Link Prediction

Hossam S. Sharara
Walaa Eldin M. Moustafa

Outline

- Overview
- Link Prediction Variants
- Deterministic Methods
- Probabilistic Methods
- Supervised Learning Approaches
- Feature Construction

Problem Definition

- *Given a snapshot of a social network at time t (or network evolution between t_1 and t_2), we seek to accurately predict the edges that will be added to the network during the interval from time t (or t_2) to a given future time t'.*

Applications

- Identifying the structure of a criminal network
  - Predicting missing links in a criminal network using incomplete data.

Applications

- Overcoming the data-sparsity problem in recommender systems using collaborative filtering (Huang et al, 2005).

Applications

- Accelerating a mutually beneficial professional- or academic connection that would have taken longer to form serendipitously (Farrell et al, 2005).

Picture from www.businessinnovationinsider.com

Applications

- To analyze users' navigation history to generate tools that increase navigational efficiency (Zhu 2003)
  - ie. Predictive server prefetching

Picture from corbis.com

Applications

- Monitoring and controlling computer viruses that use email as a vector (Lim et al, 2005).

Picture from www.robocup.de

Link Completion

- Goldenberg et al, 2003
- Links can be incomplete.
- Links can link more than two entities.
- Given a node (or nodes) that is (are) known to have a link, the task is to determine to which other node the link is attached.

Link Completion

- Example
  - When a user buys five books online and the name of one book is corrupted in transfer.
  - A link completion algorithm could infer the name of the missing book based on the user's name and the other books she bought.

Link Completion

- Example
  - Alice, Bob, and a third person attended a meeting.
  - Given people's previous co-occurrences, who is the third person?

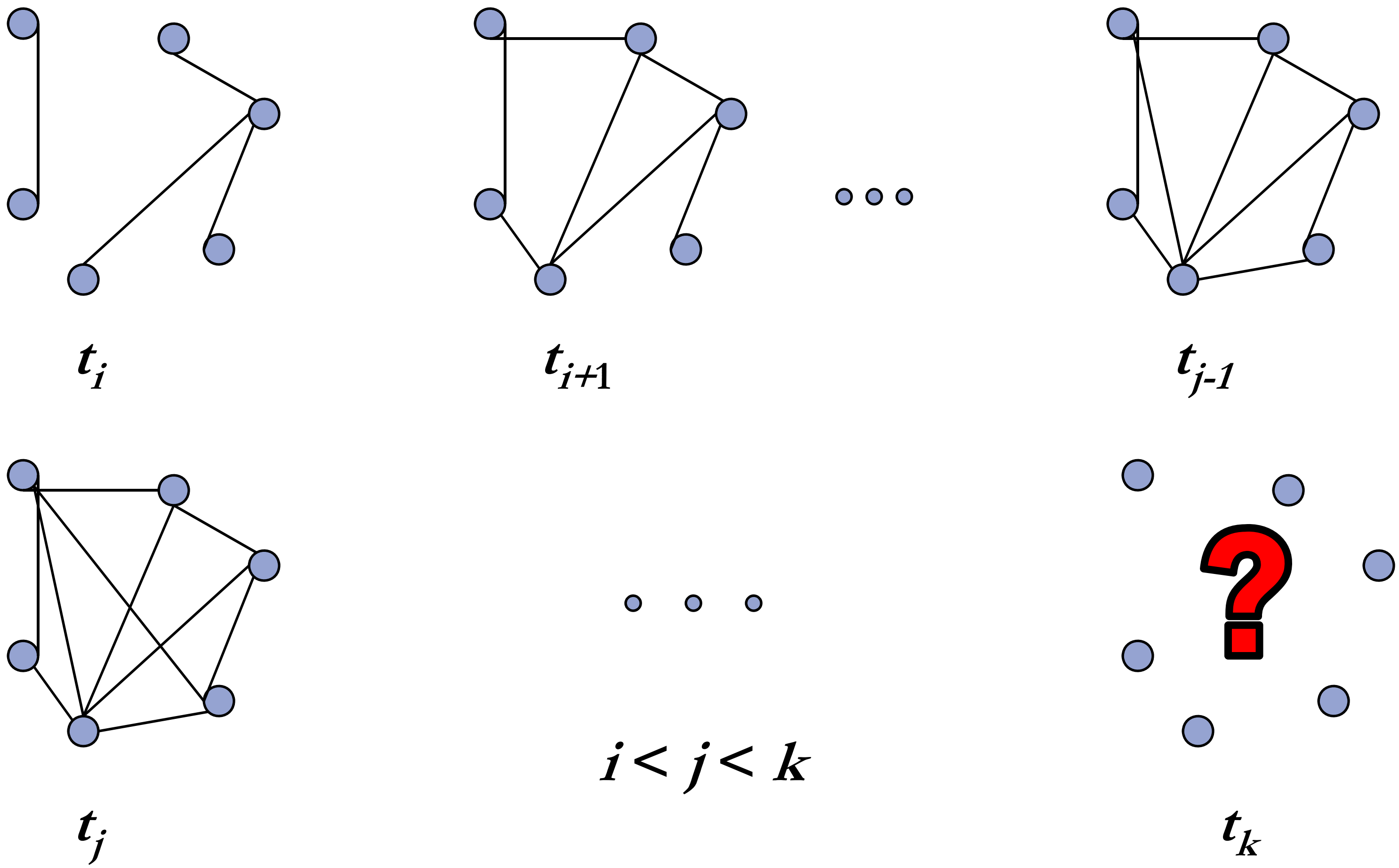
Simple Solutions

- Every entity is assigned a score.
- Co-occurrences:
  - $score(A) = \sum$ number of previous co-occurrences of A and other members of the link.
- Popular Person
  - $score(A)$ = number of occurrences of A in other links.

Anomalous Link Discovery

- Rattigan et al, 2005.
- Link Prediction: Number of Dyads to be evaluated increases quadratically.
- Networks are sparse → extremely few positive cases.
- Focus on discovering surprising links in the existing ones.
- Very few common neighbors, or too distant apart.

Link Prediction



Methods for Link Prediction

- All the methods assign a connection weight *score*(x, y) to pairs of nodes x, y, based on the input graph, and then produce a ranked list in decreasing order of *score*(x, y).
  - Can be viewed as computing a measure of proximity or "similarity" between nodes *x* and *y*.

Shortest Path

- Negated length of shortest path between x and y.
  - All nodes that share one neighbor will be linked.

Common Neighbors

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

- Newman 2001: The probability of scientists collaborating increases with the number of other collaborators they have in common.

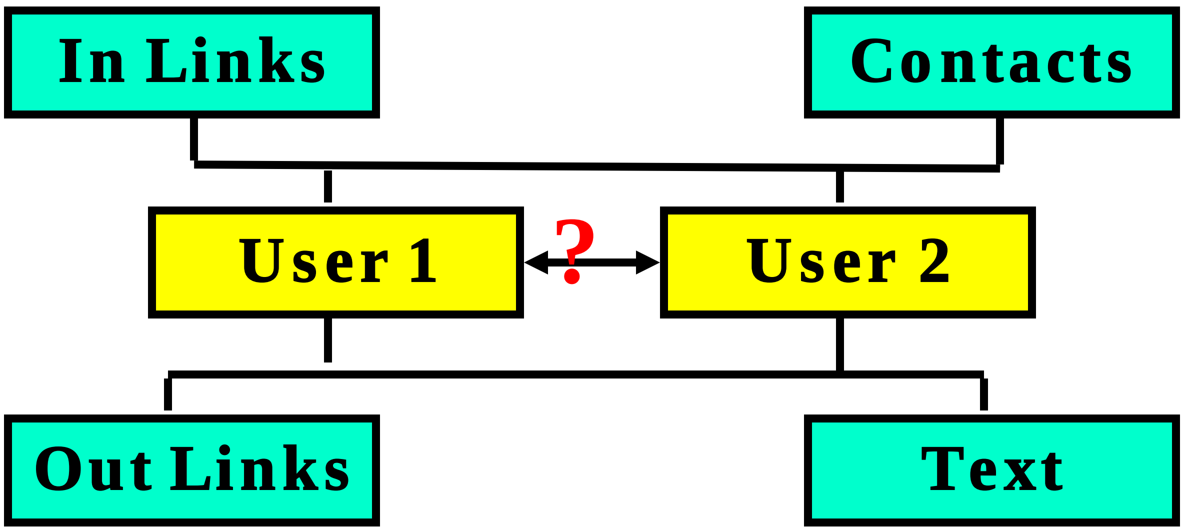
Jaccard Similarity

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

- May be they have common neighbors because each one has a lot of neighbors, not because they are strongly related to each others.

