



Introduction to Data Mining

Link Prediction

U Kang
Seoul National University



In This Lecture

- Link prediction: problem definition, motivation, and applications
- Methods
 - Based on node similarity



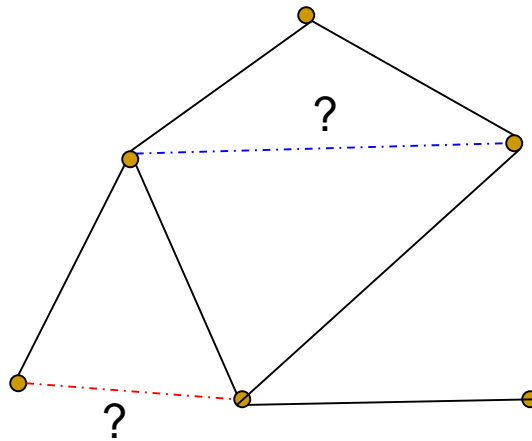
Outline

- ➔ **Problem Definition**
- Methods based on Node Similarity
- Results based on Node Similarity



Link Prediction Problem

- Given: a snapshot of a social network (or graph)
- Infer: which new interactions among its members are likely to occur in the near future





Link Prediction

- Answer to the following question:
 - To what extent can the evolution of a social network be modeled using features intrinsic to the network itself?
- “A network model is useful to the extent that it can support meaningful inferences from observed network data”



Link Prediction

- Application
 - Link recommendation (e.g. Facebook)
 - Predict social network evolution (e.g. Twitter)
 - Suggest promising interactions or collaborations in a company with hierarchical organization



Data

- Co-authorship network from arXiv
 - Astro-ph : astrophysics
 - Cond-mat : condensed matter
 - Gr-qc : general relativity and quantum cosmology
 - Hep-ph : high energy physics-phenomenology
 - Hep-th : high energy physics - theory



Evaluation

- Choose four timestamps
 - $t_0 < t_0' < t_1 < t_1'$
- Use $G[t_0, t_0']$ as training data
 - Predict future links
- Use $G[t_1, t_1']$ as test data
 - Evaluate the prediction



Outline

Problem Definition

 **Methods based on Node Similarity**

Results based on Node Similarity

David Liben-Nowell, Jon Kleinberg, The Link Prediction Problem for Social Networks



Method for Link Prediction

- Rank according to the 'similarity': $\text{score}(x,y)$
 - Graph distance
 - Methods based on node neighborhoods
 - Methods based on the ensemble of all paths

 - Higher-level approaches
 - Can be combined with the approaches above



Graph Distance

- $\text{score}(x,y)$ = (negated) length of shortest path between x and y



Methods based on node neighborhoods

$\Gamma(x)$: set of neighbors of x

common neighbors	$ \Gamma(x) \cap \Gamma(y) $
Jaccard's coefficient	$\frac{ \Gamma(x) \cap \Gamma(y) }{ \Gamma(x) \cup \Gamma(y) }$
Adamic/Adar	$\sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log \Gamma(z) }$
preferential attachment	$ \Gamma(x) \cdot \Gamma(y) $



Common Neighbors and Jaccard's Coefficient

$\Gamma(x)$: set of neighbors of x

common neighbors	$ \Gamma(x) \cap \Gamma(y) $
Jaccard's coefficient	$\frac{ \Gamma(x) \cap \Gamma(y) }{ \Gamma(x) \cup \Gamma(y) }$

