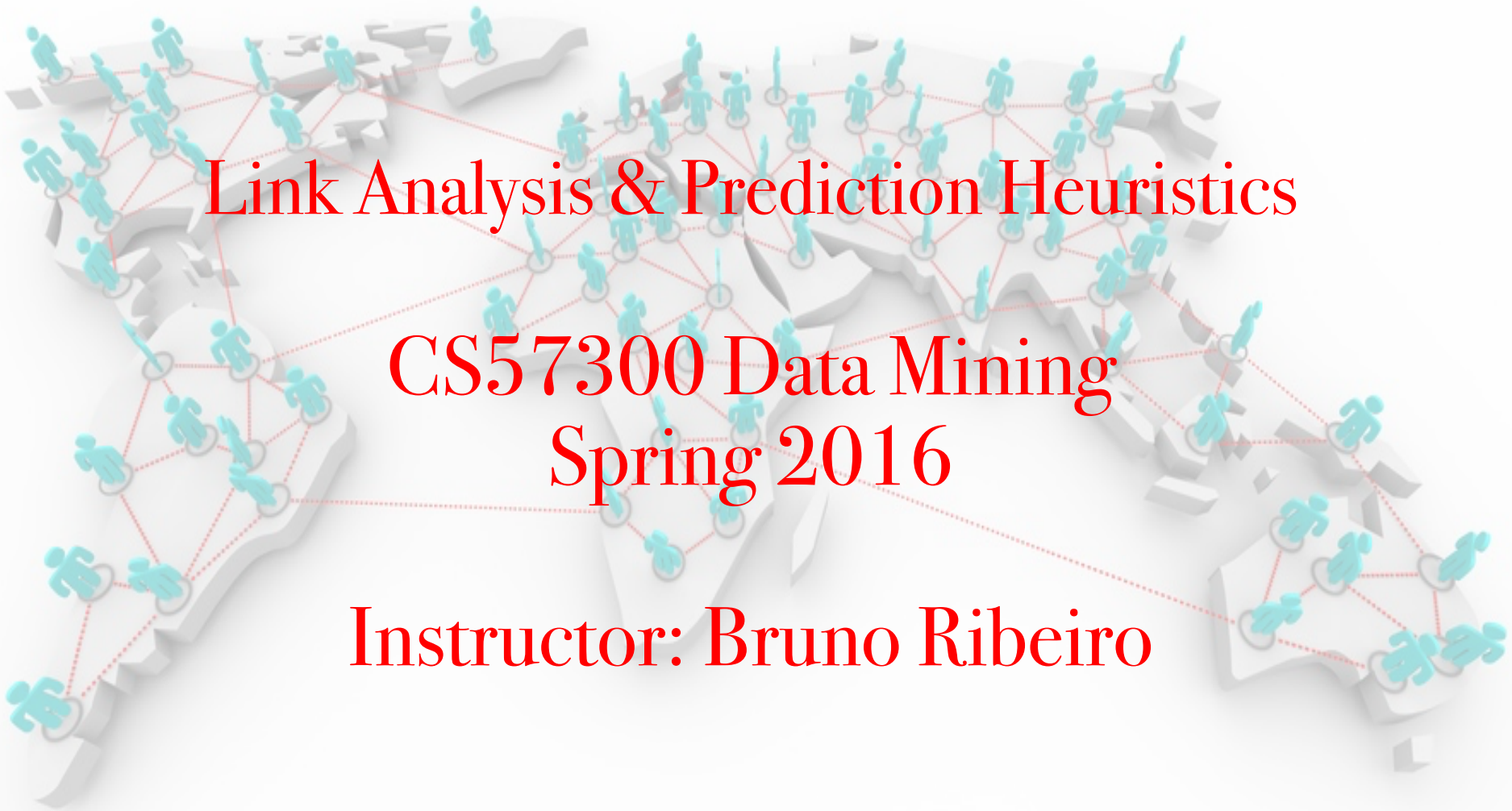


Link Analysis & Prediction Heuristics

CS57300 Data Mining
Spring 2016

Instructor: Bruno Ribeiro

Overview

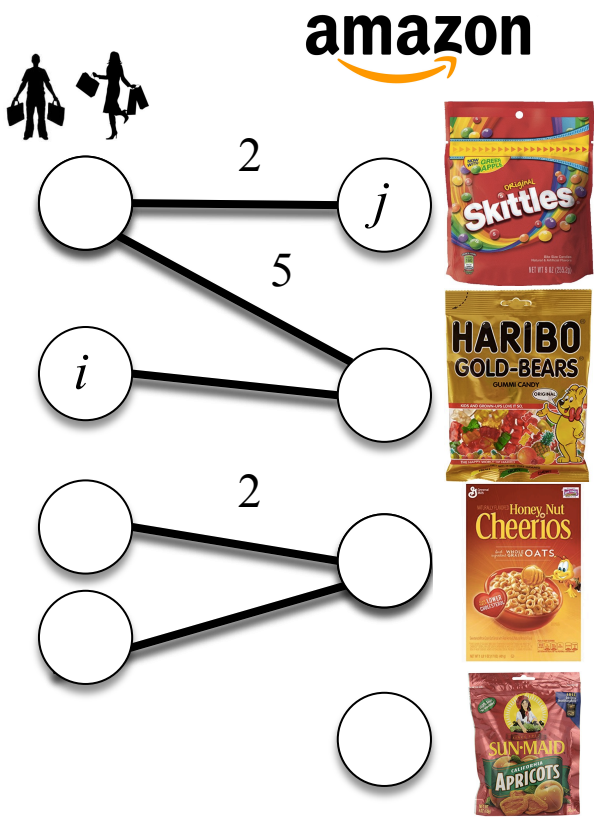
- Deterministic Heuristics Methods
- Matrix / Probabilistic Methods
- Supervised Learning Approaches