Adamic/Adar

Adamic et al 2003



Adamic/Adar

user 1: kpsounis	**user 2:stoumpis**
Konstantinos Psounis	Stavros Toumpis

Things in common

CITIES:	Escondido, Cambridge, Athens
NOUN GROUPS:	birth date, undergraduate studies, student association
MISC:	general lyceum, NTUA, Ph.D., electrical engineering, computer science, TOEFL, computer
COUNTRIES:	Greece

Out links in common

http://www.stanford.edu/group/hellas	Hellenic association
http://www.kathimerini.gr	Athens news
http://ee.stanford.edu	Electrical Engineering Department
http://www.ntua.gr	National Technical University of Athens

In links in common

http://www.stanford.edu/~dkarali	Dora Karali's homepage
http://171.64.54.173/filarakia.html	Dimitrios Vamvatsikos friends list

Mailing lists in common

greek-sports	Soccer/Basketball mailing lists for members of Hellas
hellenic	Hellenic association members
ee261-list	Fourier transform class list
ee376b	Information theory class list

Adamic/Adar

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

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

- This actually counts common neighbors but:
  - Neighbors who are linked with only 2 nodes are given the weight 1/log(2) = 1.4
  - Neighbors who are linked with 5 nodes their weight drops down to 1/log(5) = 0.62

Preferential Attachment

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

- Newman 2001: The probability of co-authorship of x and y is correlated with the product of the number of collaborators of x and y

Katz (1953)

$$score(x,y) = \sum_{l=1}^{\infty} \beta^l . |paths_{x,y}^{<l>}|$$

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.

A very small β yields predictions much like common neighbors, since paths of length three or more contribute very little to the summation

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

Hitting Time

$-H_{x,y}$

$-H_{x,y} \cdot \pi_y$

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

$-(H_{x,y} \cdot \pi_y + H_{y,x} \cdot \pi_x)$

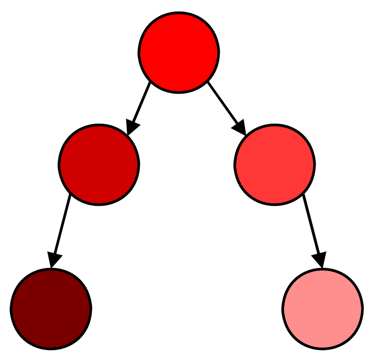
$$\text{where } \begin{array}{rcl} H_{x,y} & := & \text{expected time for random walk from } x \text{ to reach } y \\ \pi_y & := & \text{stationary distribution weight of } y \\ & & \text{(proportion of time the random walk is at node } y\text{)} \end{array}$$

Rooted PageRank

- Stationary distribution weight of y under the following random walk:
  - With probability a, jump to x
  - With probability $1 - a$, go to random neighbor of current node.

SimRank

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



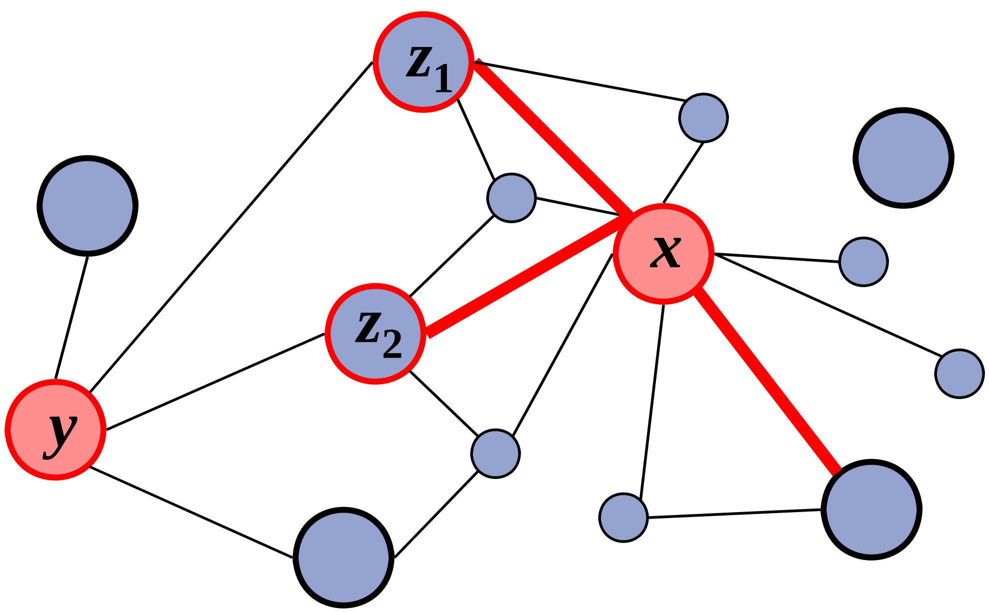
- Jeh 2002
- Only directed graphs
- Can use pruning

Large Graphs

- Represent the adjacency matrix M with a lower rank matrix M_k.
- How to map the original graph to the reduced graph?

Unseen Bigrams

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



Red links: **Strong similarity**

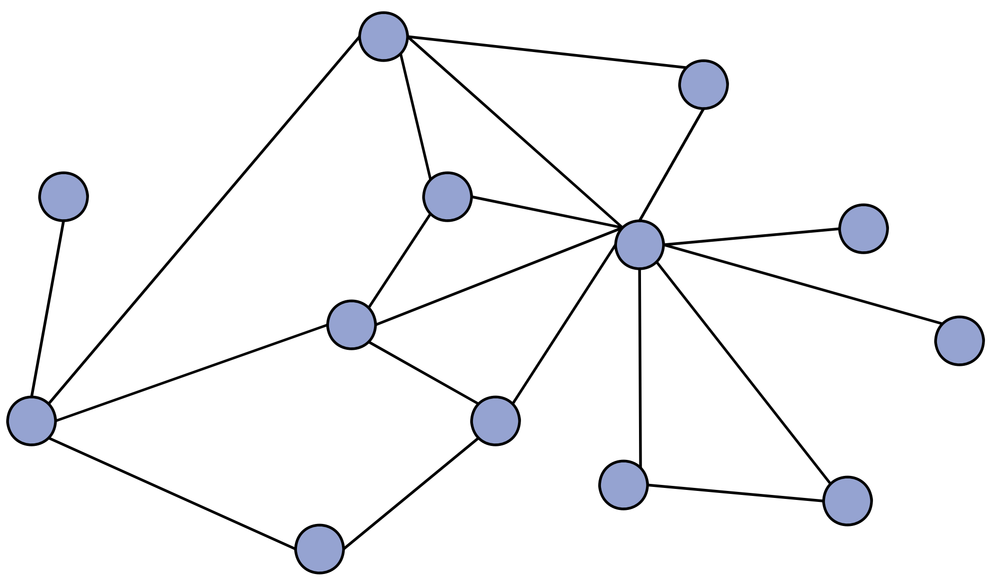
Black Links: **Graph edges**

Unseen Bigrams

- Should we incorporate score(x,y) into the equation?
- Can we do that iteratively?

