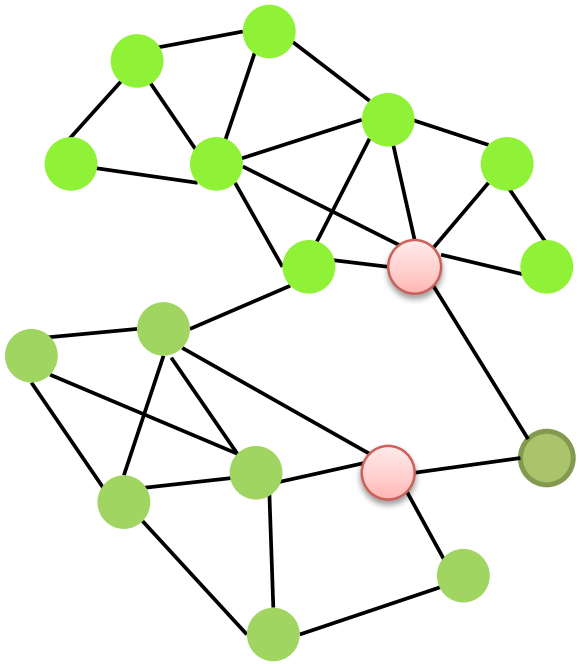


Observed graph

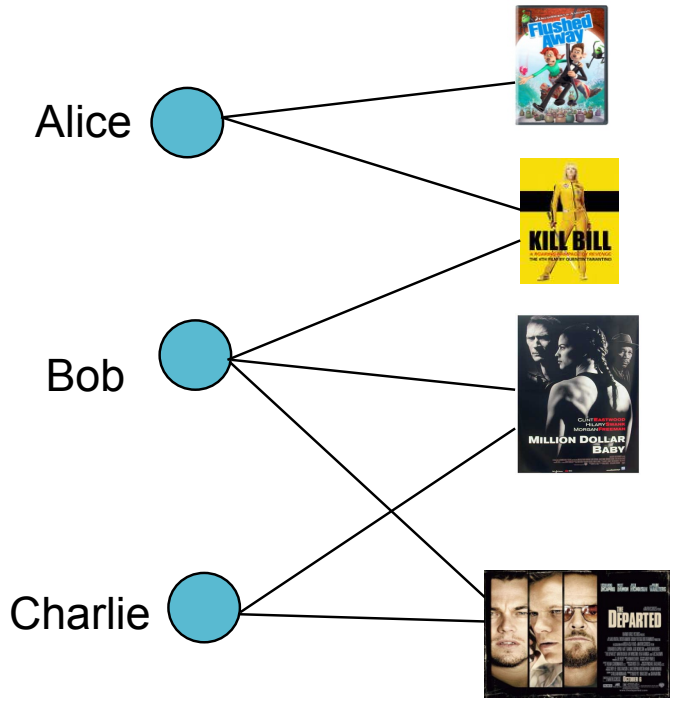
- Graph $G(V, E)$
- Number of "missing edges": $|V|(|V|-1)/2 - |E|$
- In sparse graphs $|E| \ll |V|^2$, Prob. of correct random guess $O(\frac{1}{|V|^2})$

Link prediction

Social link prediction examples:



Friend recommendation in Facebook



Movie recommendation in Netflix

Definition

- **Link prediction problem:** Given the links in a social network at time t or during a time interval I, we wish to predict the links that will be added to the network during the later time interval from time t' to a some given future time.

- **Main approach:** Use measures of network-proximity adapted from graph theory, computer science, and the social sciences to determine which unconnected nodes are close together in the topology of the network.