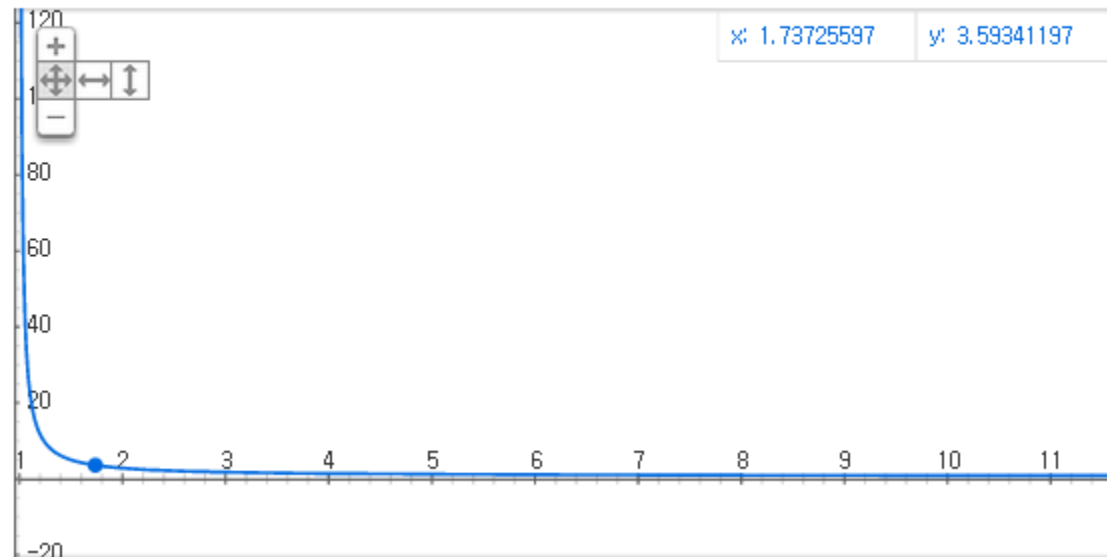


Adamic/Adar

Adamic/Adar

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

$$1/\log(x) - 1/x$$



$\log()$ improves the contribution of low degree nodes



Preferential Attachment

$\Gamma(x)$: set of neighbors of x

preferential attachment $|\Gamma(x)| \cdot |\Gamma(y)|$

- Intuition: “rich gets richer” = power law



Methods based on the ensemble of all paths

- Katz proximity

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

where $\text{paths}_{x,y}^{\langle \ell \rangle} := \{\text{paths of length exactly } \ell \text{ from } x \text{ to } y\}$

weighted: $\text{paths}_{x,y}^{\langle 1 \rangle} := \text{number of collaborations between } x, y.$

unweighted: $\text{paths}_{x,y}^{\langle 1 \rangle} := 1 \text{ iff } x \text{ and } y \text{ collaborate}$

- Solution : $(I - \beta M)^{-1} - I$



Methods based on the ensemble of all paths

- Hitting time, commute time

hitting time	$-H_{x,y}$
stationary-normed	$-H_{x,y} \cdot \pi_y$
commute time	$-(H_{x,y} + H_{y,x})$
stationary-normed	$-(H_{x,y} \cdot \pi_y + H_{y,x} \cdot \pi_x)$

where $H_{x,y} :=$ expected time for random walk from x to reach y
 $\pi_y :=$ stationary distribution weight of y
(proportion of time the random walk is at node y)



Methods based on the ensemble of all paths

- Rooted PageRank = (RWR)

stationary distribution weight of y under the following random walk:
with probability α , jump to x .
with probability $1 - \alpha$, go to random neighbor of current node.