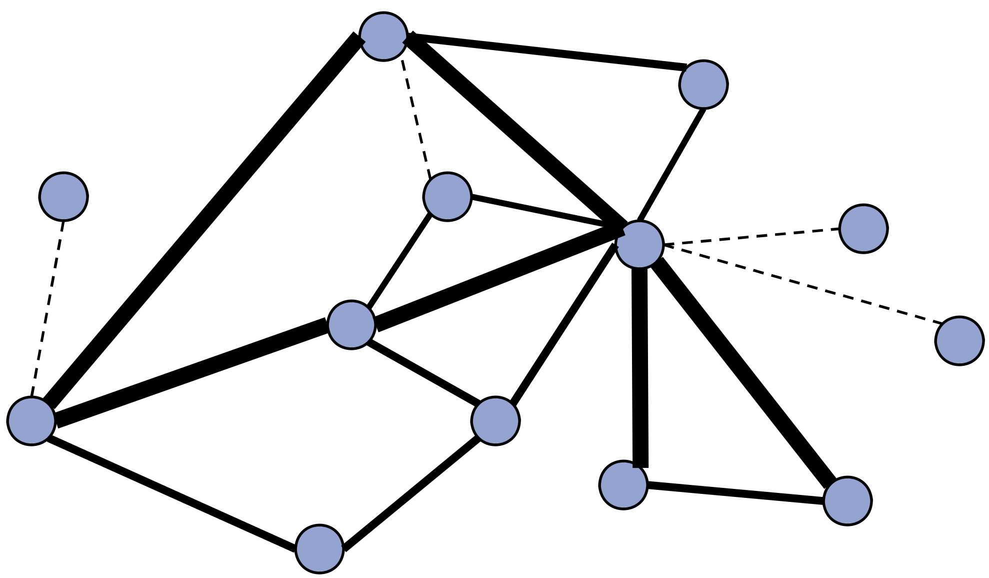
Clustering

- Compute score(u,v) for all edges in E_{old}.
- Remove k% edges with low score.
- Calculate score(x,y) for all node pairs.



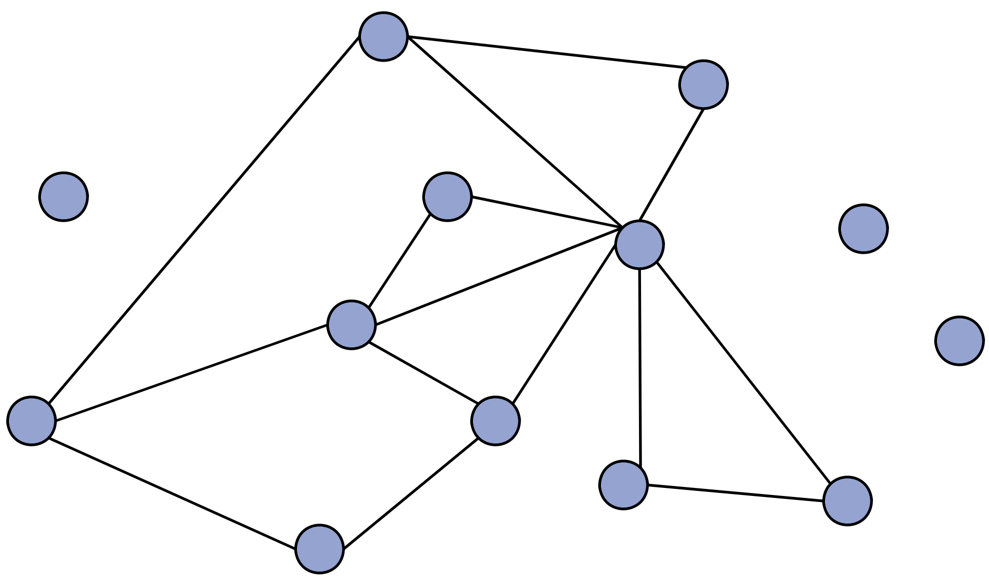
Clustering

- Compute score(u,v) for all edges in E_{old}.
- Remove k% edges with low score.
- Calculate score(x,y) for all node pairs.



Clustering

- Compute score(u,v) for all edges in E_{old}.
- Remove k% edges with low score.
- Calculate score(x,y) for all node pairs.



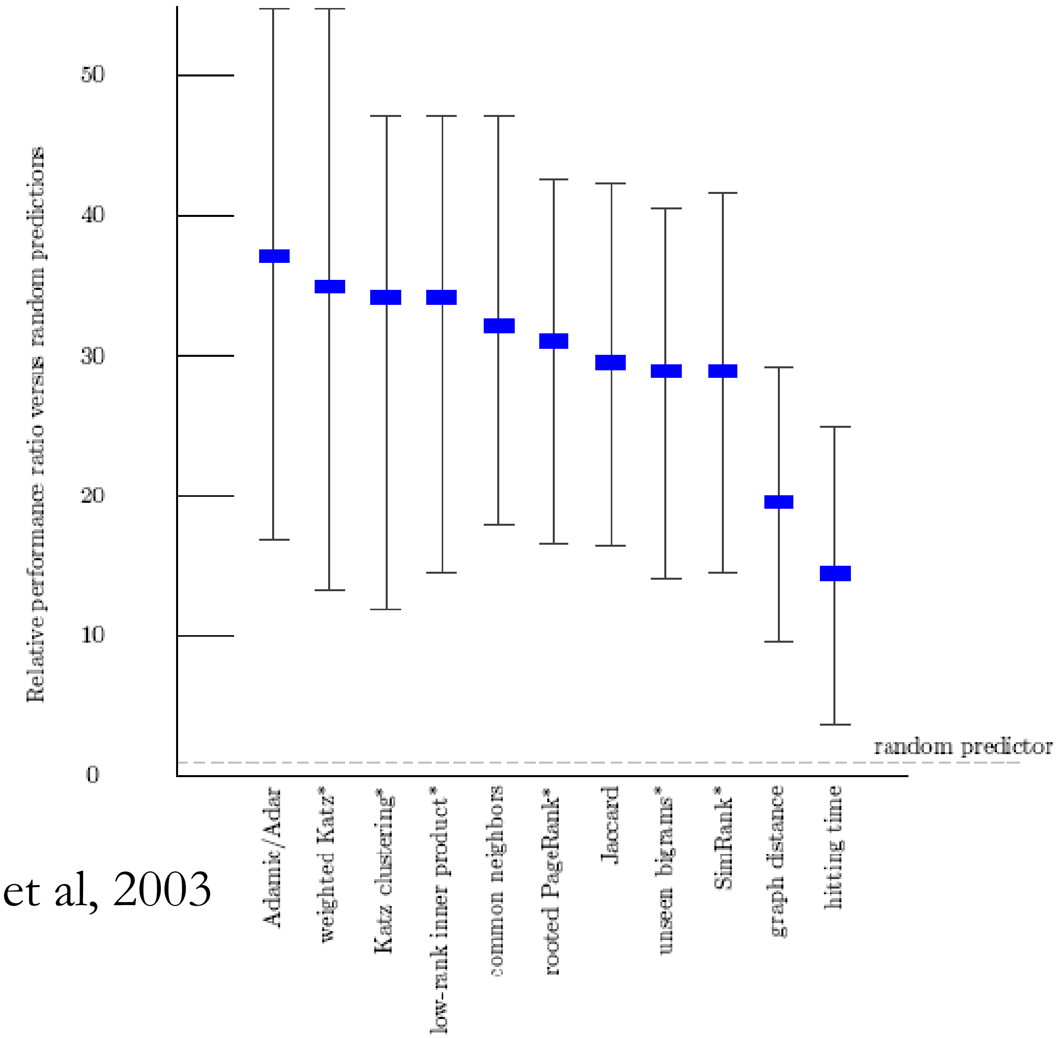
Clustering

- Questions
  - Removing edges can decrease the score of nodes connected to these edges.
  - Will this technique be able to predict the edges that has been removed?
  - How to determine k?

Hybrid Techniques

- Combine metrics to see the effect on link prediction.
  - Features: CN, Katz, Adamic/Adar, …
  - Class: Link formed or not.

Performance Comparison



Liben-Nowell et al, 2003

Observations

- Adamic/Adar and common neighbors perform surprisingly well, despite being simple.
- The accuracies are generally low → still much to improve.

How Close?

	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

How Close?

	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

How Close?

