Introduction to Web and Data Science

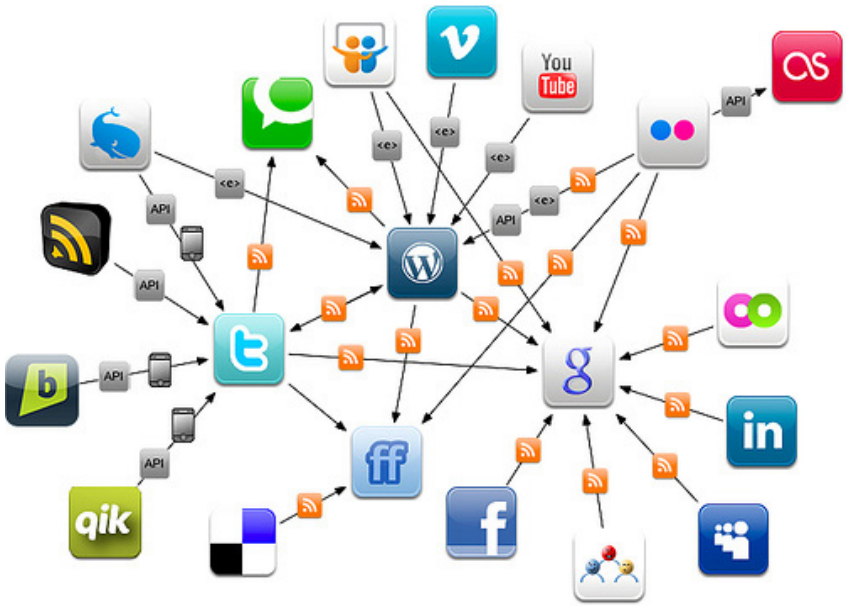
Link Prediction

Prof. Dr. Ralf Möller
Universität zu Lübeck
Institut für Informationssysteme

Acknowledgements

Hong Kong University of Science and Technology
Advanced Data Mining
COMP 4332 / RMBI 4310

Computer Science and Engineering
IIT Kharagpur
Link Prediction in Social Networks
Pabitra Mitra



University of Southern California
CS 599: Social Media Analysis
Social Ties and Link Prediction
Kristina Lerman

A Theoretical Justification of Link Prediction Heuristics
Deepayan Chakrabarti, Purnamrita Sarkar, Andrew Moore

Stanford University
Graph Representation Learning
Jure Leskovec

Applications of Link Prediction on Graphs

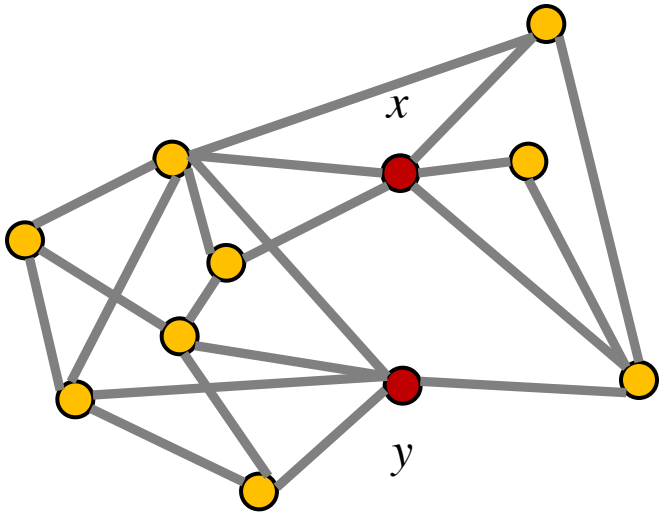
- Who are/will become friends?
- Who will collaborate in drug racketeering?
- Which products to recommend to which persons?
- Are there unknown commonalities between species?
- Where will new protein interactions show up?

Informal Definitions

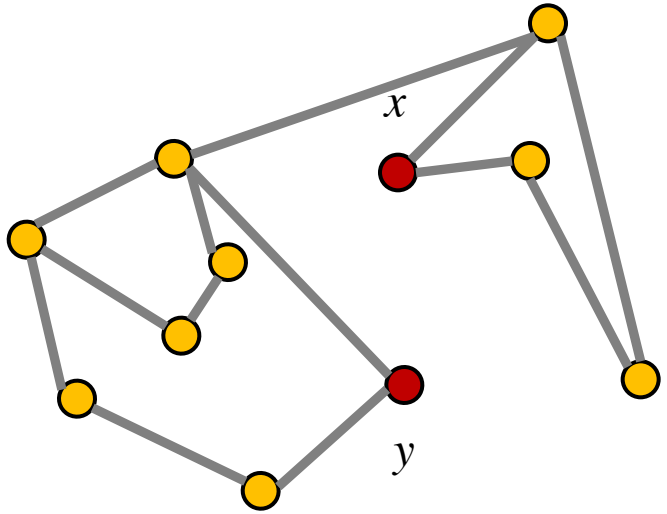
- Link Prediction Problem
  - Given a snapshot of a network, can we infer which new interactions among its nodes are likely to occur in the near future?
- Link Completion Problem
  - If the network is known to be incomplete, can we infer which interactions are possibly missing (and should be added)?
  - Then, solve link prediction problem on completed data
- Both problems to be formalized based on "proximity" of nodes in a network

The Intuition

- In many networks, people who are “close” belong to the same social circles and will inevitably encounter one another and become linked themselves.
- Link prediction heuristics measure how “close” people are



Red nodes are close to each other

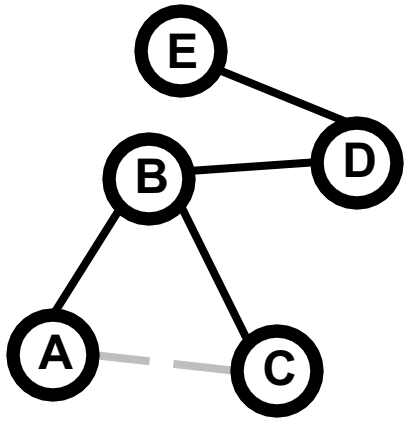


Red nodes are more distant

Challenges

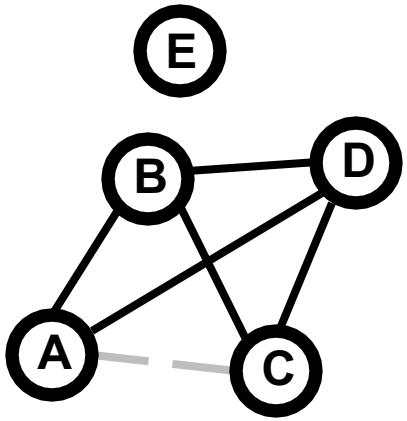
- Data is usually sparse
  - Missing data/relationships
- Imbalance
  - So many possibilities, so few choices
  - Ill-posed problem
  - Low accuracy in practice
- Accuracy vs. scalability
  - Modeling (unobserved/unknown factors)
  - Tasks of approximation/optimization