Outline

Proximity scoring

Link prediction by proximity scoring

- For each pair of nodes compute proximity (similarity) score $score(v_1, v_2)$
- Sort all pairs by the decreasing score
- Select top n pairs (or above some threshold) as new links

Neighborhood based proximity metrics

Let $\mathcal{N}(v_i)$ denote the set of local neighbors of v_i

- **Common neighbors**: Based on the idea that links are formed between nodes who share many common neighbors

$$score(v_i, v_j) = \mid \mathcal{N}(v_i) \cap \mathcal{N}(v_j) \mid$$

Neighborhood based proximity metrics

Let $\mathcal{N}(v_i)$ denote the set of local neighbors of v_i

- **Common neighbors**: Based on the idea that links are formed between nodes who share many common neighbors

$$score(v_i, v_j) = \mid \mathcal{N}(v_i) \cap \mathcal{N}(v_j) \mid$$

- **Jaccard's coefficient**: Measure how likely a neighbor of v_i is to be a neighbor of v_j and vice versa

$$score(v_i, v_j) = \frac{\mid \mathcal{N}(v_i) \cap \mathcal{N}(v_j) \mid}{\mid \mathcal{N}(v_i) \cup \mathcal{N}(v_j) \mid}$$

Neighborhood based proximity metrics

Let $\mathcal{N}(v_i)$ denote the set of local neighbors of v_i

- **Common neighbors**: Based on the idea that links are formed between nodes who share many common neighbors

$$score(v_i, v_j) = \mid \mathcal{N}(v_i) \cap \mathcal{N}(v_j) \mid$$

- **Jaccard's coefficient**: Measure how likely a neighbor of v_i is to be a neighbor of v_j and vice versa

$$score(v_i, v_j) = \frac{\mid \mathcal{N}(v_i) \cap \mathcal{N}(v_j) \mid}{\mid \mathcal{N}(v_i) \cup \mathcal{N}(v_j) \mid}$$

- **Adamic/Adar**: Assigns large weight to common neighbors x of v_i and v_j which themselves have few neighbors $\mid \mathcal{N}(x) \mid$

$$score(v_i, v_j) = \sum_{x \in \mathcal{N}(v_i) \cap \mathcal{N}(v_j)} \frac{1}{\log \mid \mathcal{N}(x) \mid}$$

Neighborhood based proximity metrics

Let $\mathcal{N}(v_i)$ denote the set of local neighbors of v_i

- **Preferential attachment**: Based on the premise that a new edge has node v_i as its endpoint is proportional to $\mid \mathcal{N}(v_i) \mid$. i.e., nodes like to form ties with 'popular' nodes

$$score(v_i, v_j) = \mid \mathcal{N}(v_i) \mid \cdot \mid \mathcal{N}(v_j) \mid$$

  Researchers found empirical evidence to suggest that co-authorship is correlated with the product of the neighborhood sizes.

- A variant:

$$score(v_i, v_j) = \mid \mathcal{N}(v_i) \mid + \mid \mathcal{N}(v_j) \mid$$

Path based proximity metrics

Paths and ensembles of paths between v_i and v_j

- **Shortest path**:

$$-\min_x \{path_{ij}^x > 0\}$$

Path based proximity metrics

Paths and ensembles of paths between v_i and v_j

- **Shortest path**:

$$-\min_x \{path_{ij}^x > 0\}$$

- **Katz score**: Sums over all possible paths between v_i and v_j, giving higher weight to shorter paths.

$$score(v_i, v_j) = \sum_{l=1}^{\infty} \beta^l \mid paths^{(l)}(v_i, v_j) \mid$$

Path based proximity metrics

Paths and ensembles of paths between v_i and v_j

- **Shortest path**:

$$-\min_x \{path_{ij}^x > 0\}$$

- **Katz score**: Sums over all possible paths between v_i and v_j, giving higher weight to shorter paths.

$$score(v_i, v_j) = \sum_{l=1}^{\infty} \beta^l \mid paths^{(l)}(v_i, v_j) \mid$$

- Two variants of the Katz measure are considered
  - *unweighted:* $paths^{(l)}(v_i, v_j) = 1$ if v_i and v_j have collaborated and 0 otherwise
  - *weighted:* $paths^{(l)}(v_i, v_j)$ is the number of times that v_i and v_j have collaborated.

Random walk based proximity metrics

Consider a random walk on G which starts at v_i and iteratively moves to a neighbor of v_i chosen uniformly at random from $\mathcal{N}(v_i)$.