	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

Temporality

- Recency:
  - Potgieter et al, 2008
  - $Rec(i)$ = The number of time steps elapsed since the node has connected to another node.
  - This is monadic metric. Can be transformed to a dyadic metric:
    - $score(x,y) = rec(x).rec(y)$

Temporality

- Potgieter et al, 2008
- Percentage increase or decrease of score over a period of time.

$$score(x,y) = \frac{score_{50}(x,y) - score_1(x,y)}{score_1(x,y)}$$

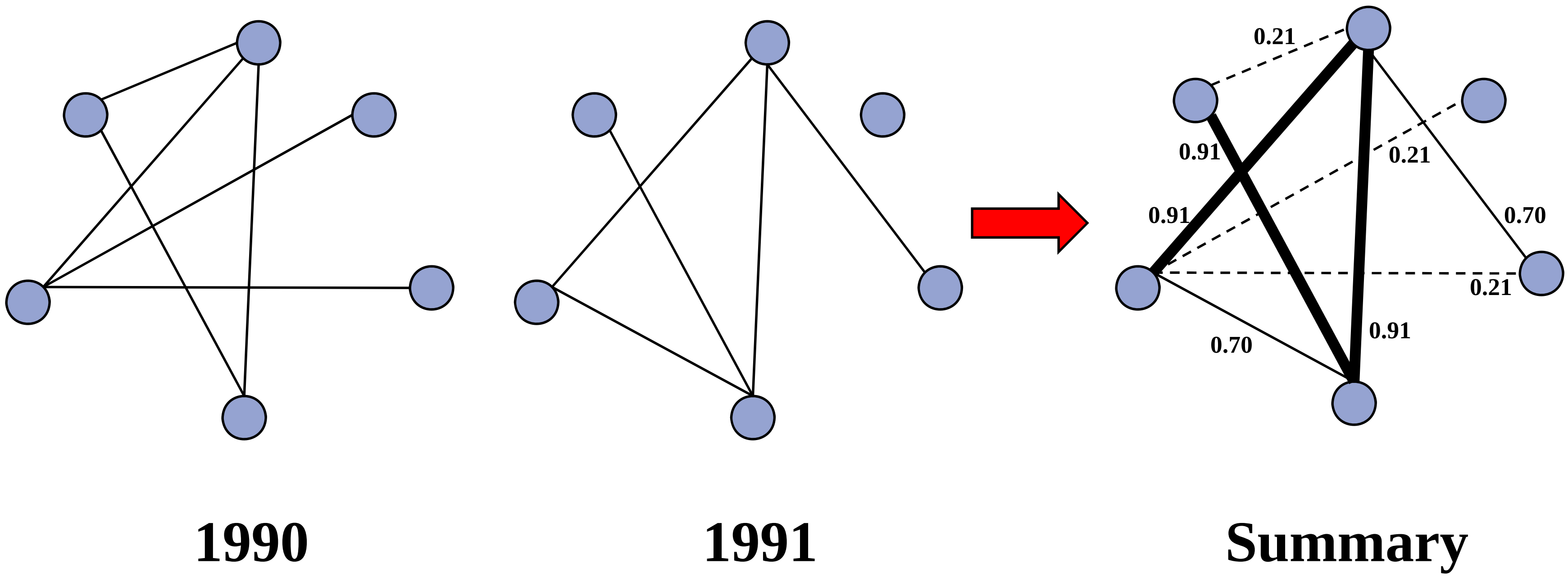
  - Does not capture the "strength" of relationship between nodes.
  - What if
    - $score_1$(a,b)=100, $score_{50}$(a,b) = 120
    - $score_1$(c,d)=1, $score_{50}$(c,d) = 2
    - *score*(c,d) has doubled but it is still low.

Temporality

- Sharan et al, 2007
- Graph Summary
- Simple and efficient
- Can handle both forming and disappearing links
- Stresses on more recent relations

$$G_t^S = \begin{cases} (1-\theta)G_{t-1}^S + \theta G_t & \text{if } t > 1 \\ \theta G_t & \text{if } t = 1 \end{cases}$$

Temporality

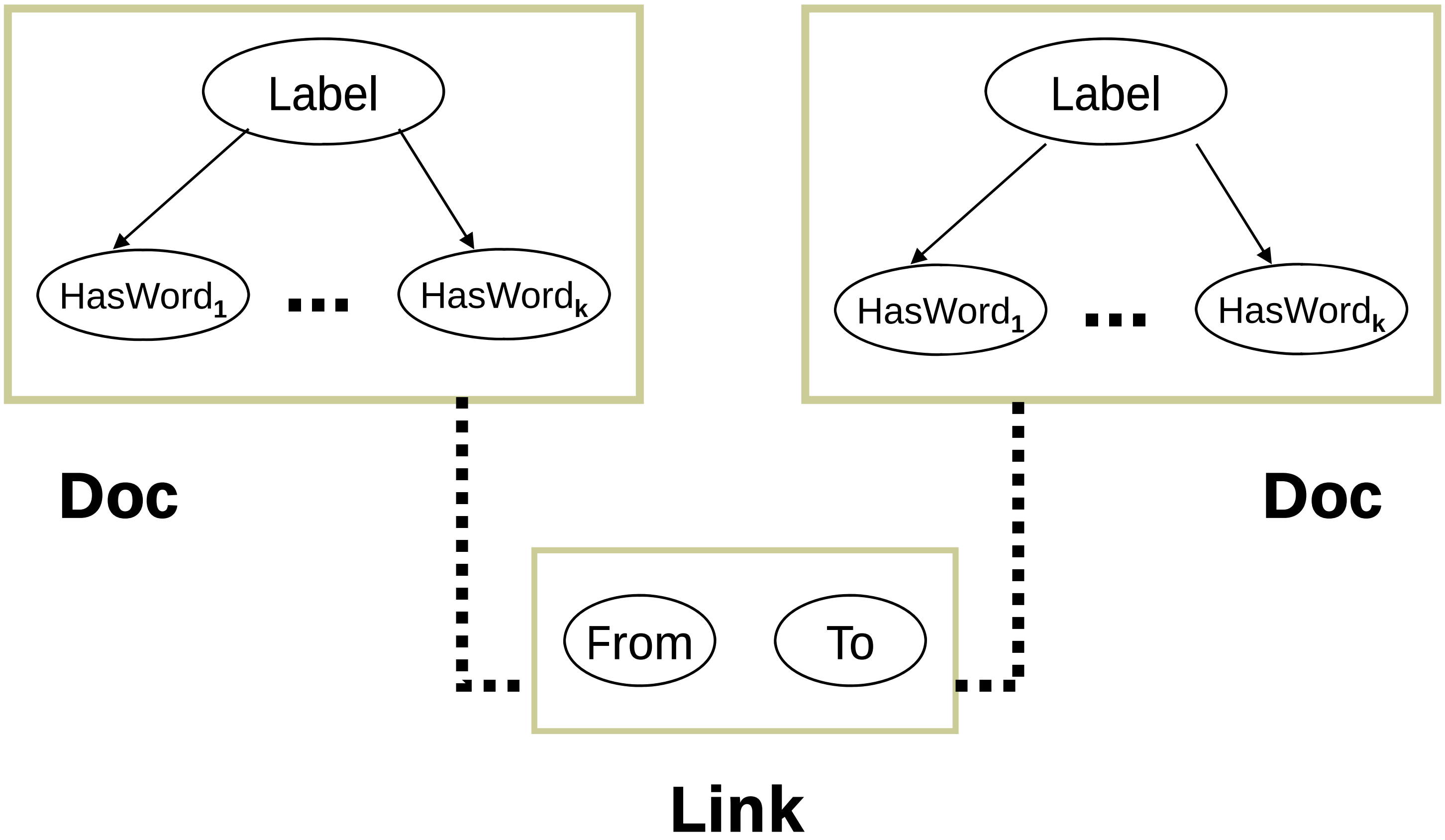


Probabilistic Models

Probabilistic Models

- Directed Graphical Models
  - Bayesian networks and PRMs (Getoor *et al.*, 2001)
  - Easily captures the dependence of link existence on attributes of entities
  - Constraint that the probabilistic dependency graph be a directed acyclic graph
- Undirected Graphical Models
  - Markov Networks

Link Prediction and RMN



Link Prediction and RMN

- Consider both links that exist and those that do not exist

- Each potential link is associated with a binary *existence* attribute *Exists*

Role of RMN

- Define a single probabilistic model over the entire link graph
- Train model to maximize the probability of the (object and) link labels given the known attributes
- Use probabilistic inference to predict and classify links using any observed attributes and other links.