Graph distance & Common Neighbors



- Graph distance: (Negated) length of shortest path between x and y

(A, C)	-2
(C, D)	-2
(A, E)	-3

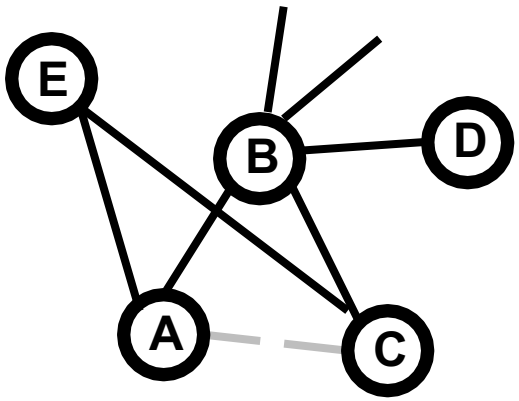


- Common Neighbors: A and C have 2 common neighbors, more likely to collaborate

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

where $\Gamma(x)$ denotes the neighbors of x

Jaccard's coefficient and Adamic / Adar



- Jaccard's coefficient: same as common neighbors, adjusted for degree

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

- Adamic / Adar: weighting rarer neighbors more heavily

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

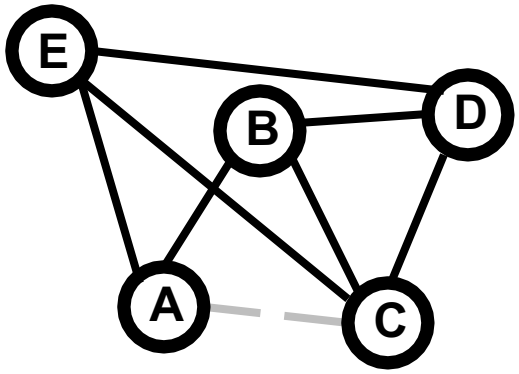
Adamic, Lada A; Adar, Eytan. "Friends and neighbors on the web". *Social Networks*. Elsevier. **25** (3): 211–230. **2003**.

Preferential Attachment

- Preferential Attachment: Probability that a new collaboration involves x is proportional to $\Gamma(x)$, the current neighbors of x

- score (x, y) := $|\Gamma(x)| \cdot |\Gamma(y)|$

Considering all Paths: Katz



- Katz: Measure that sums over the collection of paths, exponentially damped by length (to count short paths heavily)

$\sum_{\ell=1}^{\infty} \beta^{\ell} \cdot |\mathsf{paths}_{x,y}^{\langle \ell \rangle}|$ β is chosen to be a very small value (for dampening)

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

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

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

Katz, L. A New Status Index Derived from Sociometric Analysis. Psychometrika, 39–43. **1953**.

Hitting time, PageRank

- Hitting time: expected number of steps for a random walk starting at x to reach y
- Commute time: $-(H_{x,y} + H_{y,x})$

- If y has a large stationary probability, $H_{x,y}$ is small. To counterbalance, we can normalize

$$\text{score}(x, y) := -(H_{x,y} \cdot \pi_y + H_{y,x} \cdot \pi_x)$$

- Rooted PageRank: to cut down on long random walks, walk can return to x with a probablity α at every step y

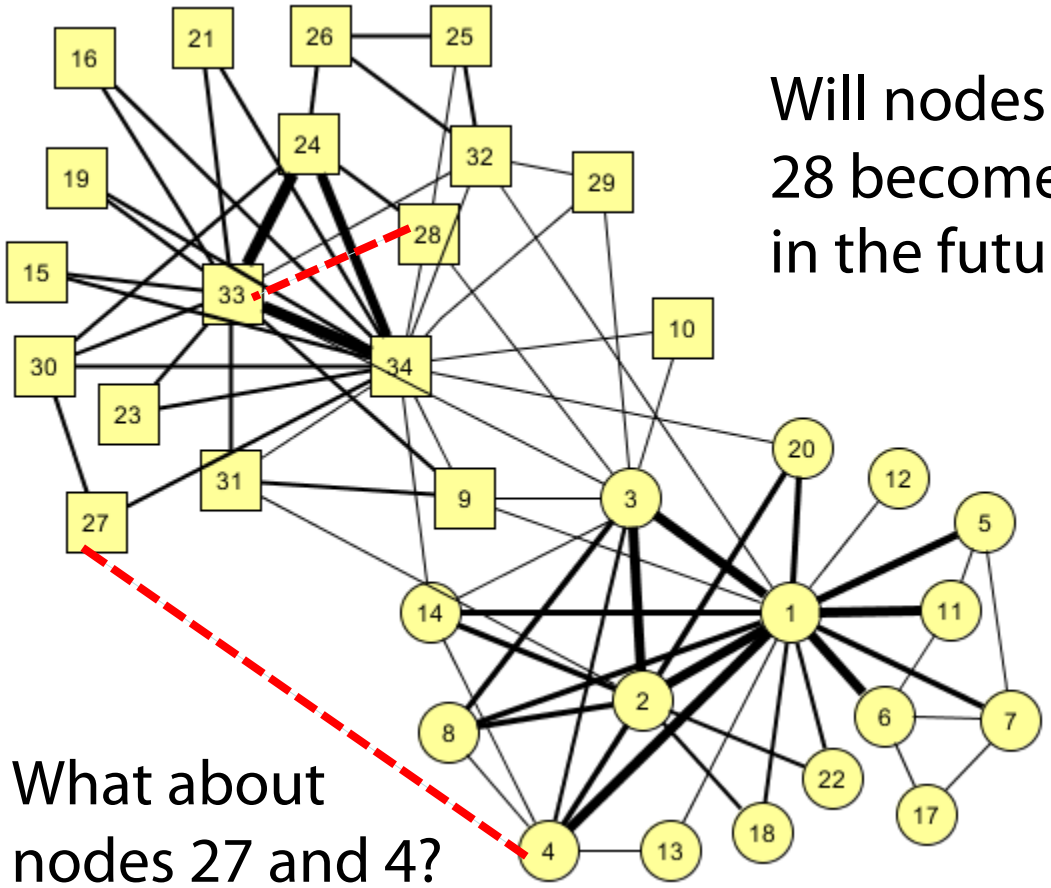
SimRank

Defined by this recursive definition: two nodes are similar to the extent that they are joined by similar neighbors

$$\mathsf{similarity}(x, y) := \gamma \cdot \frac{\sum_{a \in \Gamma(x)} \sum_{b \in \Gamma(y)} \mathsf{similarity}(a, b)}{|\Gamma(x)| \cdot |\Gamma(y)|}$$

$$\mathsf{score}(x, y) := \mathsf{similarity}(x, y)$$

Link Prediction



Will nodes 33 and 28 become friends in the future?