- **Hitting Time**: H_{v_i,v_j} from v_i to v_j is the expected number of steps it takes for the random walk starting at v_i to reach v_j.

$$score(v_i, v_j) = -H_{v_i,v_j}$$

Random walk based proximity metrics

Consider a random walk on G which starts at v_i and iteratively moves to a neighbor of v_i chosen uniformly at random from $\mathcal{N}(v_i)$.

- **Hitting Time**: H_{v_i,v_j} from v_i to v_j is the expected number of steps it takes for the random walk starting at v_i to reach v_j.

$$score(v_i, v_j) = -H_{v_i,v_j}$$

- **Commute Time**: $C_{v_i,v_j} = H_{v_i,v_j} + H_{v_j,v_i}$ is the expected number of steps to travel from v_i to v_j then back to v_i.

$$score(v_i, v_j) = -C_{v_i,v_j} = -(H_{v_i,v_j} + H_{v_j,v_i})$$

Random walk based proximity metrics

Consider a random walk on G which starts at v_i and iteratively moves to a neighbor of v_i chosen uniformly at random from $\mathcal{N}(v_i)$.

- **Hitting Time**: H_{v_i,v_j} from v_i to v_j is the expected number of steps it takes for the random walk starting at v_i to reach v_j.

$$score(v_i, v_j) = -H_{v_i,v_j}$$

- **Commute Time**: $C_{v_i,v_j} = H_{v_i,v_j} + H_{v_j,v_i}$ is the expected number of steps to travel from v_i to v_j then back to v_i.

$$score(v_i, v_j) = -C_{v_i,v_j} = -(H_{v_i,v_j} + H_{v_j,v_i})$$

- Normalized hitting/commute time: stationary-normed versions

$$score(v_i, v_j) = -H_{v_i,v_j}\pi_{v_j}$$
$$score(v_i, v_j) = -(H_{v_i,v_j}\pi_{v_j} + H_{v_j,v_i}\pi_{v_i})$$

Random walk based proximity metrics

The hitting time and commute time measures are sensitive to parts of the graph far away from v_i and v_j, even when v_i and v_j are connected by very short paths.

A way of counteracting this is to allow the random walk from v_i to v_j to periodically "reset", returning to v_i with a fixed probability at each step; in this way, distant parts of the graph will almost never be explored.

- **Personalized (Rooted) PageRank**: Consider the random walk on G. At each step, it has a probability of α of jumping to v_i, and a probability $1 - \alpha$ going to a random neighbor.

  PR $=$ The stationary distribution weight of v_j under this scheme.

Random walk based proximity metrics

- **SimRank**: Let $similarity(v_i, v_j)$ be a fixed point of

$$similarity(v_i, v_j) = \gamma \frac{\sum_{a \in \mathcal{N}(v_i)} \sum_{b \in \mathcal{N}(v_j)} similarity(a, b)}{\mid \mathcal{N}(v_i) \mid\mid \mathcal{N}(v_j) \mid}$$

where $\gamma \in [0, 1]$

$$score(v_i, v_j) = similarity(v_i, v_j)$$

This is the expected value of γ^l under the random walk probabilities, where l is the time at which random walks started from v_i and v_j first meet

Vertex feature aggregations based proximity metrics

- **Clustering Coefficient**:

$$score(v_i, v_j) = CC(v_i) \cdot CC(v_j)$$

or

$$score(v_i, v_j) = CC(v_i) + CC(v_j)$$

Comparison of proximity metrics

A most cited paper for this problem:

The Link Prediction Problem for Social Networks*

David Liben-Nowell†
Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139 USA
dln@theory.lcs.mit.edu