Methods based on the ensemble of all paths

- SimRank: $\text{score}(x,y)$ is defined by

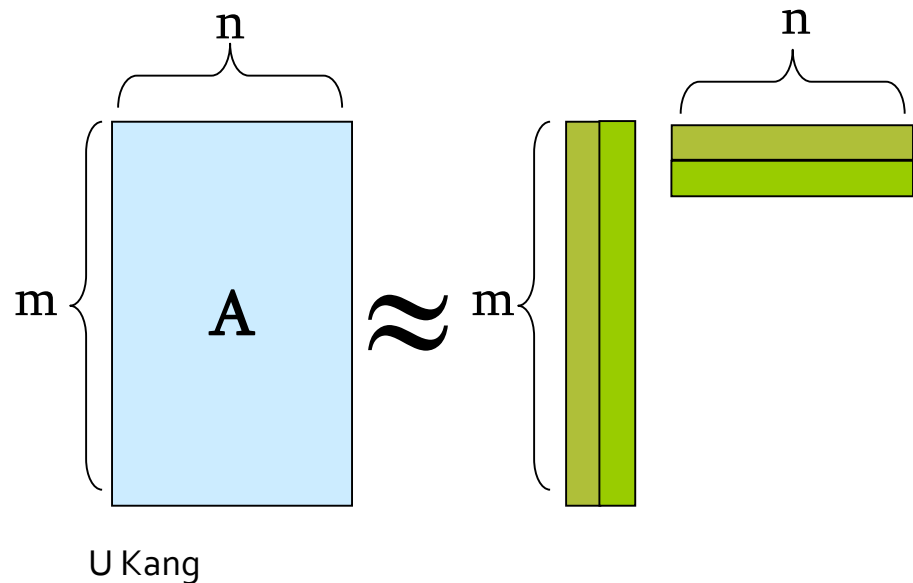
$$\begin{cases} 1 & \text{if } x = y \\ \gamma \cdot \frac{\sum_{a \in \Gamma(x)} \sum_{b \in \Gamma(y)} \text{score}(a,b)}{|\Gamma(x)| \cdot |\Gamma(y)|} & \text{otherwise} \end{cases}$$

- Intuition: x and y are similar if its neighbors are similar
- The expected value of γ^l , where l is a random variable giving the time at which random walks started from x and y first meet



Higher-Level Approaches

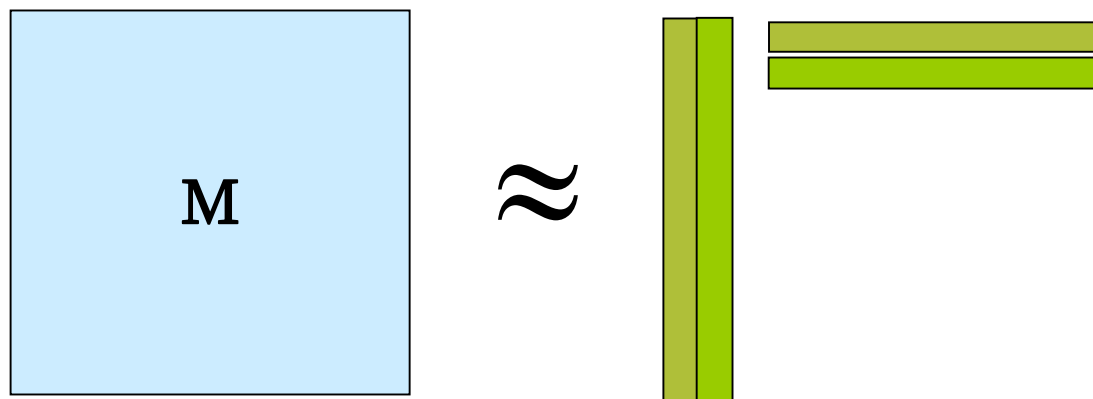
1. Low rank approximation of M by M_k
 - “noise reduction” technique
 - E.g.
 - Ranking by the Katz measure on M_k
 - Common neighbors using M_k
 - $\text{Score}(x,y) = M_k(x,y)$





Higher-Level Approaches

1. Low rank approximation of M by M_k
 - “noise reduction” technique
 - E.g.
 - Ranking by the Katz measure on M_k
 - Common neighbors using M_k
 - $\text{Score}(x,y) = M_k(x,y)$





Higher-Level Approaches

2. Unseen bigram

- Augment our estimates $\text{score}(x,y)$ using values of $\text{score}(z,y)$ for nodes z that are similar to x

Nodes similar to x


$$\text{score}_{unweighted}^*(x, y) := \left| \{z : z \in \Gamma(y) \cap S_x^{\langle \delta \rangle}\} \right|$$
$$\text{score}_{weighted}^*(x, y) := \sum_{z \in \Gamma(y) \cap S_x^{\langle \delta \rangle}} \text{score}(x, z).$$