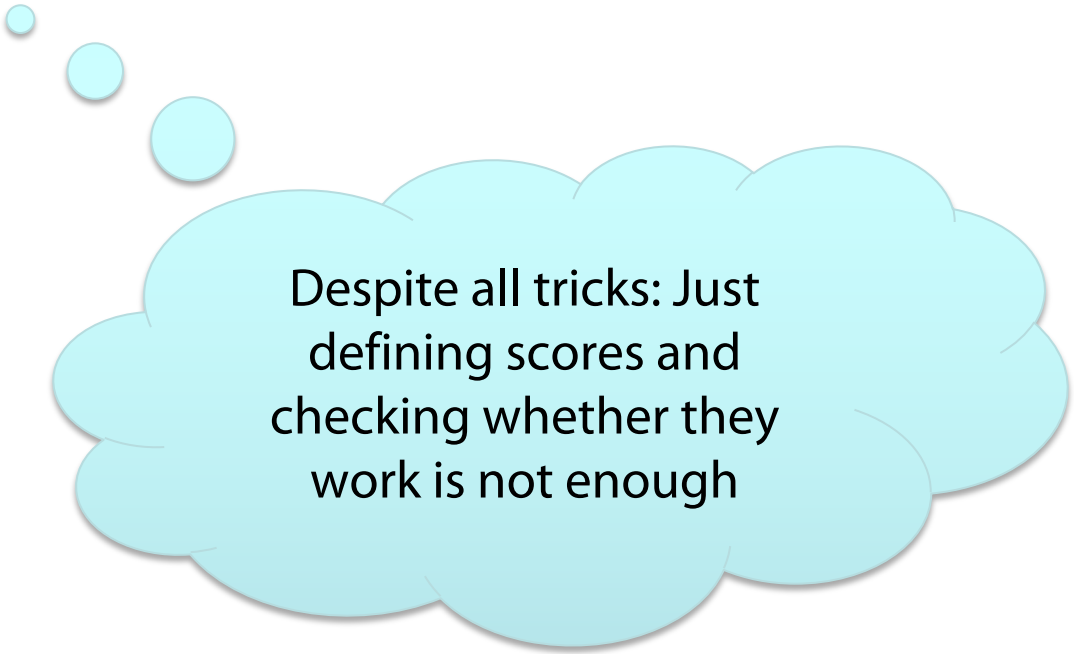
Does network structure contain enough information to predict what new links will form in the future?

What about nodes 27 and 4?

Clustering

Idea:

- Delete tenuous (sparse) edges with a clustering procedure
- Run predictors on the “cleaned-up” subgraphs

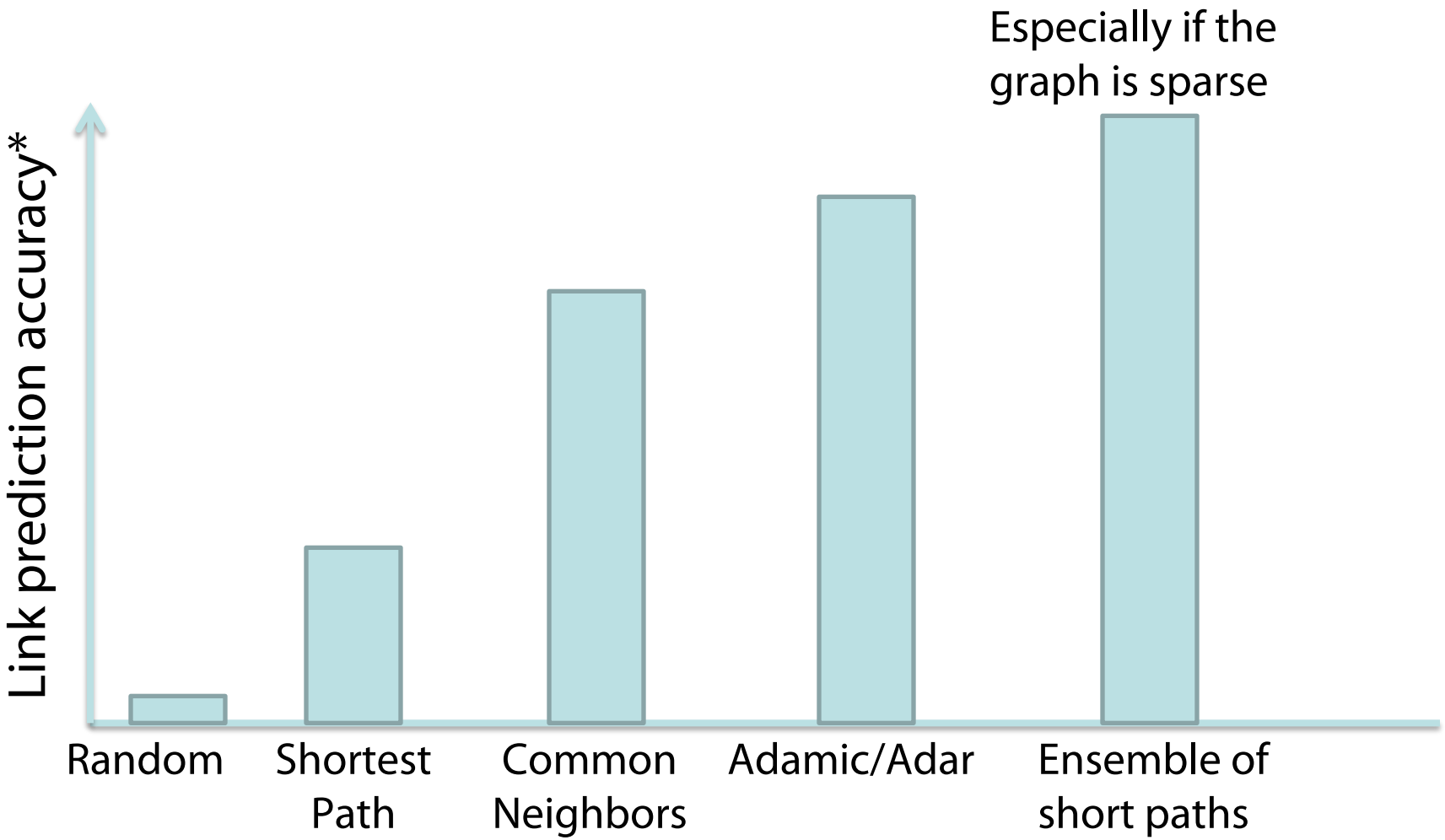


Empirical Results

- No single clear winner
- Many methods outperform the random predictor
  => there is useful information in the network topology
- Katz + clustering + low-rank approximation* perform significantly well
- Some simple measures, i.e., common neighbors and Adamic/ Adar perform quite well