Markov Networks

- A Markov network is an undirected graphical model that defines a joint distribution over some set of random variables **V**

- Components:
  - Undirected dependency graph **G**
  - Qualitative component – Cliques
    - A clique ***C*** is a set of nodes **Vc** in graph **G**, where each V_i, $V_j \in$ **Vc** are connected by an edge in G
  - Quantitative component – Potentials
    - Clique Potential $\boldsymbol{\phi_c(V_c)}$ is a table of values that define a *"compatibility"* between values of variables in the clique

Markov Networks

- Markov Network defines the distribution

$$P(v) = \frac{1}{Z} \prod_{c \in C(G)} \phi_c(v_c)$$

v $\longrightarrow$ an assignment of values to random variables **V**

C(G) $\longrightarrow$ set of cliques in **G**

Z $\longrightarrow$ *partition function* (a normalization constant) which is the sum of the product of potential functions of cliques in *C(G)* over all possible assignments

$$Z = \sum_{v'} \prod \phi_c(v'_c)$$

Clique Potential

- Representation
  - Log-Linear combination of small set of indicator functions (*features*)

$$f(V_c) \equiv \delta(V_c = v_c)$$

Thus $$\phi_c(v_c) = \exp\{\sum_i w_i f_i(v_c)\} = \exp\{\mathrm{w}_c.\mathrm{f}_c(v_c)\}$$

Hence we can write
$$\log P(v) = \sum_c \mathrm{w}_c.\mathrm{f}_c(v_c) - \log Z = \mathrm{w.f}(v) - \log Z$$

where **w** and **f** are vectors of all weights and features

Conditional Markov Networks

- Specify the probability of a set of target variables **Y** given a set of conditioning variables **X**

$$P(y \mid x) = \frac{1}{Z(x)} \prod_{c \in C(G)} \phi_c(x_c, y_c)$$

Where partition function now is dependent on ***x***

$$Z(x) = \sum_{y'} \prod \phi_c(x_c, y'_c)$$

Relational Markov Networks

- A Relational Markov network specifies cliques and the potentials between attributes of related entities at a template level
- Single model can provide a coherent distribution for any collection of instances from the schema

Relational Clique Template

- Specify tuples of variables in the network instantiation using relational query language

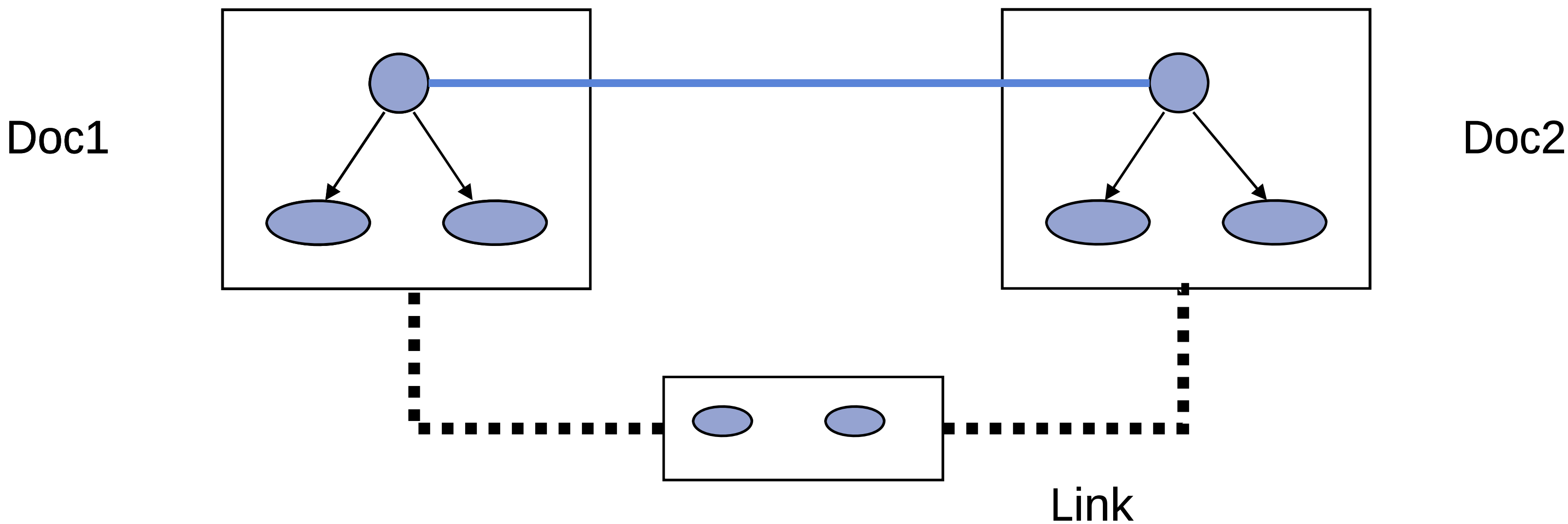
- Components
  - **F** → a set of entity variables (From)
  - **W**→ condition about the attributes of entity variables (Where)
  - **S** → subset of attributes of the entity variables (Select)

Relationship to SQL query

SELECT doc1.Category,doc2.Category

FROM doc1,doc2,Link link

WHERE link.From=doc1.key and link.To=doc2.key

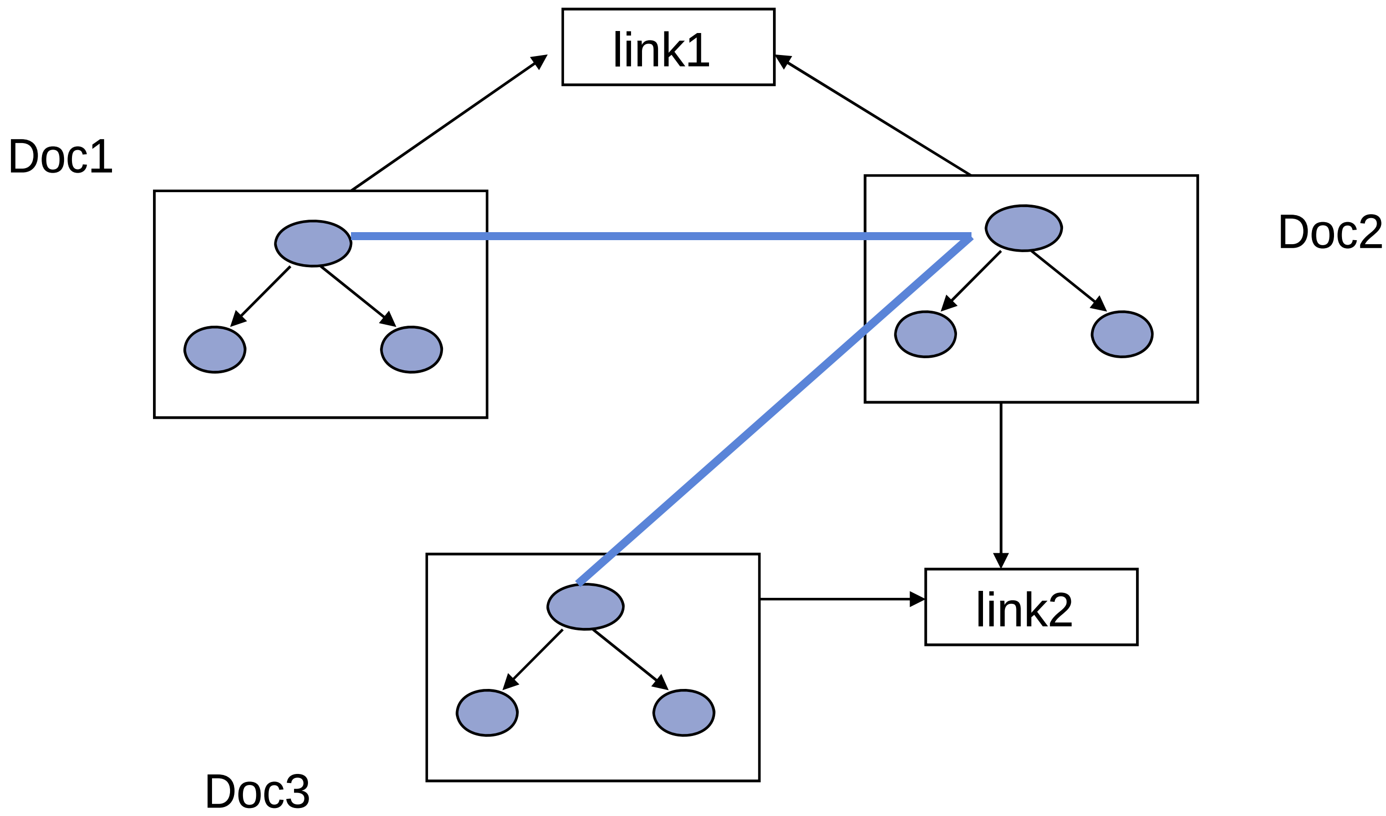


Slide adopted from Presentation by Guohua Hao, Oregon State University

Unrolling the RMN

- Given an instantiation of a relational schema, unroll the RMN as follows
  - Find all the cliques in the unrolled the relational schema where the relational clique templates are applicable
  - The potential of a clique is the same as that of the relational clique template which this clique belongs to