3. Clustering

- Delete weak edges, and recompute $\text{score}(x,y)$

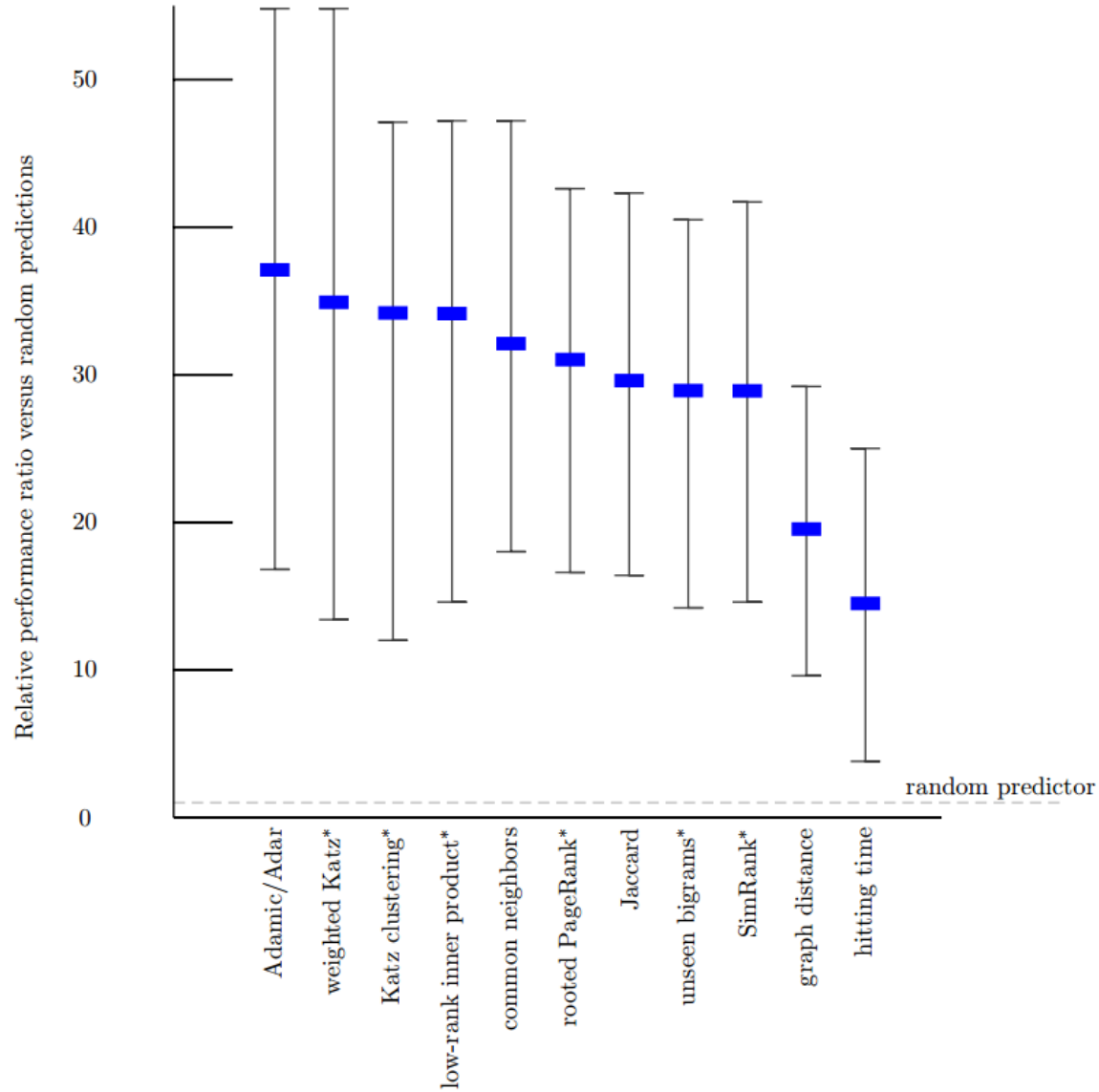


Outline

- Problem Definition
- Methods based on Node Similarity
-  **Results based on Node Similarity**