* To be explained in later semesters

Previous Empirical Studies*

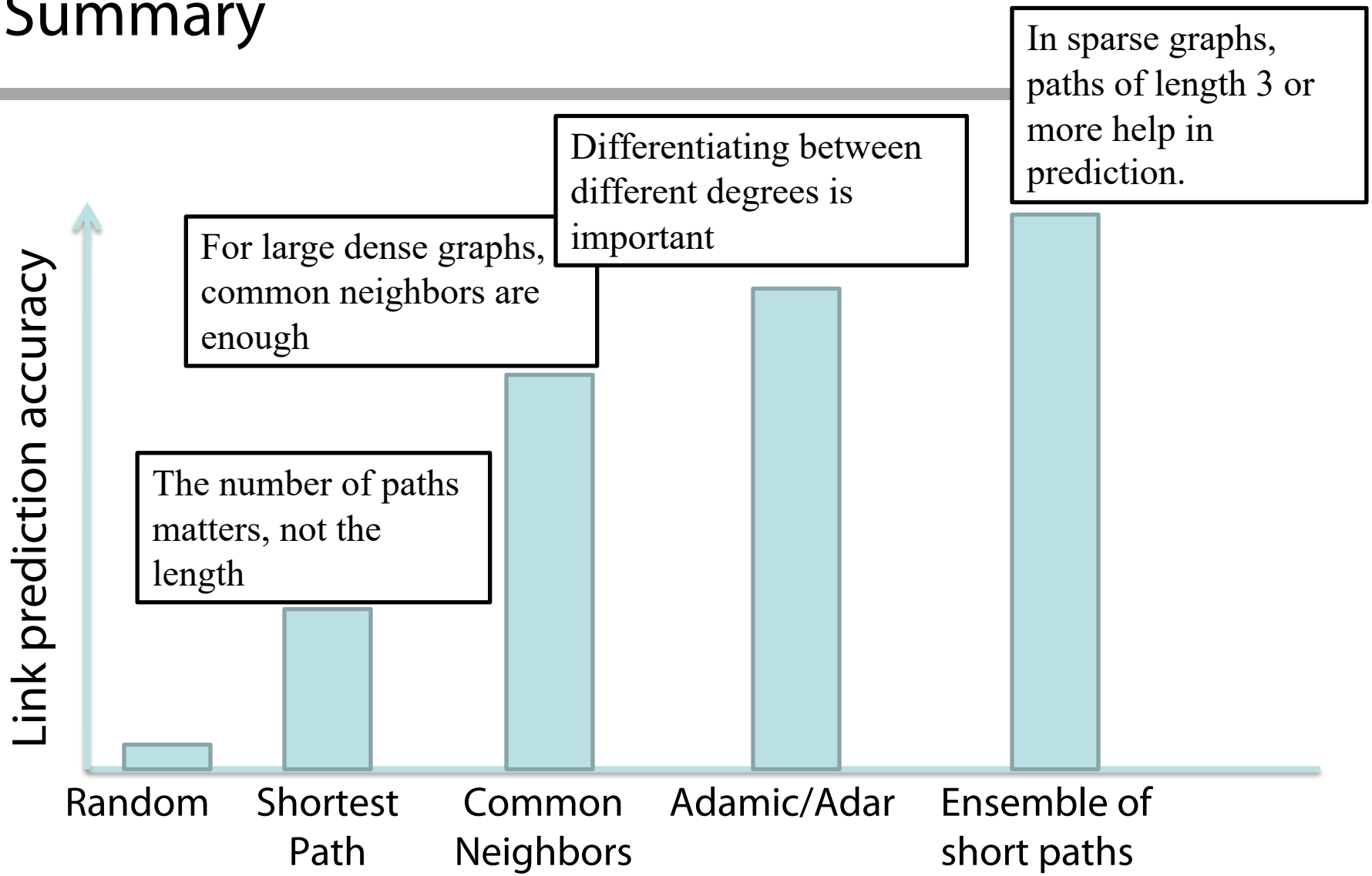


*Liben-Nowell & Kleinberg, 2003; Brand, 2005; Sarkar & Moore, 2007

Critique

- Even the best predictor (Katz) is correct on only 16% of predictions
- How good is that?
- Maybe more information about the meaning of nodes and edged is required

Summary



Purnamrita Sarkar, Deepayan Chakrabarti, and Andrew W. Moore.
Theoretical justification of popular link prediction heuristics.
In: Proc. IJCAI-11. pp. 2722–2727. **2011**.

Link Prediction using Collaborative Filtering

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1 	8	1	?	2	7
User 2 	2	?	5	7	5
User 3 	5	4	7	4	7
User 4 	7	1	7	3	8
User 5 	1	7	4	6	?
User 6 	8	3	8	3	7

Link Prediction using Collaborative Filtering

- Memory-based Approach
  - User-based approach [Twitter]
  - Item-based approach [Amazon & Youtube]
- Model-based Approach
  - Latent Factor Model [Google News]

- Hybrid Approach

Memory-based Approach

- Few modeling assumptions
- Few tuning parameters to learn
- Easy to explain to users
  - Dear Amazon.com Customer, We've noticed that customers who have purchased or rated *How Does the Show Go On: An Introduction to the Theater* by Thomas Schumacher have also purchased *Princess Protection Program #1: A Royal Makeover (Disney Early Readers)*.

Algorithms: User-Based Algorithms (Breese et al, UAI98)

- $v_{i,j}$= vote of user *i* on item *j*
- I_i = items for which user *i* has voted
- Mean vote for *i* is

$$\overline{v}_i = \frac{1}{|I_i|} \sum_{j \in I_i} v_{i,j}$$

- Predicted vote for "active user" *a* is weighted sum

$$p_{a,j} = \overline{v}_a + \kappa \sum_{i=1}^{n} w(a,i)(v_{i,j} - \overline{v}_i)$$

normalizer

weights of *n* similar users

Algorithms: User-Based Algorithms (Breese et al, UAI98)

- K-nearest neighbor

$$w(a,i) = \begin{cases} 1 & \text{if } i \in \text{neighbors}(a) \\ 0 & \text{else} \end{cases}$$

- Pearson correlation coefficient (Resnick ' 94, Grouplens):