Unrolling the RMN



Slide adopted from Presentation by Guohua Hao, Oregon State University

Probability in RMN

$$P(I.\mathbf{y} \mid I.\mathbf{x}, I.\mathbf{r}) = \frac{1}{Z(I.\mathbf{x}, I.\mathbf{r})} \prod_{C \in \mathbf{C}} \prod_{c \in C(I)} \phi_C(I.\mathbf{x}_c, I.\mathbf{y}_c)$$

$$Z(I.\mathbf{x}, I.\mathbf{r}) = \sum_{I.\mathbf{y}'} \prod_{C \in \mathbf{C}} \prod_{c \in C(I)} \phi_C(I.x_c, I.y_c')$$

C $\longrightarrow$ Set of clique templates

C(I) $\longrightarrow$ Set of cliques in instantiation *I* from a template C

Log-Linear Model in RMN

$$\begin{aligned}
&\log P(\mathcal{I}.\mathbf{y} \mid \mathcal{I}.\mathbf{x}, \mathcal{I}.\mathbf{r}) \\
&\quad = \sum_{C \in \mathbf{C}} \sum_{c \in C(\mathcal{I})} \mathbf{w}_C \cdot \mathbf{f}_C(\mathcal{I}.\mathbf{x}_c, \mathcal{I}.\mathbf{y}_c) - \log Z(\mathcal{I}.\mathbf{x}, \mathcal{I}.\mathbf{r}) \\
&\quad = \sum_{C \in \mathbf{C}} \mathbf{w}_C \cdot \mathbf{f}_C(\mathcal{I}.\mathbf{x}, \mathcal{I}.\mathbf{y}, \mathcal{I}.\mathbf{r}) - \log Z(\mathcal{I}.\mathbf{x}, \mathcal{I}.\mathbf{r}) \\
&\quad = \mathbf{w} \cdot \mathbf{f}(\mathcal{I}.\mathbf{x}, \mathcal{I}.\mathbf{y}, \mathcal{I}.\mathbf{r}) - \log Z(\mathcal{I}.\mathbf{x}, \mathcal{I}.\mathbf{r})
\end{aligned}$$

where

$$\mathbf{f}_C(\mathcal{I}.\mathbf{x}, \mathcal{I}.\mathbf{y}, \mathcal{I}.\mathbf{r}) = \sum_{c \in C(\mathcal{I})} \mathbf{f}_C(\mathcal{I}.\mathbf{x}_c, \mathcal{I}.\mathbf{y}_c)$$

Learning in RMN

- To get clique potentials, we need to estimate the feature weights (**w**)
- Any particular setting for **w** fully specifies a probability distribution $P_{\boldsymbol{w}}$ on training data
- Use gradient descent over **w** to find the weight setting that maximizes the likelihood (ML) of the link existence given other attributes

Inference in RMN

- Compute the posterior distribution over the link label variables in the instantiation given the other variables.
- Exact inference is intractable in the cases where and network is large and densely connected.
- *Belief Propagation* is used to provide good approximation to the correct posteriors.

Back to Link Prediction

- Factors affecting the relations of different entities:
  - Entity's Attributes

    Properties, Labels
  - Entity's Structural Properties

    Similarity, Transitivity

Link Prediction and RMN

- RMN easily capture attribute correlations and graph-based properties using relational clique templates
- Even more complex patterns can captured using cliques that represent the dependencies and correlations

Experiments

- Dataset:
  - CS dept webpages from 3 schools
  - 8 labels

    organization, student, research group, faculty, course, research project, research scientist, staff
  - 5 relationship types

    Advisor, Member, Teach, TA, Part-Of

Experiments

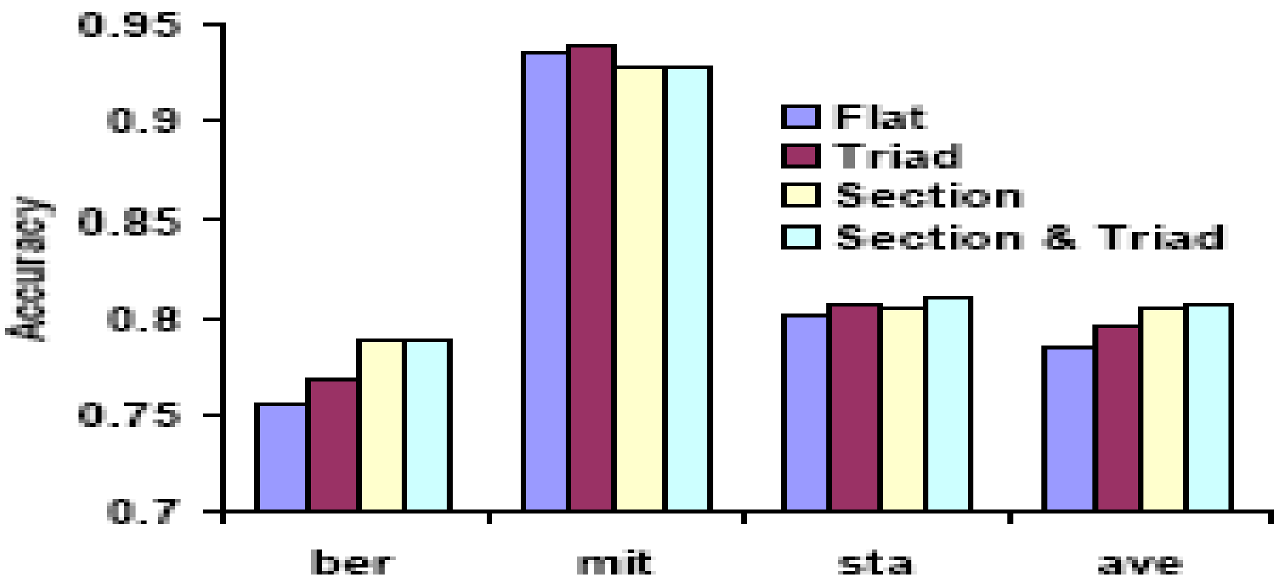
- Observed Attributes
  - Page
    - *words on the page*
    - *meta-words (title, heading, anchors, etc…)*
  - Link
    - *anchor text*
    - *text before link*
    - *heading of section where link appears*
- Target attribute is link type

Experiments

- Two Experiments
  - *Observed Page Labels*
  - *Unobserved Page Labels*

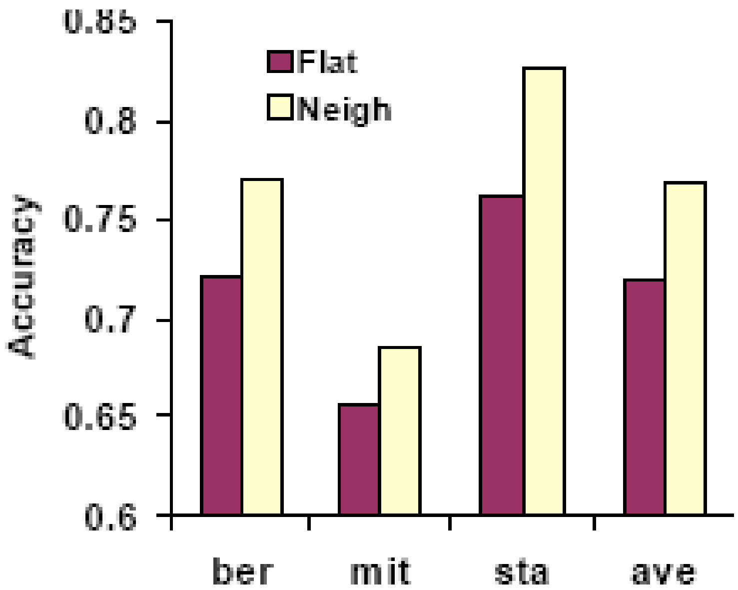
- Train on two schools and test on the remaining one