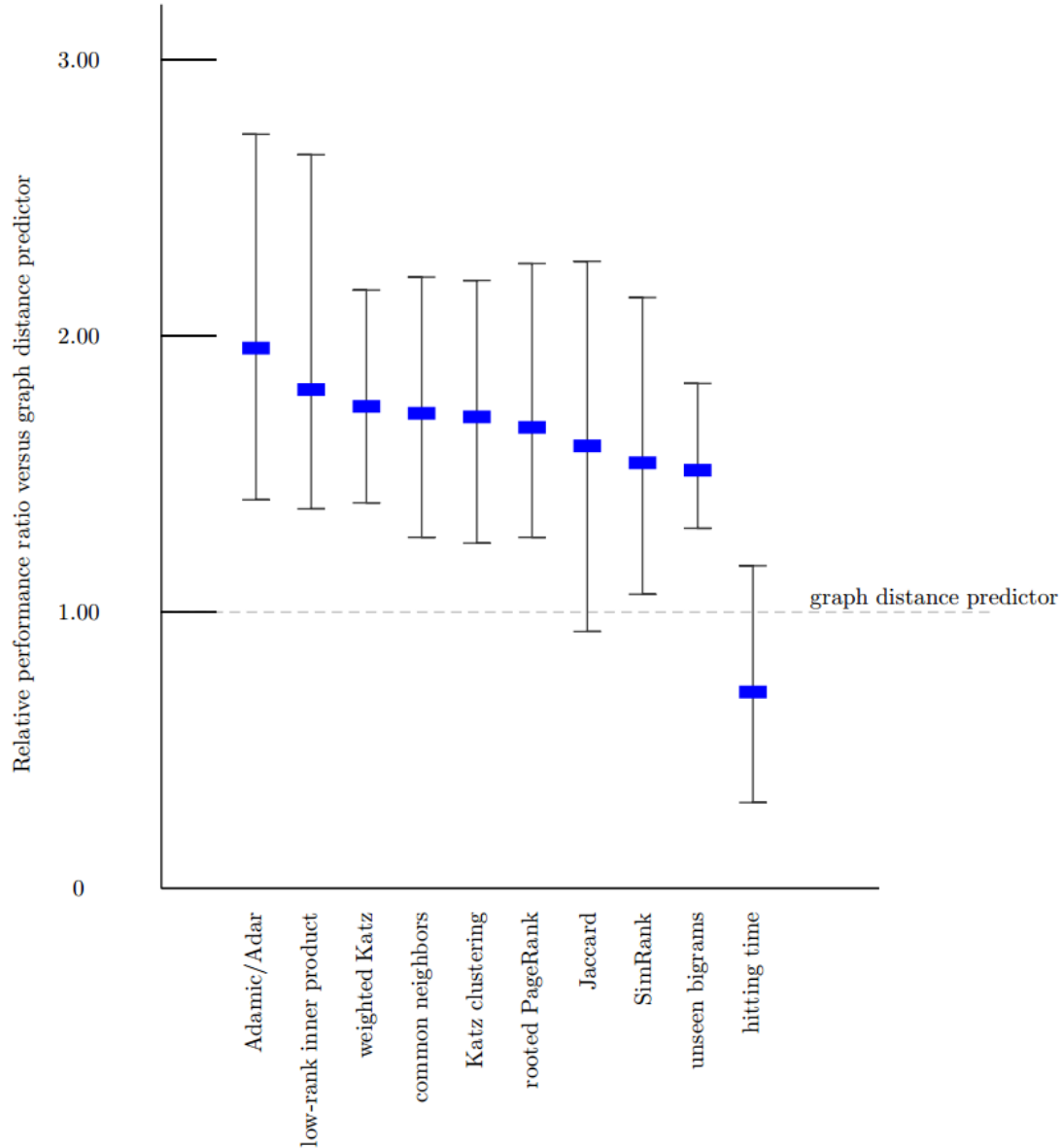


Performance



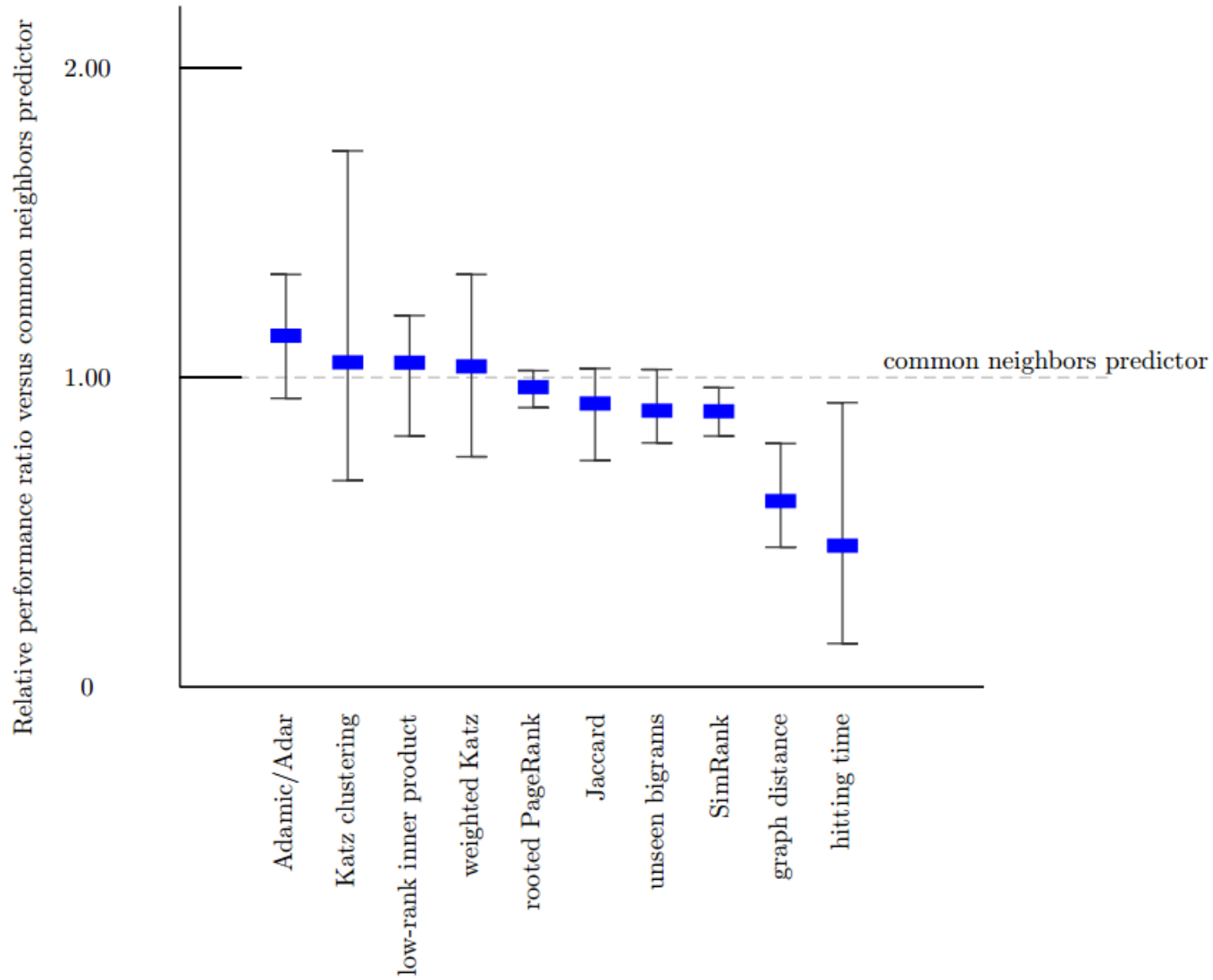


Performance





Performance





Observation #1:

Similarities among Predictors

- Adamic/Adar \sim Katz \sim Low rank inner product
- Jaccard \sim rooted PageRank \sim SimRank