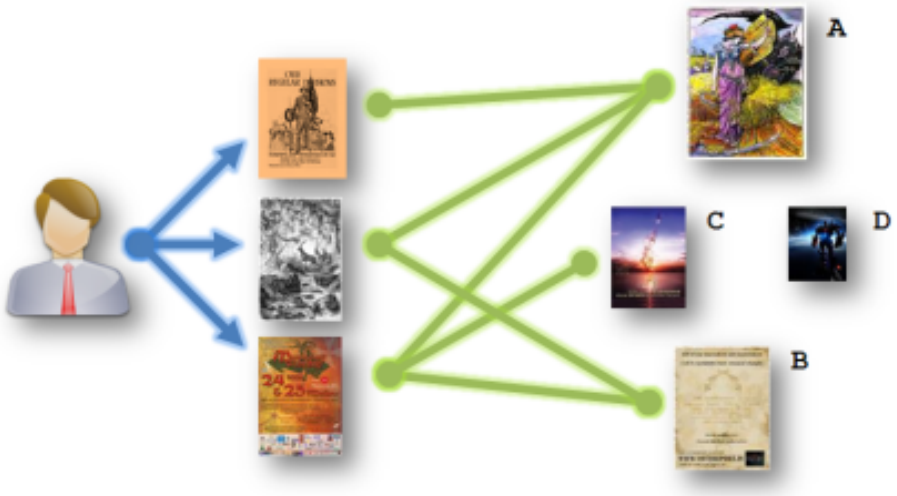
$$w(a,i) = \frac{\sum_j (v_{a,j} - \overline{v}_a)(v_{i,j} - \overline{v}_i)}{\sqrt{\sum_j (v_{a,j} - \overline{v}_a)^2 \sum_j (v_{i,j} - \overline{v}_i)^2}}$$

- Cosine distance (from IR)

$$w(a,i) = \sum_j \frac{v_{a,j}}{\sqrt{\sum_{k \in I_a} v_{a,k}^2}} \frac{v_{i,j}}{\sqrt{\sum_{k \in I_i} v_{i,k}^2}}$$

Algorithm: Amazon's Method

- Item-based Approach
  - Similar with user-based approach but is on the item side



Item-based CF Example: infer (user 1, item 3)

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1 	8	1	?	2	7
User 2 	2	?	5	7	5
User 3 	5	4	7	4	7
User 4 	7	1	7	3	8
User 5 	1	7	4	6	?
User 6 	8	3	8	3	7

How to Calculate Similarity (Item 3 and Item 5)?

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1 	8	1	?	2	7
User 2 	2	?	5	7	5
User 3 	5	4	7	4	7
User 4 	7	1	7	3	8
User 5 	1	7	4	6	?
User 6 	8	3	8	3	7

Similarity between Items

Item 3	Item 4	Item 5
?	2	7
5	7	5
7	4	7
7	3	8
4	6	?
8	3	7

- How similar are items 3 and 5?
  - How to calculate their similarity?

Similarity between items

Item 3	Item 5
?	7
5	5
7	7
7	8
4	?
8	7

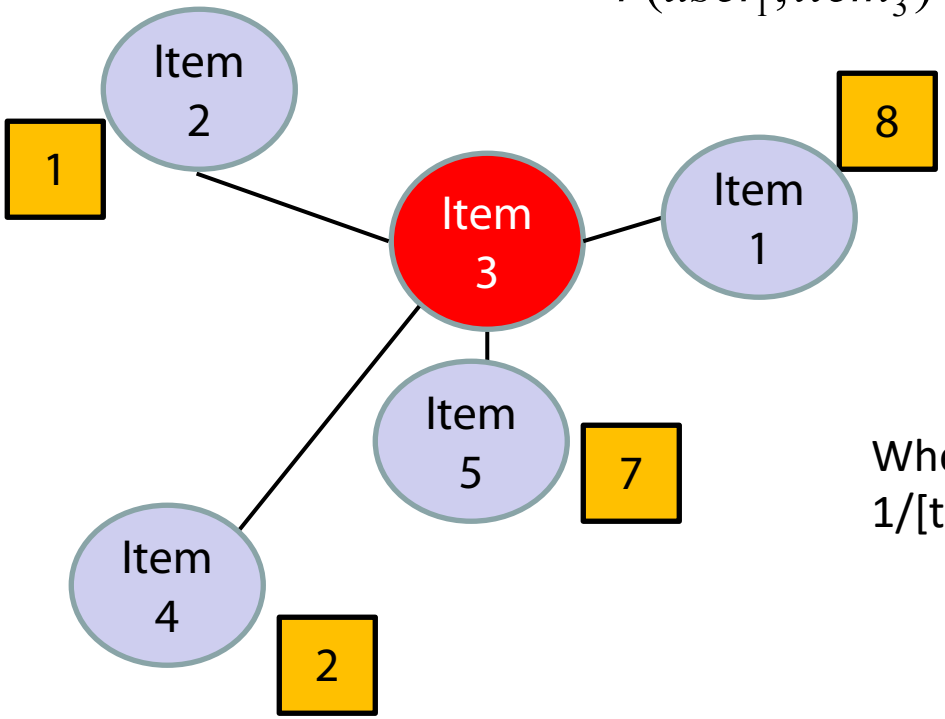
- Only consider users who have rated both items
- For each user:
  Calculate difference in ratings for the two items
- Take the average of this difference over the users

sim(item 3, item 5) = cosine((5, 7, 7), (5, 7, 8))

$= (5*5 + 7*7 + 7*8)/(\text{sqrt}(5^2+7^2+7^2)* \text{sqrt}(5^2+7^2+8^2))$

- Can also use Pearson Correlation Coefficients as in user-based approaches

Prediction: Calculating ranking r(user1,item3)



$$r(user_1, item_3) = \alpha * \{r(user_1, item_1) sim(item_1, item_3) + r(user_1, item_2) sim(item_2, item_3) + r(user_1, item_4) sim(item_4, item_3) + r(user_1, item_5) sim(item_5, item_3)\}$$