Goal

- Input: *Snapshot of social network at time t*
- Ouput: *Predict edges that will be added to network at time t' > t*

Product Recommendation

Example of link prediction application



Twitter's Who to Follow

Another example of link prediction application

Other Applications

A few more examples:

- Fraud Detection: (Beutel, Akoglu, Faloutsos, KDD'15)
  - Focuses on discovering surprising link patterns
- Anomaly detection: (Rattigan et al, 2005)
  - Focuses on finding unlikely existing links

Heuristics

A Very Naïve Approach

- Every entity is assigned a score
- *Score*(*v*) = how many friends person already has

General Approach

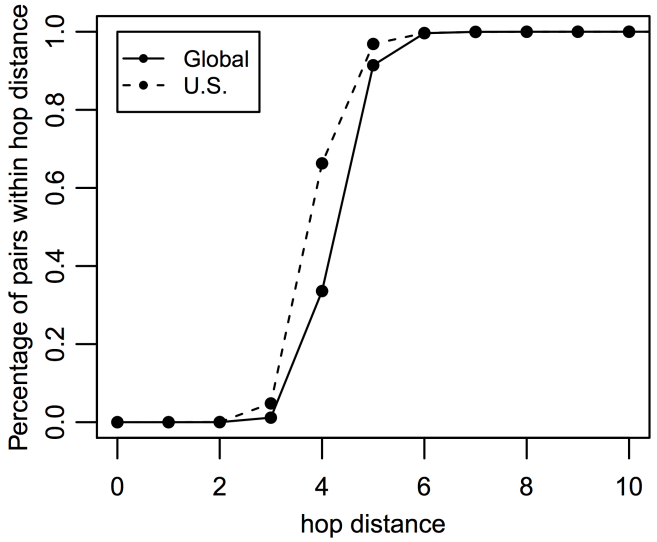
- Assign connection weight *score*(u, v) to non-existing edge (u, v)
- Rank edges and recommend ones with highest *score*
- Can be viewed as computing a measure of proximity or "similarity" between nodes *u* and *v*.

Shortest Path

- Proximity measured by length of shortest path between u and v. Suggests connections between nodes that are nearby

- **Problem**: Network diameter often very small & distribution very concentrated.

Distribution of shortest paths on Facebook [1]:



[1] Ugander, Johan, et al.
"The anatomy of the facebook social grap
arXiv:1111.4503 (2011).

Common Neighbors

Common neighbors uses as score the number of common neighbors between vertices u and v.

Neighbors of u

$$\text{score}(u, v) = |N(u) \cap N(v)|$$

Problem: Large scores for vertices with too many neighbors

Preferential Attachment

$$\text{score}(u, v) = |N(u)| \cdot |N(v)|$$

- The probability of co-authorship of u and v is proportional to the product of the number of collaborators of u and v

Problem: Large scores for vertices with too many neighbors

Jaccard Similarity

$$\text{score}(u, v) = \frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|}$$

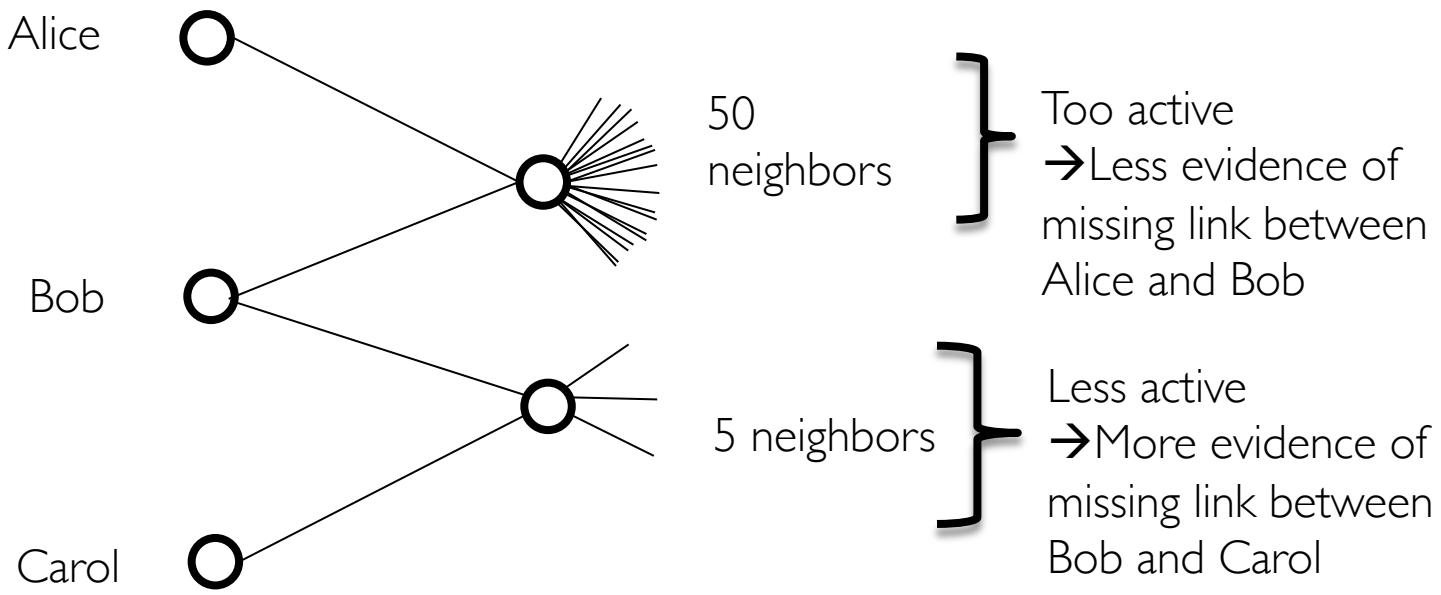
- Fixes *Common Neighbors* "u or v have too many neighbors" problem by dividing the intersection by the union
- Score value between 0 and 1