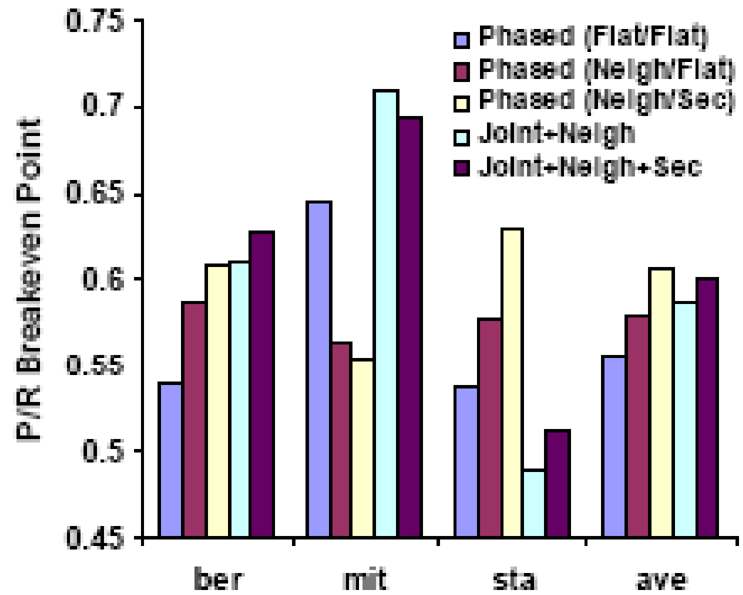
Observed Entity Label



Flat → Baseline, predicting one link at a time using logistic regression
Triad → Transitivity template (cliques over 3 candidate links to form triangle)
Section → Similarity template (cliques over relations whose links appear consecutively in a section on a page)

Unobserved Entity Label





Neigh → Pages with similar URLs often belong to similar/related categories

Phased → First classify the pages, then use add labels to classify links

Joint → Predict both page and relation labels simultaneously

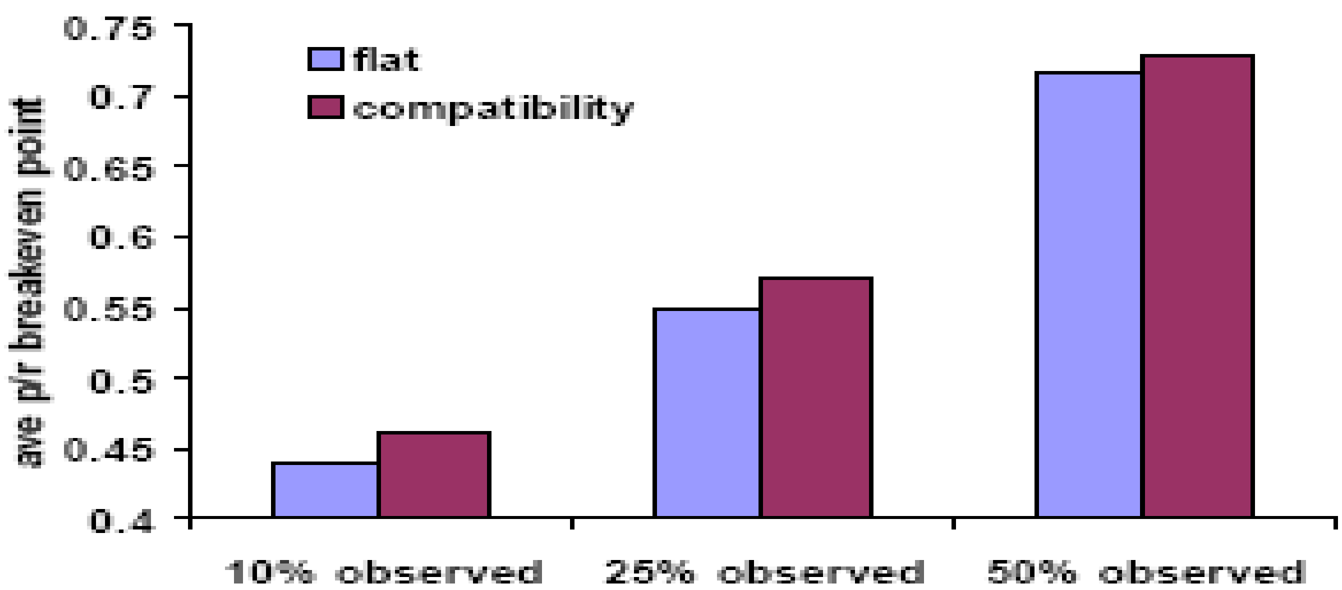
Experiment on Social Network Data

- Dataset:
  - Online community for students
  - 16 residences
  - Friendship relationship

Experiment on Social Network Data

- Eight-fold train/test splits (Trained on 14 residence, tested on two)
- Some proportion (*10%, 25%, and 50%*)of the actual links is included in the test data (randomly selected), used to provide evidence to detect the rest

Experiment on Social Network Data



- **Compatibility →** Similarity template (cliques to capture similarity between each pair of links emanating from each person)

Supervised Learning Approaches

Supervised Learning Approaches

- Learning a binary classifier that will predict whether a link exists between a given pair of nodes or not

(Hassan *et al.*, 2006)

Experimental Setup

- Dataset:
  - Co-authorship Network

- Partition the range of publication year into two non-overlapping sub-ranges; one used as a training set and the other as a test set

Classification Dataset

- Choosing author pairs that appeared in the train years, but did not publish any papers together in those years

__Positive Example__

published at least one paper together in test years

__Negative Example__

didn’t publish any paper together in test years

Experimental Setup

- Datasets Used:
  - DBLP
    - Train (1990 – 2000), Test (2001 – 2004)
  - BIOBASE
    - Train (1998 – 2001), Test (2002)
- Author pairs representing positive and negative examples chosen at random from eligible lists

Experimental Setup

Features

- Proximity Features
  - *Keyword match count*
- Aggregation Features
  - *Sum of Papers*
  - *Sum of Neighbours*
- Topological Features
  - *Shortest Distance*
  - *Clustering Index*

Algorithms

- SVM
- Decision Trees
- Multilayer Perceptron
- KNN
- Naïve Bayes
- RBF
- Bagging

Results

BIOBASE

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

DBLP

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

Feature Construction

Feature Generation and Selection

- Traditional Approach
  - Manually generate a set of possible predictors
  - Model selection process to make decisions regarding their inclusion in a model

- Problems
  - Not always obvious what features should be generated
  - Crucial to navigate richer data structures to discover potentially new and complex sources of relevant evidence, which are not always immediately obvious to humans.

Search Heuristics Approach

- Proposed by *Alexandrin Popescul* and *Lyle H. Ungar*

  (Popescul *et al.*, 2003)
- Coupling feature generation and selection processes
- Feature generation process as *search in the space of relational database queries*
- The level of complexity of search space can be controlled by specifying the types of queries allowed in the search

Search Space in Citation Network

- Schema

  Citation(*from*:Document, *to*:Document),
  Author(*doc*:Document, *auth*:Person),
  PublishedIn(*doc*:Document, *vn*:Venue),
  WordCount(*doc*:Document, *word*:Word, *cnt*:Int).