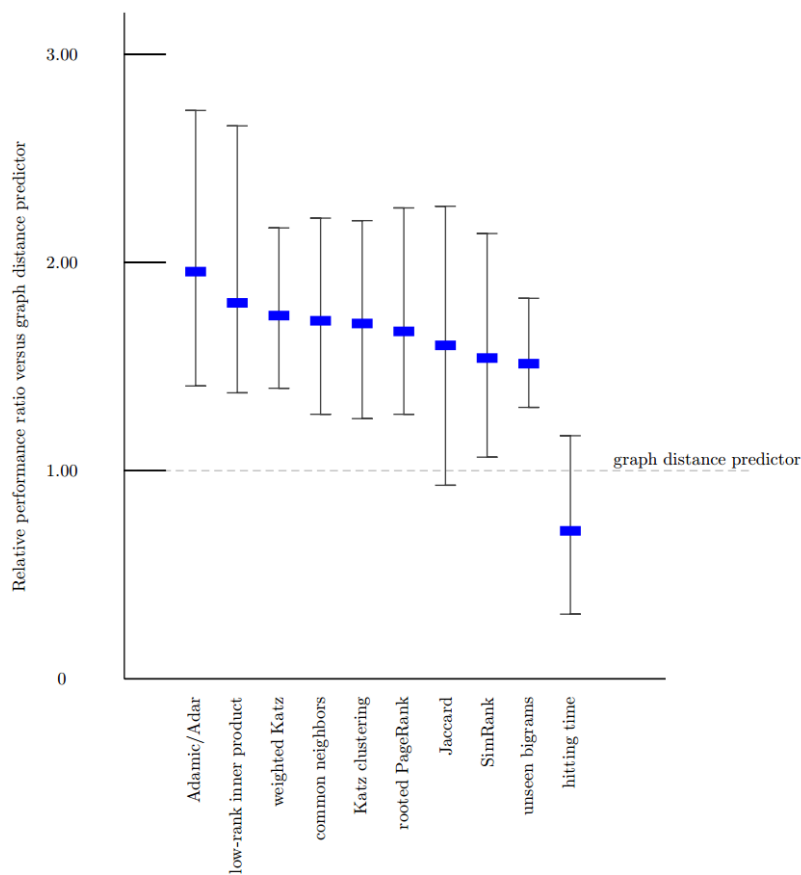
	Adamic/Adar	Katz clustering	common neighbors	hitting time	Jaccard's coefficient	weighted Katz	low-rank inner product	rooted Pagerank	SimRank	unseen bigrams
Adamic/Adar	1150	638	520	193	442	1011	905	528	372	486
Katz clustering		1150	411	182	285	630	623	347	245	389
common neighbors			1150	135	506	494	467	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

Figure 8: The number of common predictions made by various predictors on the cond-mat dataset



Observation #2: Implication of “Small World”

- Graph distance predictor does not work well





Observation #3: Collaborations beyond distance 2

- Many collaborations beyond distance 2
 - 71 % (hep-ph), 83% (cond-mat)

	astro-ph	cond-mat	gr-qc	hep-ph	hep-th
# pairs at distance two	33862	5145	935	37687	7545
# new collaborations at distance two	1533	190	68	945	335
# new collaborations	5751	1150	400	3294	1576

Figure 10: Relationship between new collaborations and graph distance.



Observation #4: Breadth of the data

- As the topical focus of the data set widens, random prediction works poorly

STOC/FOCS	arXiv sections	all arXiv's	Citeseer
6.1	18.0—41.1	71.2	147.0

Performance of common-neighbor vs. random predictor



What You Need to Know

- Link Prediction Problem
 - Infer which new interactions among its members are likely to occur in the near future
- Methods
 - Graph distance, node neighborhood, ensemble of all paths, higher-level approaches (e.g. low rank approx.)
- Results
 - Some predictors work well (Common Neighbor, Adamic/Adar, Katz)
 - Graph distance does not work well (“small world”)



Questions?