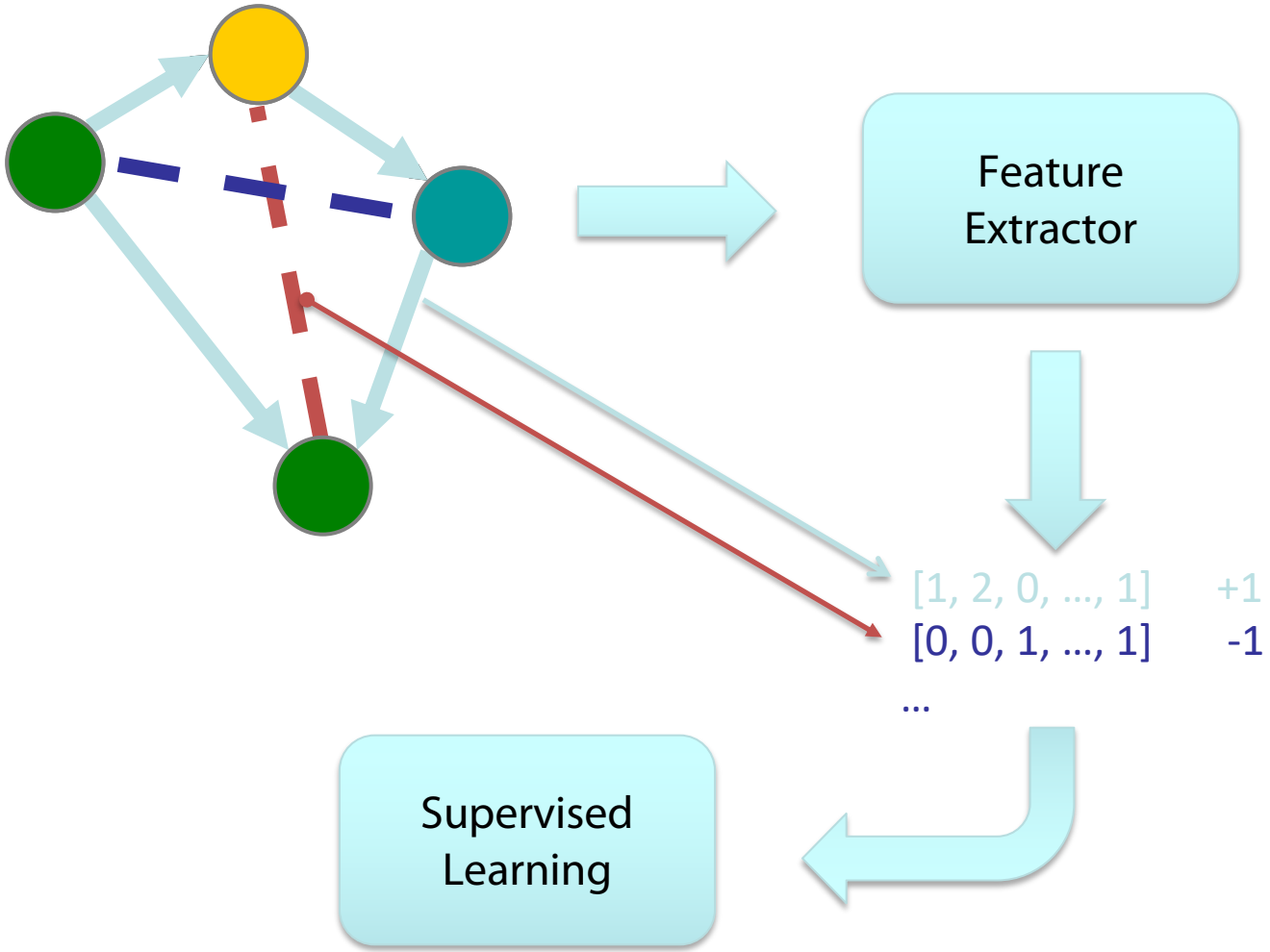
Where α is a normalization factor, which is 1/[the sum of all sim($item_i$,$item_3$)].

Algorithm: Youtube's Method

- Youtube also adopt item-based approach

- Adding more useful features
  - Num. of views
  - Num. of likes
  - etc.

Link Prediction using Supervised Learning Methods



David Liben-Nowell and Jon Kleinberg. The link prediction problem for social networks. In Proceedings of the twelfth international conference on Information and knowledge management (CIKM '03), 556–559. **2003**.

The Fundamental Challenge

How to estimate as much signal as possible where there are sufficient data, without over fitting where data are scarce?

Node2vec: Unsupervised Feature Learning

- Intuition: Find embedding of nodes to d-dimensions that preserves similarity
- Idea: Learn node embedding such that nearby nodes are close together
- How do we define nearby nodes?

A. Grover, J. Leskovec. node2vec: Scalable Feature Learning for Networks. In Proc. KDD **2016**.

Feature Learning as Optimization

- Given $G = (V, E)$
- Goal: Learn $f: u \rightarrow \mathbb{R}^d$ (coordinates of u)
- Find representation $f(u)$ of u that is predictive neiborhood nodes $N_S(u)$

Unsupervised Feature Learning

- Find embedding $f(u)$ that predicts nearby nodes $N_S(u)$

$$\arg\max_f \sum_{u \in V} \log \Pr(N_S(u) | f(u))$$

- Assume that conditional likelihood factorizes

$$Pr(N_S(u)|f(u)) = \prod_{n_i \in N_S(u)} Pr(n_i|f(u))$$

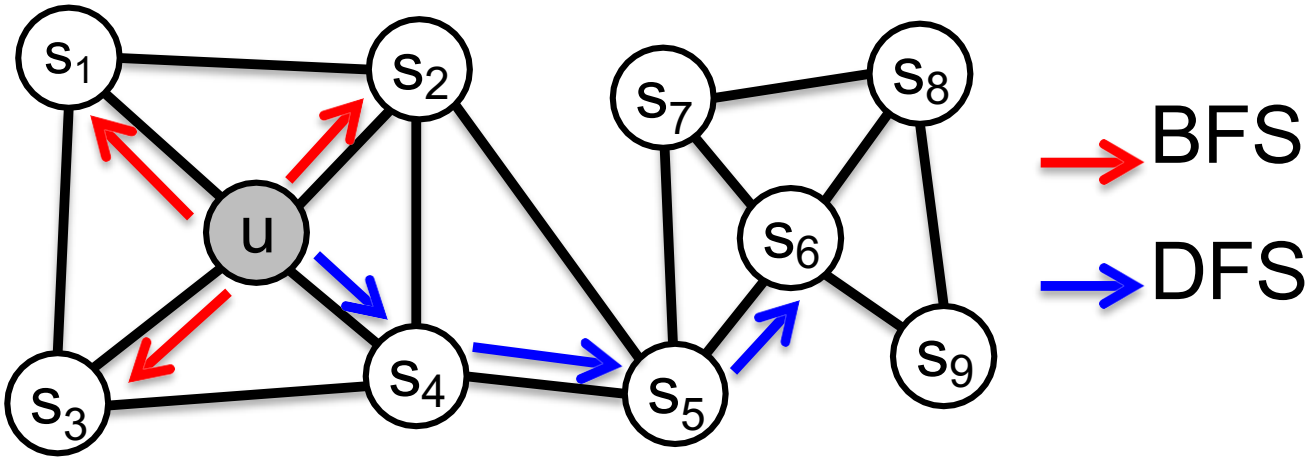
- Then softmax

$$Pr(n_i|f(u)) = \frac{\exp(f(n_i) \cdot f(u))}{\sum_{v \in V} \exp(f(v) \cdot f(u))}$$

- Estimate $f(u)$ using gradient descent

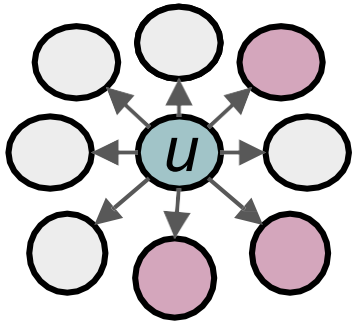
How to Determine $N_S(u)$?