- **Problem**: These folks are still the problem. Why?

Adamic / Adar

- This score gives more weight to neighbors that that are not shared with many others.

$$\text{score}(u, v) = \sum_{z \in N(u) \cap N(v)} \frac{1}{\log |N(z)|}$$



Katz score (1953)

Binary symmetric adjacency matrix $A = \begin{bmatrix} & v & & u & \\ & 0 & 1 & 0 & \dots \\ & 1 & 0 & 1 & \dots \\ & 0 & 1 & 0 & \dots \\ & & & \vdots & \end{bmatrix}$

$$\text{score}(u, v) = \sum_{l=1}^{\infty} \alpha^l (A^l)_{u,v}$$

- Exponentially weighted sum of number of paths of length l
- For $\alpha \ll 1$: predictions ≈ common neighbors
- α is such that $\rho(\alpha A) < 1$, where $\rho(\alpha A)$ is the spectral radius of αA

Hitting Time

$$\text{score}(u, v) = H_{u,v}$$

- where, $H_{u,v}$ is the random walk hitting time between u and v

Personalized PageRank

$$\text{score}(u, v) = \pi_v^{(u)}$$

- Stationary probability walker is at v under the following random walk:
  - With probability α, jump back to u
  - With probability 1 – α, go to random neighbor of current node

SimRank

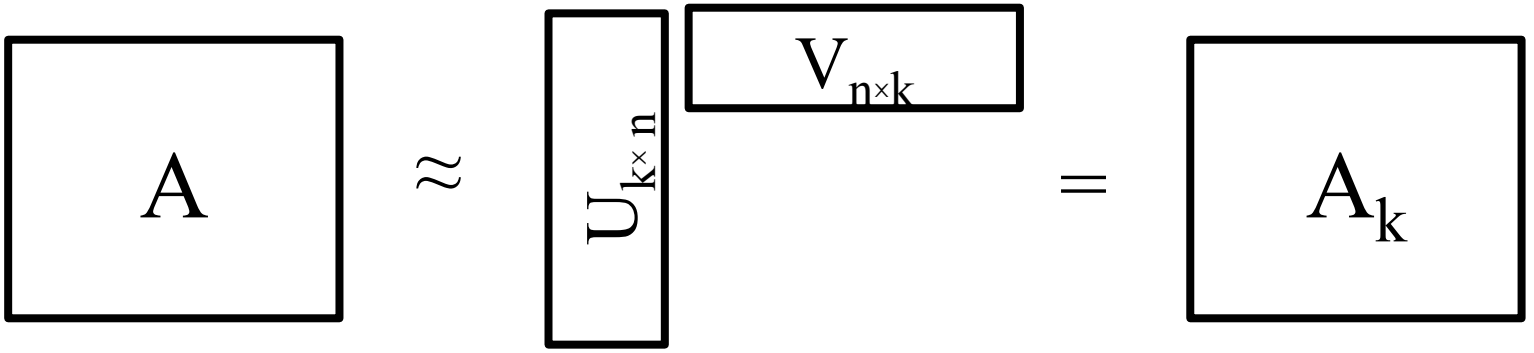
$$\text{score}(u,v) = \frac{C}{|N(u)| \cdot |N(v)|} \sum_{i=1}^{|N(u)|} \sum_{j=1}^{|N(v)|} \text{score}(N_i(u), N_j(v))$$

- N(u) and N(v) are number of in-degrees of nodes u and v
- Only directed graphs
- score(u,v) ∈ [0,1]. If u=v then, **score(u,v)=1**

Latent Space Models

Low Rank Reconstruction

- Represent the adjacency matrix A with a lower rank matrix A_k.