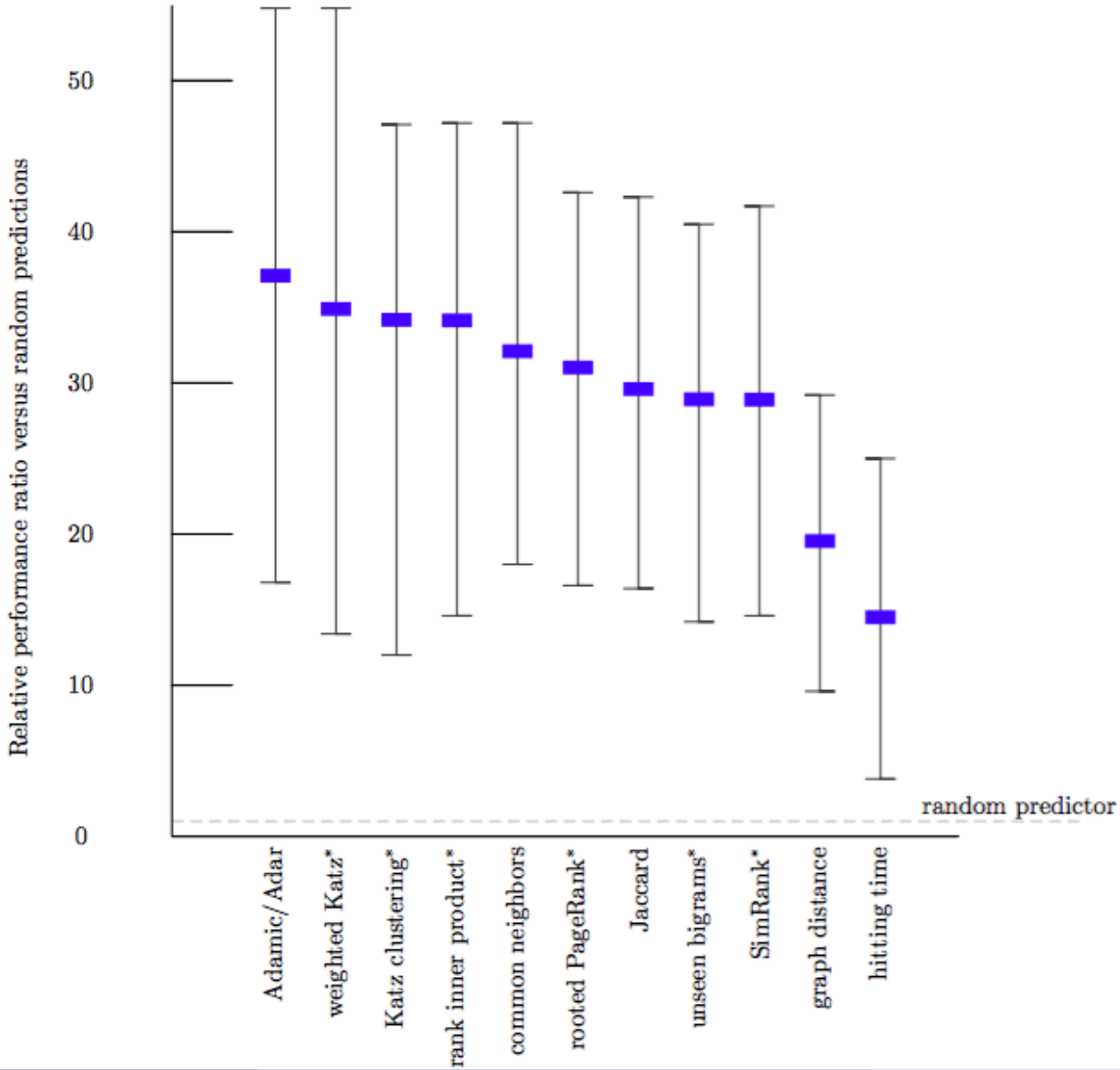
Jon Kleinberg‡
Department of Computer Science
Cornell University
Ithaca, NY 14853 USA
kleinber@cs.cornell.edu