- Strategies S: Breadth-First or Depth-First



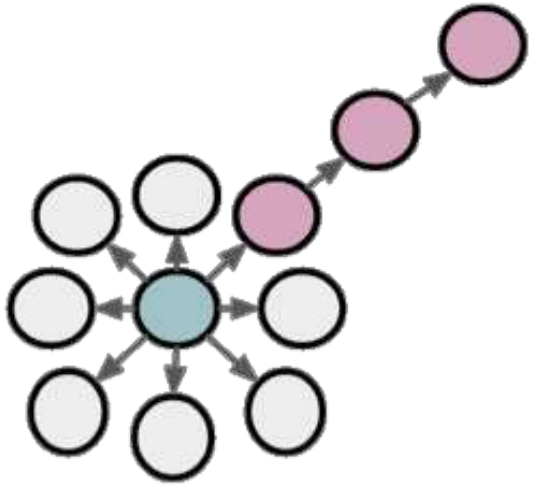
- $N_{BFS}(u) = \{s_1, s_2, s_3\}$ local view
- $N_{DFS}(u) = \{s_4, s_5, s_6\}$ global view

BFS vs. DFS



BFS:

Micro-view of neighbourhood



DFS:

Macro-view of neighbourhood

Interpolating BFS and DFS

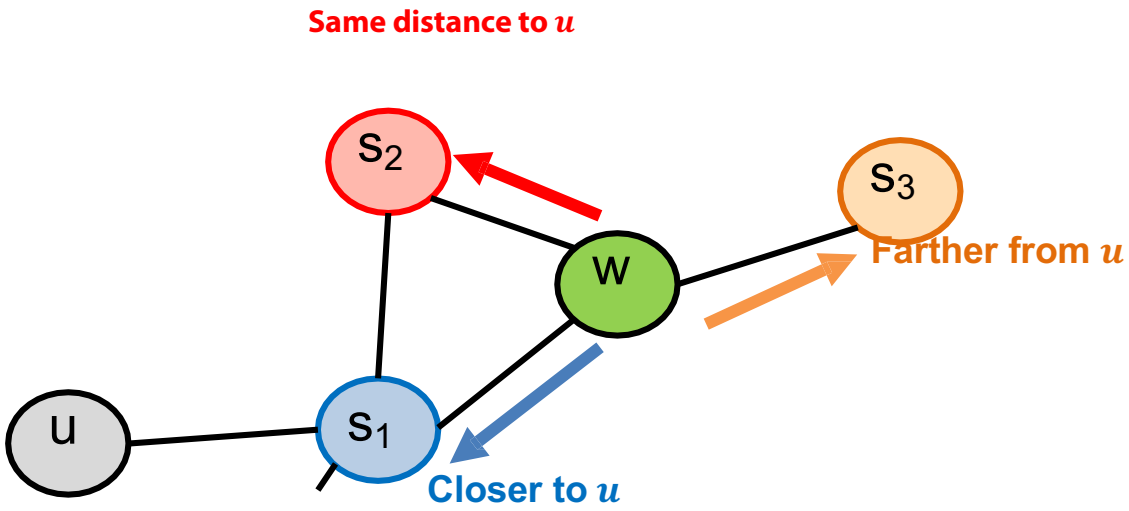
Biased random walk S that given a node u generates neighborhood $N_S(u)$

- Two parameters:
  - Return parameter p:
    - Return back to the previous node
  - In-out parameter q:
    - Moving outwards (DFS) vs. inwards (BFS)
    - Intuitively, q is the "ratio" of BFS vs. DFS

Biased Random Walks

Biased 2nd-order random walks explore network neighborhoods:

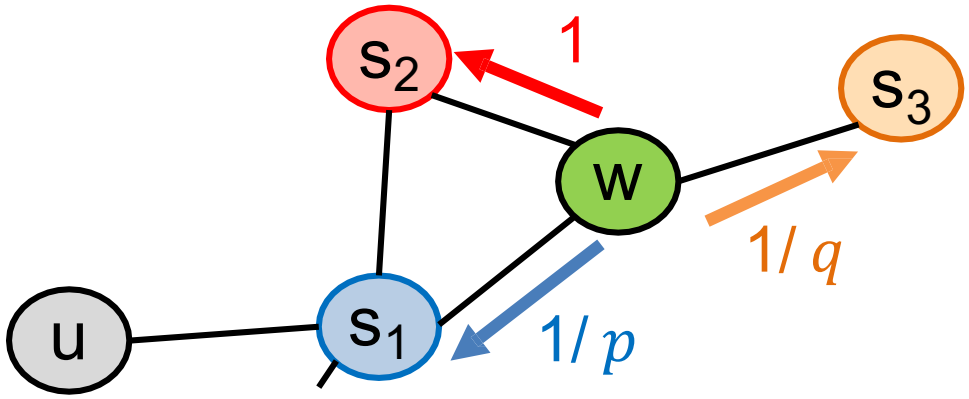
- Rnd. walk started at u and is now at w
- **Insight:** Neighbors of w can only be:



Idea: Remember where that walk came from

Biased Random Walks

- Walker is at w. Where to go next?

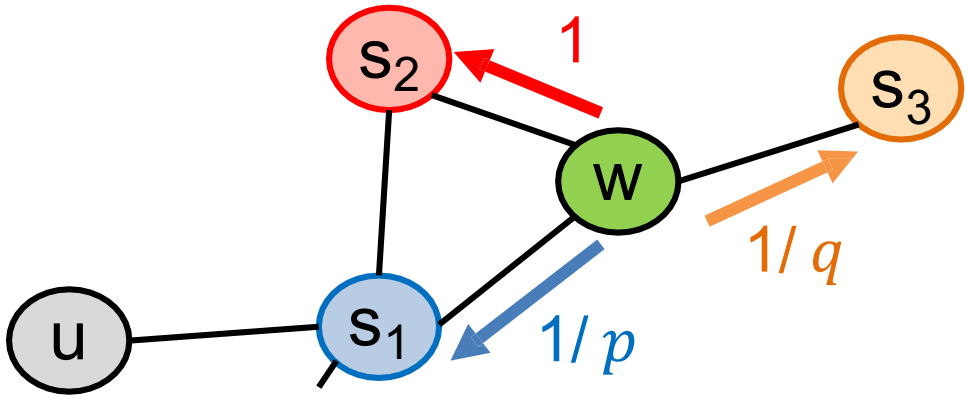


$1/p, 1/q, 1$ are unnormalized probabilities

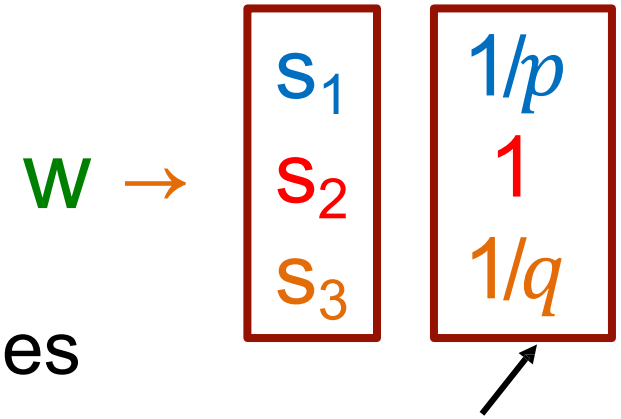
- p, q: model transition probabilities
  - p: return parameter
  - q: "walk away" parameter