- Example Generated Attribute

  Average count of the word "learning" in documents cited from an example document d^1

$$ave_{cnt}[\sigma_{word='learning' \wedge from='d'}(Citation \bowtie_{to=doc} WordCount)]$$

Relational Feature Generation

- Refinement Graphs
  - Starts with most general clauses and progresses by refining them into more specialized ones
  - A search node is expanded, or refined, applying a *refinement operator* to produce its most general specializations
- Application
  - Introduces aggregation into the search space to produce scalar numeric values to be used as features in statistical learning

Exploration vs. Exploitation

- The search space is potentially infinite, but not all subspaces will be equally useful
- Sampling
  - Used from subspaces of same type at the time of node expansion

Explore that subspace

Exploit different search spaces

Feature Selection

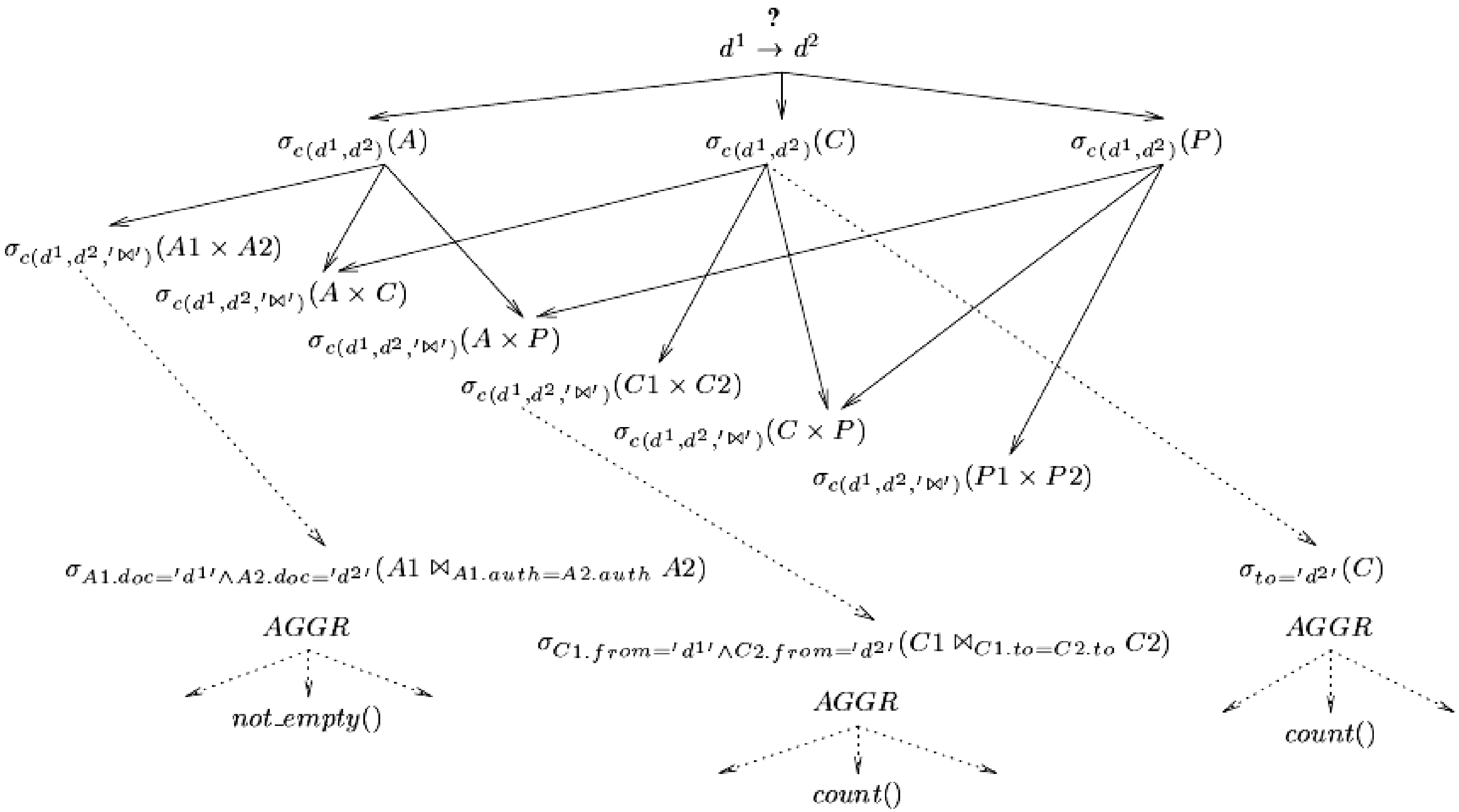
- Different Models are utilized to maximize the likelihood function
(The probability that the training data is generated by a model with learned coefficients)
  - Example: *logistic regression*

- Problem
  - More complex models will results in higher likelihood values, but at some point will likely overfit the data

Feature Selection

- Bayesian Information Criterion (BIC)
  - Works by penalizing the likelihood by a term that depends on model complexity
- Proposed Solution
  - Use stepwise model selection to find a model which generalizes well by adding one predictor at a time as long as the BIC can still be improved.

Fragment of Search Space



Obtained Features

- Obvious Features:
  - d^1 is more likely to cite d^2 if d^2 is frequently cited
  - d^1 is more likely to cite d^2 if the same person co-authored both documents
- More Interesting Features:
  - A document is more likely to be cited if it is cited by frequently cited documents

References

- David Liben-Nowell, Jon M. Kleinberg: The link prediction problem for social networks. CIKM 2003: 556-559
- Lada A. Adamic and Eytan Adar. Friends and neighbors on the web. Social Networks, 25(3):211{230, July 2003.
- Glen Jeh and Jennifer Widom. SimRank: A measure of structural-context similarity. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 2002.
- A. Goldenberg, J. Kubica, P. Komarek, A. Moore and J. Schneider. "A Comparison of Statistical and Machine Learning Algorithms on the Task of Link Completion". KDD Workshop on Link Analysis for Detecting Complex Behavior, 2003.

References

- M. E. J. Newman. The structure of scientifc collaboration networks. Proceedings of the National Academy of Sciences USA, 98:404-409, 2001.

- Huang, Z, Li, X & Chen, H 2005. 'Link Prediction Approach to Collaborative Filtering', Proceedings of the 5th ACM/IEEECS joint conference on Digital libraries, 7-11 June 2005

- Farrell, S., Campbell, C. & Myagmar, S. 2005, Relescope: An Experiment in Accelerating Relationships. Conference on Human Factors in Computing Systems, 2-7 April 2005, Portland.

References

- Zhu, J. 2003. Mining Web Site Link Structures for Adaptive Web Site Navigation and Search. PhD thesis, University of Ulster.
- A. Potgieter, K. April, R. Cooke & I. O. Osunmakinde, (2006). Temporality in Link Prediction: Understanding Social Complexity, Journal of Transactions on Engineering Management, Accepted for publication
- U. Sharan and J. Neville. Exploiting Time-Varying Relationships in Statistical Relational Models. In Proceedings of the 1st SNA-KDD Workshop, 13th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2007.

References

- Rattigan, Jensen. The case for anomalous link discovery. ACM SIGKDD, Dec. 2005.
- Leo Katz. A new status index derived from sociometric analysis. Psychometrika, 18(1):39-43, March 1953.
- Lim, M., Negnevitsky, M. & Hartnett, .J 2005, Artificial Intelligence Applications for Analysis of E-mail Communication Activities. Proceedings International Conference On Artificial Intelligence In Science And Technology, p.p. 109-113 .

References

- B. Taskar, M. F. Wong, P. Abbeel and D. Koller. "Link Prediction in Relational Data", *Neural Information Processing Systems Conference (NIPS03)*, Vancouver, Canada, December 2003.

- Alexandrin Popescul, Lyle H. Ungar. "Statistical Relational Learning for Link Prediction", *Workshop on Learning Statistical Models from Relational Data at IJCAI,* 2003.

- L. Getoor, N. Friedman, D. Koller, and B. Taskar." Probabilistic models of relational structure", *In Proceedings of ICML*, 2001.

- Mohammad Hasan, Vineet Chaoji, Saeed Salem, Mohammed J. Zaki. "Link Prediction using Supervised Learning", *Workshop on Link Analysis, Counter-terrorism and Security (with SIAM Data Mining Conference)*, Bethesda, MD, April 2006

Thank you