January 8, 2004

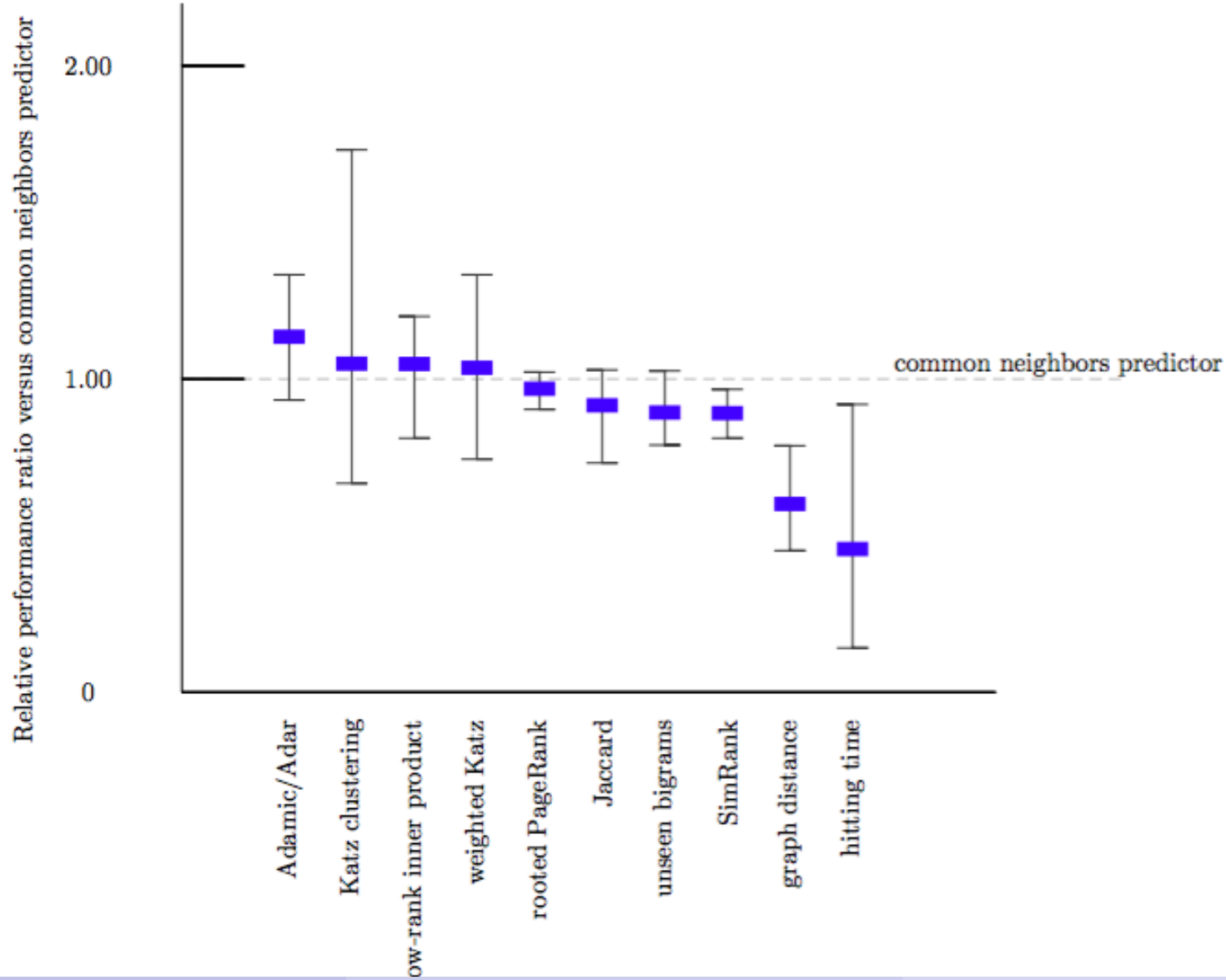
Abstract

Given a snapshot of a social network, can we infer which new interactions among its members are likely to occur in the near future? We formalize this question as the *link prediction problem*, and develop approaches to link prediction based on measures for analyzing the "proximity" of nodes in a network. Experiments on large co-authorship networks suggest that information about future interactions can be extracted from network topology alone, and that fairly subtle measures for detecting node proximity can outperform more direct measures.

Comparison of proximity metrics



Comparison of proximity metrics



Outline

Link prediction as binary classification

Treat link prediction as a binary classification problem.
It is a challenging problem:

- Computational cost of evaluating every large number of possible edges ($|V|^2$ - quadratic in number of nodes).
- Highly imbalanced class distribution: number of positive examples (existing edges) grows linearly and negative examples quadratically with number of nodes.