- If $A_k(u,v)$ has large value for a missing $A(u,v)=0$, then recommend link (u,v)

Product Recommendations

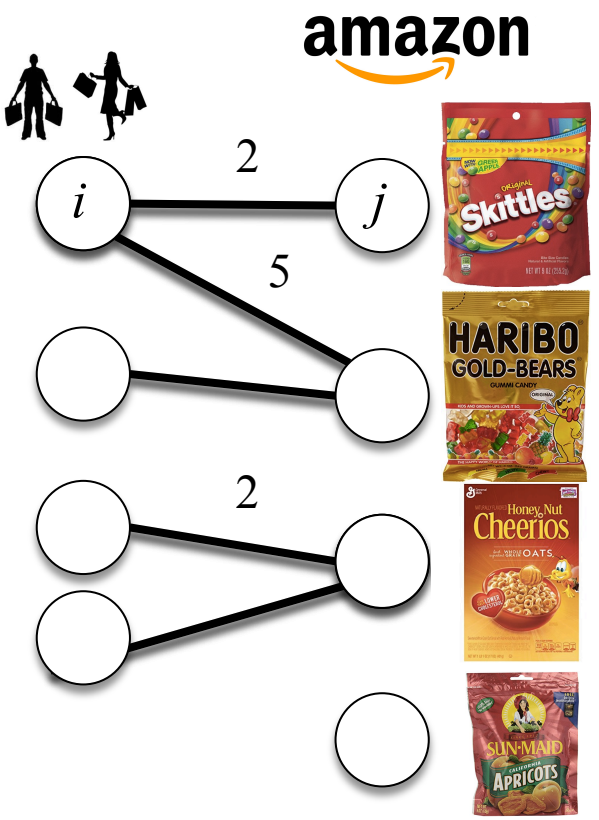
Users have a propensity rate to buy certain category of product ($\mathbf{W}$)

In a category some products have a propensity rate to be bought ($\mathbf{H}$)

$$\mathbf{X}_{ij} \sim \text{Poisson}([\mathbf{WH}]_{ij})$$

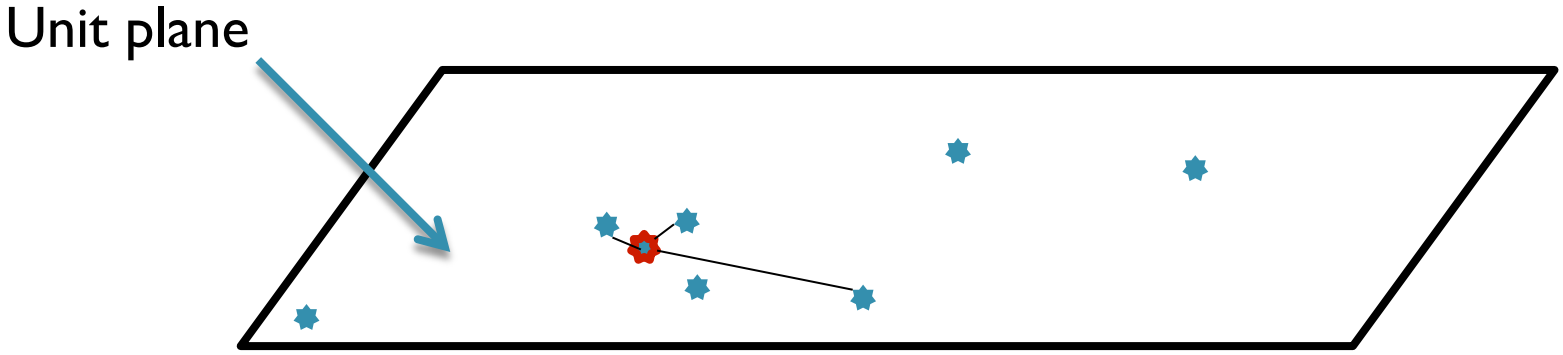
$$\text{MLE} \rightarrow \underset{\mathbf{W},\mathbf{H}}{\text{argmax}} \sum_i \sum_j [X_{ij} \log(\mathbf{WH})_{ij} - (\mathbf{WH})_{ij}]$$

(Non-negative matrix factorization)



Example Euclidean Generative Model

- Vertices uniformly distributed in latent unit hyperplane
- Vertices close in latent space more likely to be connected
- Probability of edge = f(distance on latent space)



The problem of link prediction is to infer distances in the latent space, i.e., find the nearest neighbor in latent space not currently linked

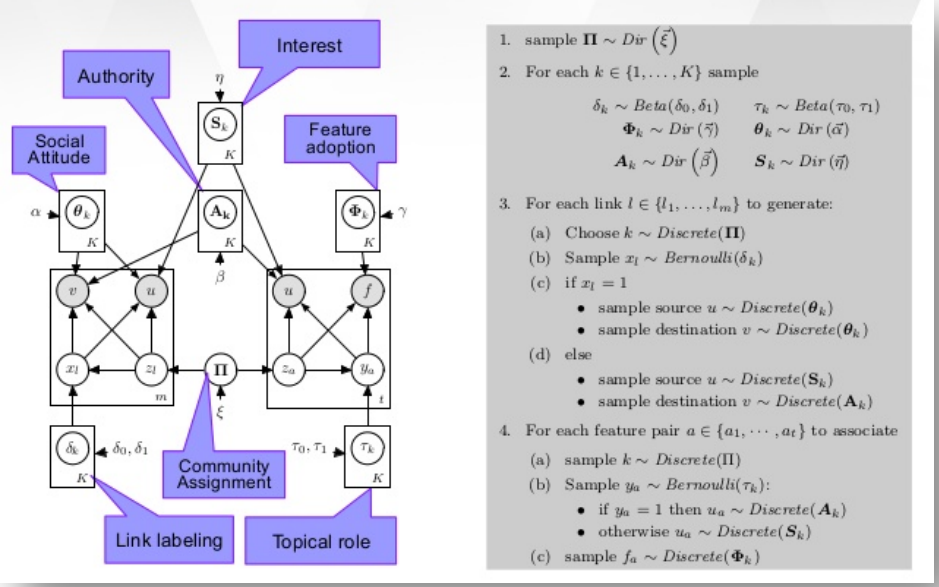
Raftery, A. E., Handcock, M. S., & Hoff, P. D. (2002). Latent space approaches to social network analysis. *J. Amer. Stat. Assoc.*, *15*, 460.