Biased Random Walks

- Walker is at w. Where to go next?



$1/p, 1/q, 1$ are unnormalized probabilities



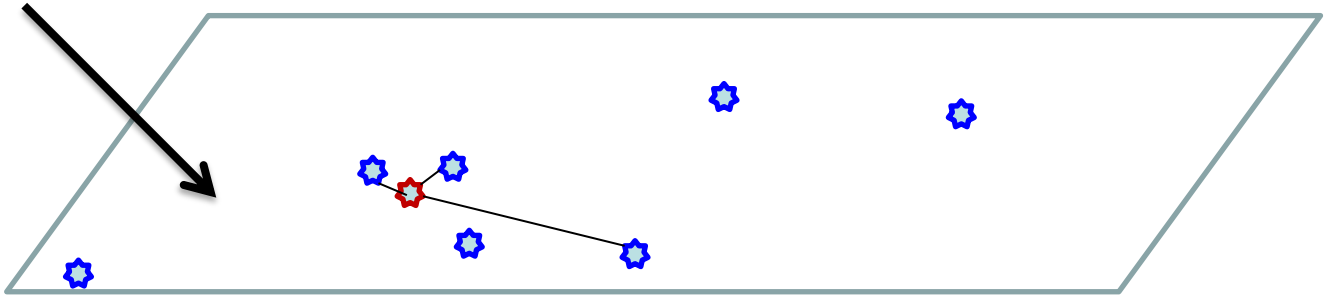
- p, q: model transition probabilities
  - p: return parameter (low = BFS)
  - q: "walk away" parameter (low = DFS)
- $N_S(u)$ are the nodes visited by the walker

Node2Vec Algorithm

- Simulate r random walk of length l starting from each node u
- Optimize the node2vec objective using Stochastic Gradient Descent

Link Prediction – Generative Model

Unit volume universe



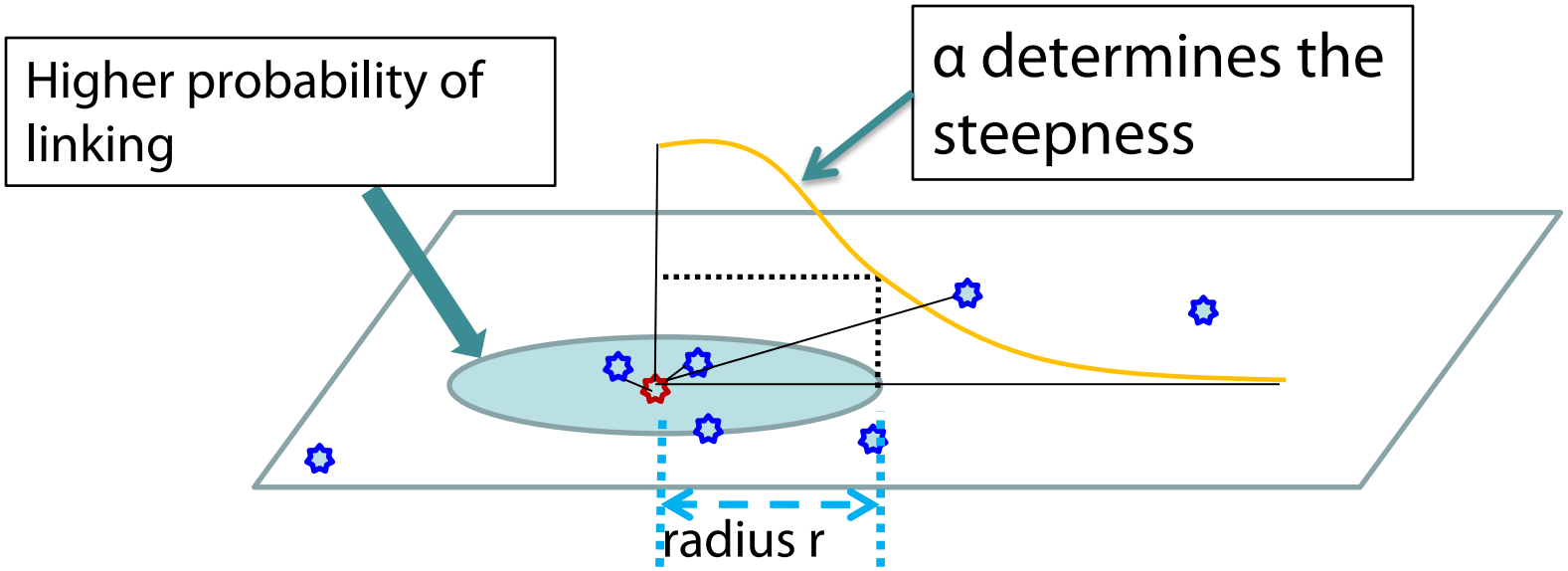
Model:

1. Nodes are uniformly distributed points in a latent space
2. This space has a distance metric
3. Points close to each other are likely to be connected in the graph

➢ Logistic distance function

Hoff PD, Raftery AE, Handcock MS. "Latent Space Approaches to Social Network Analysis." Journal of the American Statistical Association, 97(460), 1090–1098. **2002**.

Link Prediction – Generative Model



Model:

1. Nodes are uniformly distributed points in a latent space
2. This space has a distance metric
3. Points close to each other are likely to be connected in the graph

Link prediction ≈ find nearest neighbor who is not currently linked to the node
Equivalent to inferring distances in the latent space

Link Prediction: Summary

- Link prediction is the underlying problem in many applications
- No methods fits all purposes
- Need to carefully evaluate a method in a practical setting
- Methods are hard to analyze theoretically, but see

Purnamrita Sarkar, Deepayan Chakrabarti, and Andrew W. Moore.
Theoretical justification of popular link prediction heuristics.
In: Proc. IJCAI-11. pp. 2722–2727. **2011**.