Prediction difficulty

Actual and possible collaborations between DBLP authors.

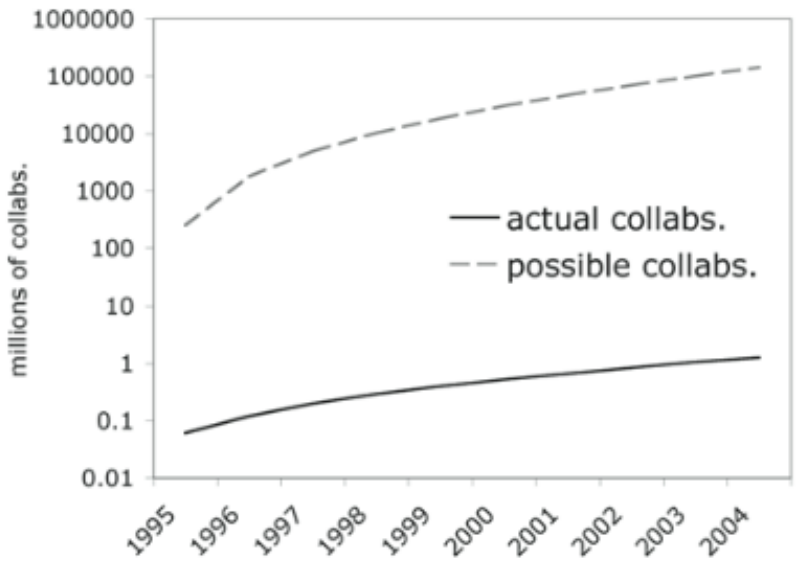


Figure 1. Logarithmic plot of actual and possible collaborations between DBLP authors, 1995-2004.

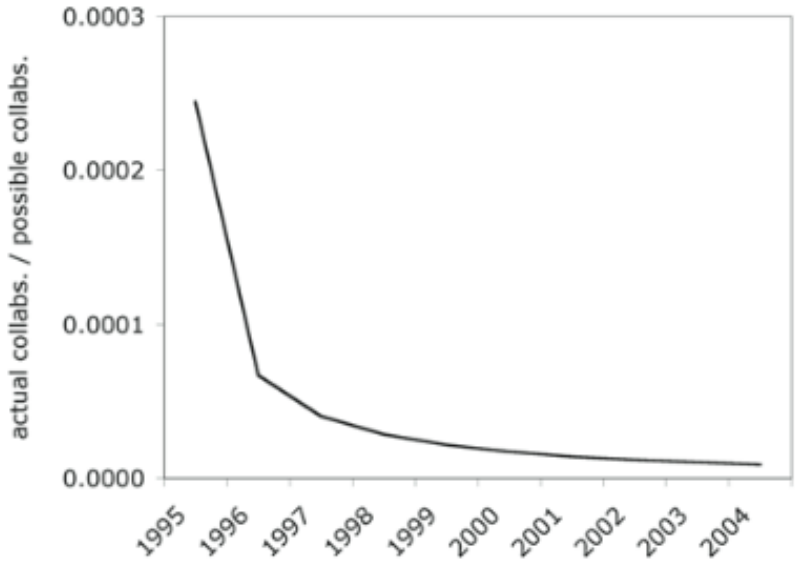


Figure 2. Publications of DBLP authors as a proportion of possible collaborations, 1995-2004.

[Rattigan, Matthew J., and David Jensen. "The case for anomalous link discovery." ACM SIGKDD Explorations Newsletter 7.2 (2005): 41-47.]