Bayesian Networks

- Directed Graphical Models
- Bayesian networks and Prob. Relational Models
  (Getoor *et al.*, 2001, Neville & Jensen 2003)
- Captures the dependence of link existence on attributes of entities
- Constrains probabilistic dependency to directed acyclic graph
- Undirected Graphical Models Markov Networks

Example of Today's Approaches

Whom To Follow and Why

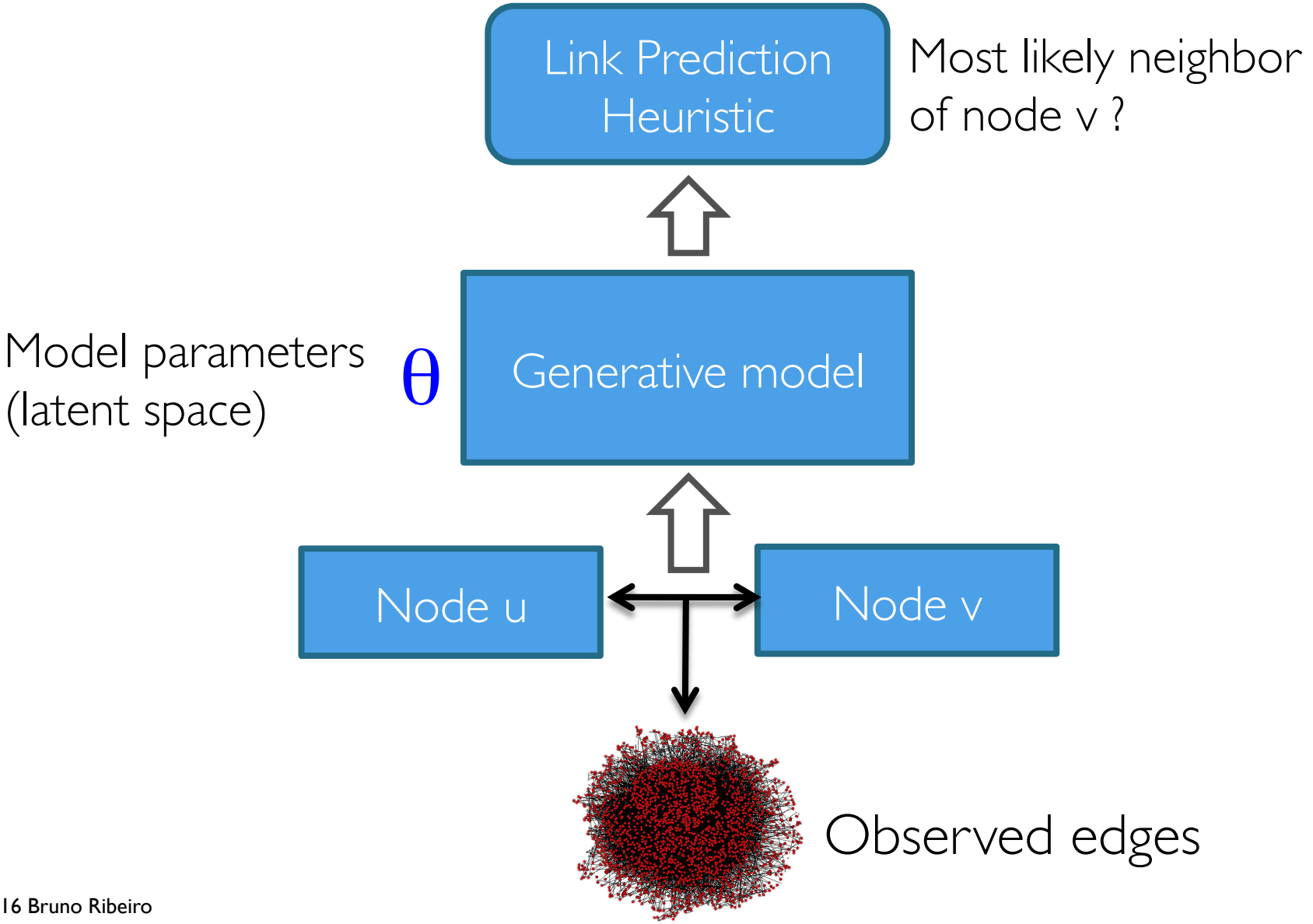


Credit: N. Barbieri, F. Bonchi, G. Manco KDD'14

Relational Markov Networks

- A Relational Markov Network (RMN) specifies cliques and the potentials between attributes of related entities at the template level
- Single model provides distribution for entire graph
- Train model to maximize the probability of observed edge and labels
- Use trained model to predict edges and unknown attributes