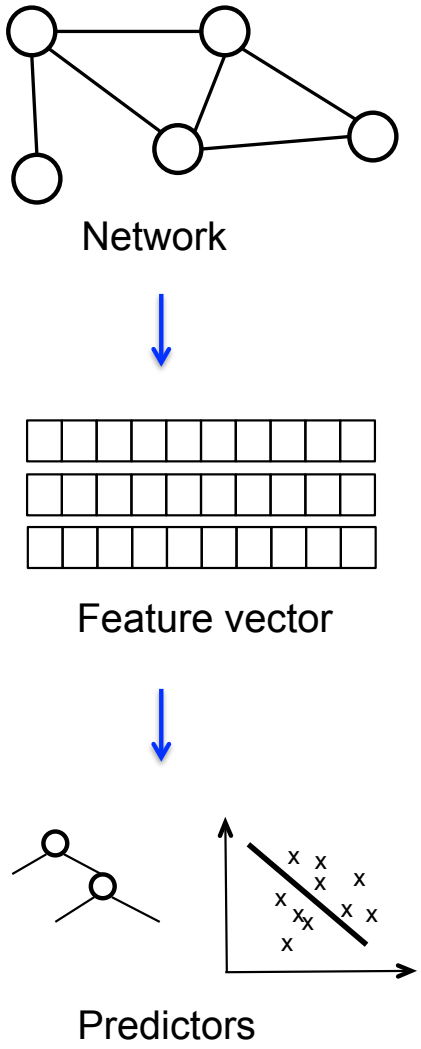
Link prediction as binary classification

Supervised Learning:

1. Feature generation/extraction
2. Feature selection (optional)
3. Model training
4. Testing (model appication)

Features:

- Topological proximity features
- Aggregated features
- Content based node proximity features



Evaluation metrics

- Precision and Recall, F-measure

$$Precision = \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN}$$

$$F = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

- True positive rate (TPR), False positive rate (FPR), ROC curve, AUC

$$TPR = \frac{TP}{TP + FN}, \quad FPR = \frac{FP}{FP + TN}$$

Performance of classification algorithms

Two collaboration databases: BIOBASE, DBLP.

Classification model	Accuracy	Precision	Recall	F-value	Squared Error
Decision Tree	90.01	91.60	89.10	90.40	0.1306
SVM(Linear Kernel)	87.78	92.80	83.18	86.82	0.1221
SVM(RBF Kernel)	90.56	92.43	88.66	90.51	0.0945
K_Nearest Neighbors	88.17	92.26	83.63	87.73	0.1826
Multilayer Perceptron	89.78	93.00	87.10	90.00	0.1387
RBF Network	83.31	94.90	72.10	81.90	0.2542
Naive Bayes	83.32	95.10	71.90	81.90	0.1665
Bagging	90.87	92.5	90.00	91.23	0.1288

Table 2: Performance of different classification algorithms for BIOBASE database

Classification model	Accuracy	Precision	Recall	F-value	Squared Error
Decision Tree	82.56	87.70	79.5	83.40	0.3569
SVM(Linear Kernel)	83.04	85.88	82.92	84.37	0.1818
SVM(RBF Kernel)	83.18	87.66	80.93	84.16	0.1760
K_Nearest Neighbors	82.42	85.10	82.52	83.79	0.2354
Multilayer Perceptron	82.73	87.70	80.20	83.70	0.3481
RBF Network	78.49	78.90	83.40	81.10	0.4041
Naive Bayes	81.24	87.60	76.90	81.90	0.4073
Bagging	82.13	86.70	80.00	83.22	0.3509

Table 3: Performance of different classification algorithms for DBLP dataset

[Al Hasan, Mohammad, et al. "Link prediction using supervised learning." SDM06: Workshop on Link Analysis, Counter-terrorism and Security. 2006.]

Outline

Probabilistic models

- Local model, Markov random fields [1]
- Hierarchical probabilistic model [2]
- Probabilistic relations models:
  - Bayesian networks [3]
  - relational Markov networks [4]
- Dynamic graphs?

[1] Wang, Chao, Venu Satuluri, and Srinivasan Parthasarathy. "Local probabilistic models for link prediction." Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on. IEEE, 2007.
[2] Clauset, Aaron, Cristopher Moore, and Mark EJ Newman. "Hierarchical structure and the prediction of missing links in networks." Nature 453.7191 (2008): 98-101.
[3] Getoor, Lise. "Link mining: a new data mining challenge." ACM SIGKDD Explorations Newsletter 5.1 (2003): 84-89.
[4] Taskar, Ben, et al. "Link prediction in relational data." Advances in neural information processing systems. 2003.

Thank You!

Next lecture series: Data Mining Essentials