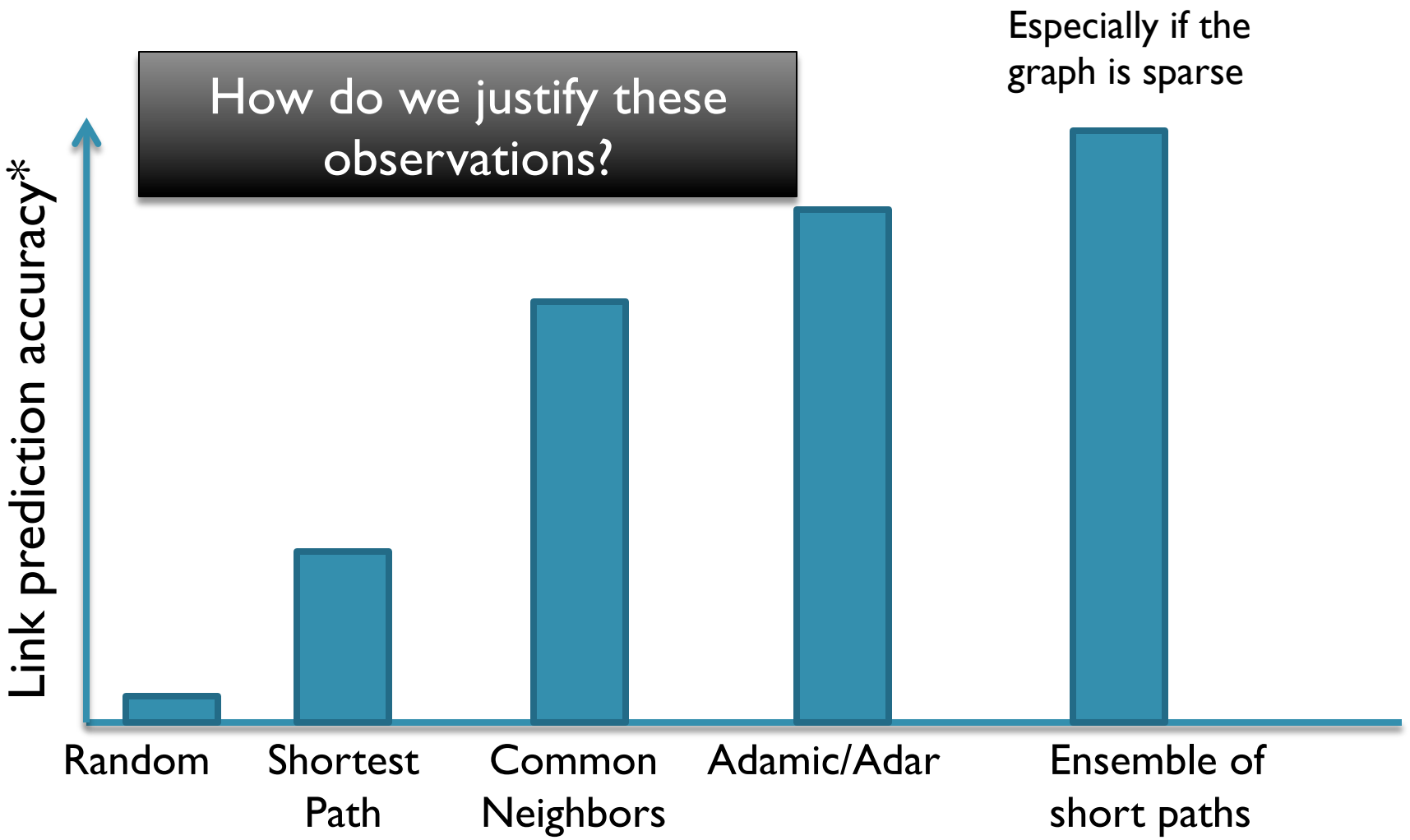
Latent Space Rational



Relationship between Deterministic & Probabilistic methods?

Yes = (Sarkar, Chakrabarti, Moore, COLT'10)

Empirically



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

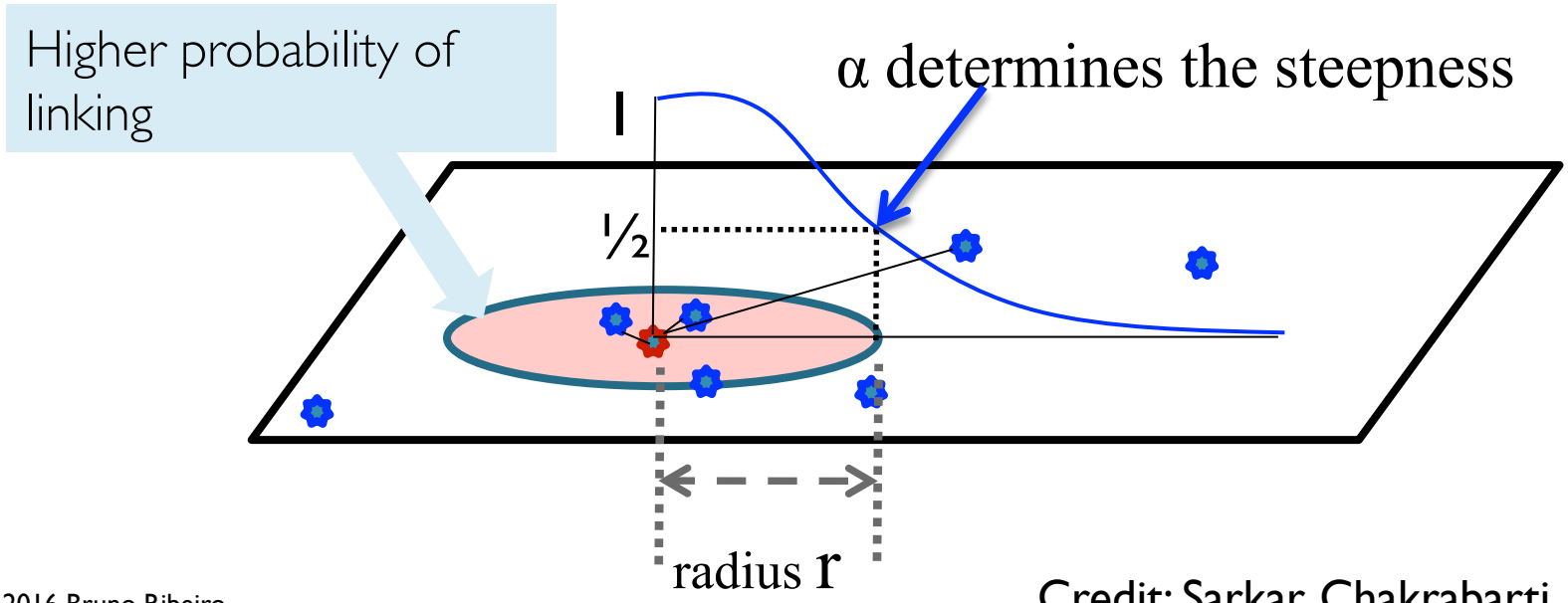
Credit: Sarkar, Chakrabarti, Moore

Raftery's Model (cont)

Two sources of randomness

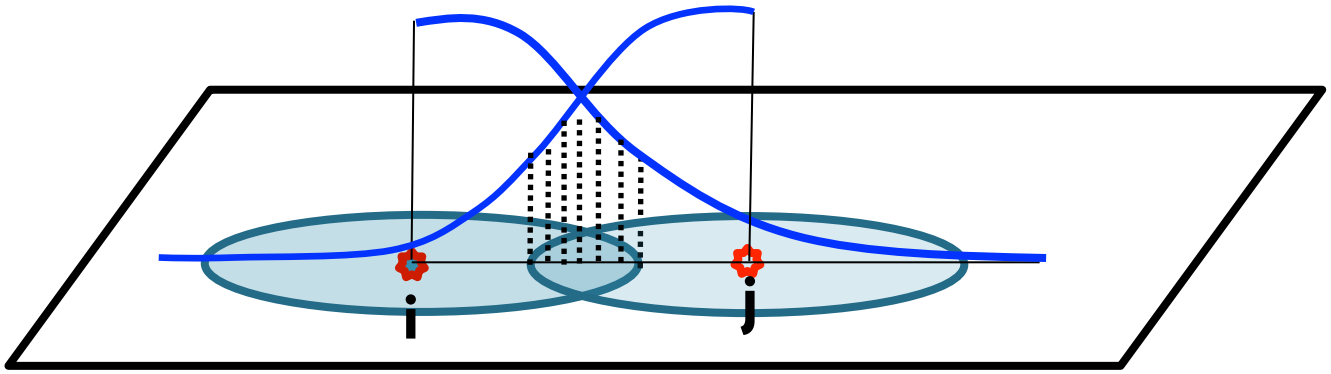
- Point positions: uniform in D dimensional space
- Linkage probability: logistic with parameters **α, r**
- α, r and D are known

$$P(i \sim j | d_{ij}) = \frac{1}{1 + e^{\alpha(d_{ij} - r)}}$$



Credit: Sarkar, Chakrabarti, Moore

Raftery's Model: Probability of New Edge

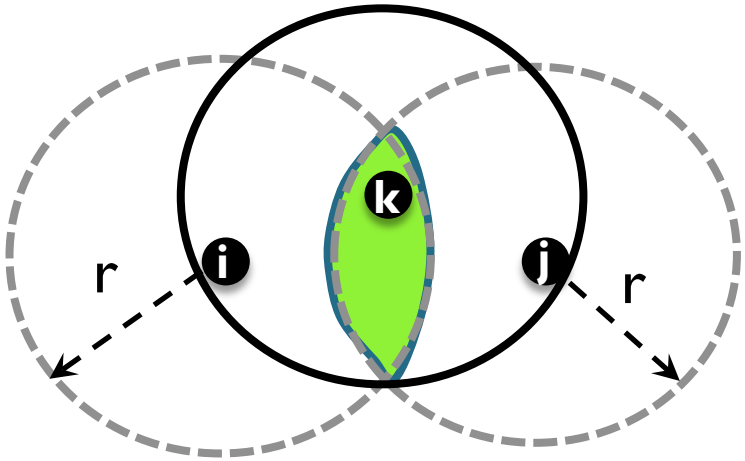


$$P[i \sim j] = \int P[i \sim k|d_{ik}]P[j \sim k|d_{jk}]P[d_{ik}, d_{jk}|d_{ij}]\partial d_{ik}\partial d_{jk}$$

Logistic function integrated over volume determined by d_{ij}

Credit: Sarkar, Chakrabarti, Moore

Connection to Common Neighbors



Distinct radius per node gives Adamic/Adar

$$lim_{\alpha \to \infty} P[i \sim j] \propto \text{No. common neighbors in intersection}$$

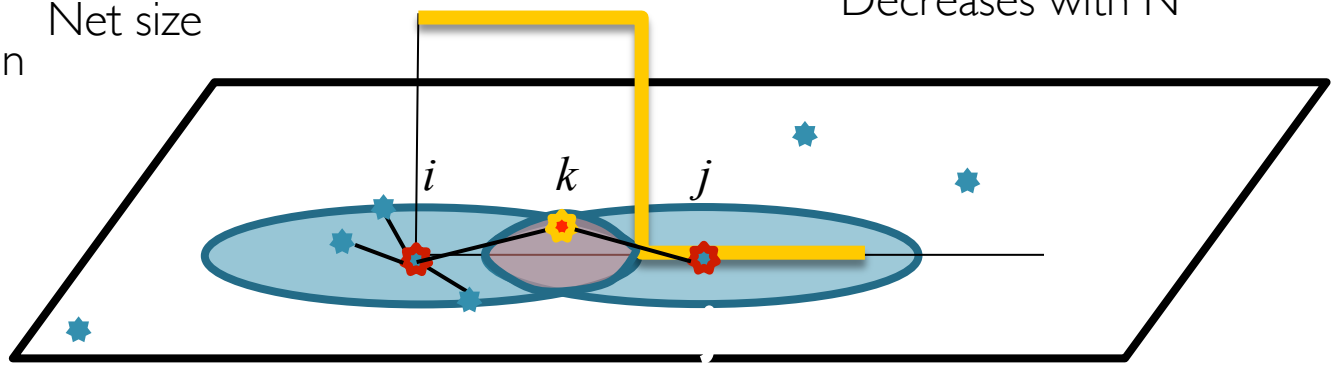
Empirical Bernstein bounds (Maurer & Pontil, 2009)

$$P\left[\left|\sum_k Y_k/N - E[Y_k]\right| \geq \sqrt{\frac{2\text{var}_N(Y)\log 2/\delta}{N}} + \frac{7\log 2/\delta}{3(N-1)}\right] \leq 2\delta$$

No. common neigh.

Net size

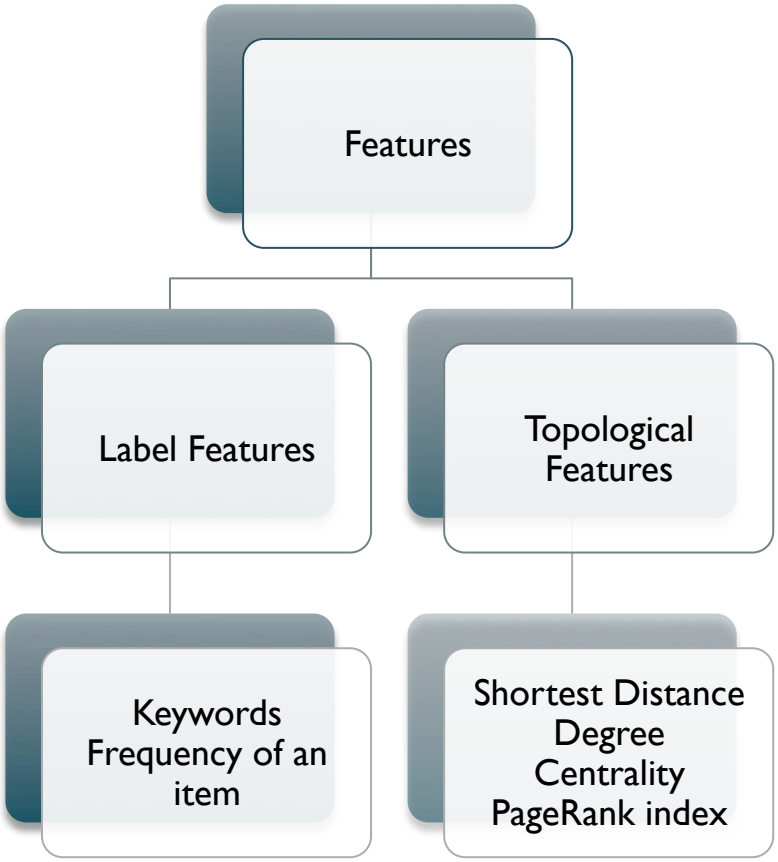
Decreases with N



Supervised Learning Approaches

- Create binary classifier that predicts whether an edge exists between two nodes

Types of Features



Classifier

- SVM
- Decision Trees
- Deep Belief Network (DBN)
- K-Nearest Neighbors (KNN)
- Naive Bayes
- Radial Basis Function (RBF)
- Logistic Regression

Link Prediction Heuristic