Link-based Similarity Measurement Techniques and Applications

Zhenjiang Lin

Supervisors:
Prof. Irwin King
Prof. Michael R. Lyu

Department of Computer Science & Engineering
The Chinese University of Hong Kong

香港中文大學
The Chinese University of Hong Kong

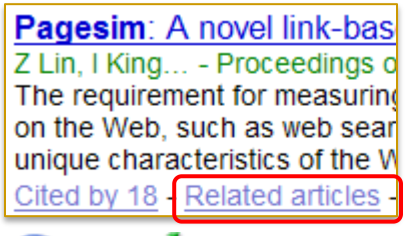
24 Aug 2011

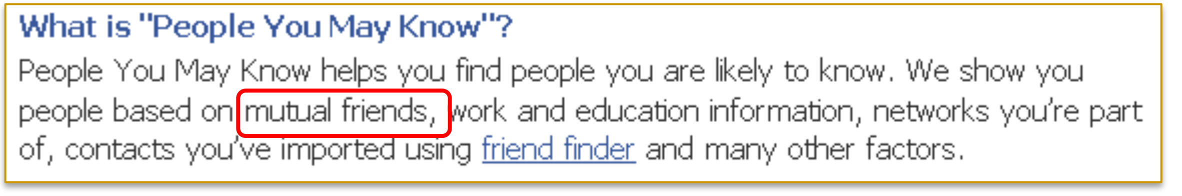
Outline

- Background
- Link-based Similarity Measurement
  - Part 1. **MatchSim**: Maximum Neighborhood Matching
  - Part 2. **PageSim**: Object's Feature Propagation
  - Part 3. **ENS**: Extended Neighborhood Structure Model
- Item-based Top-*N* Recommendation
  - Part 4. **GCP**: Generalized Conditional Probability
- Conclusion & Future Work

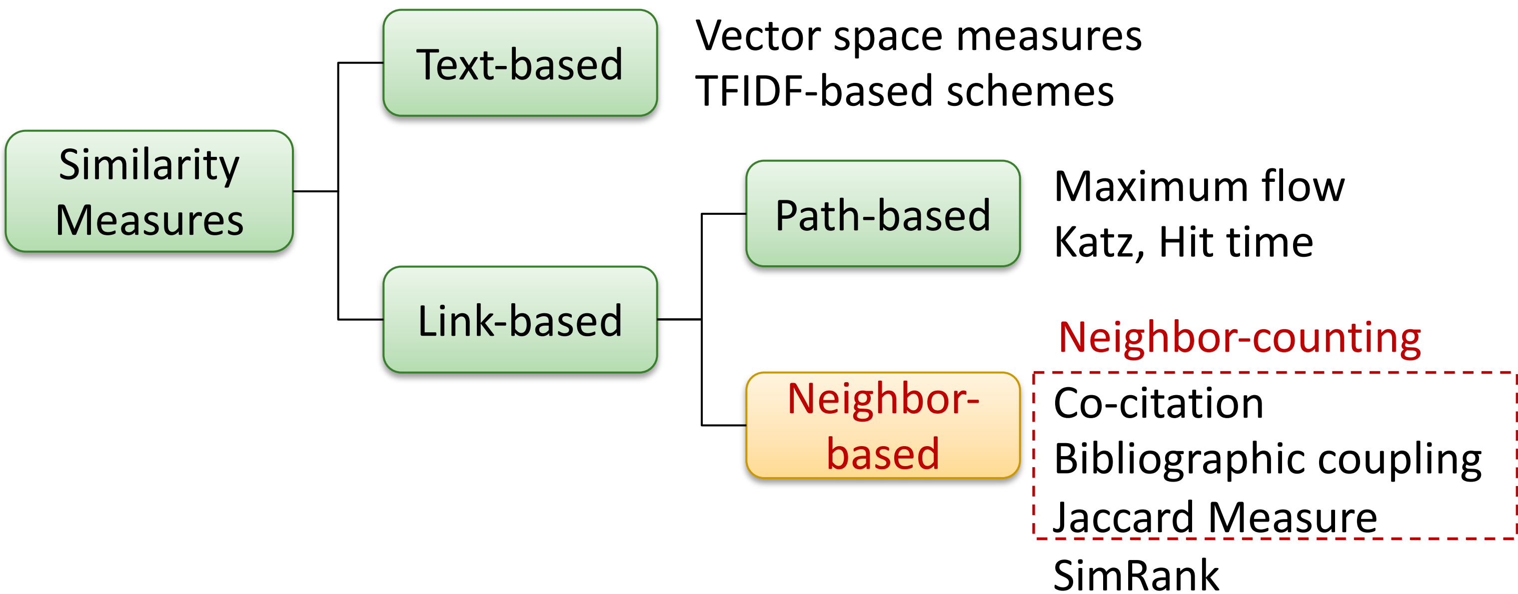
Link-based Similarity Measurement

- The Problem
  - Measuring similarity between objects in a graph
  - Very common & important
  - Arises in many popular applications and domains
    - Web Applications
    - Research Analytics
    - Social Networks





Link-based Similarity Measurement



- Current neighbor-based methods
  - Neighbor-counting: fast and easy to implement, but inflexible
  - SimRank: flexible, but counter-intuitive

Link-based Similarity Measurement

- Our solutions: making better use of neighborhood structure
  - MatchSim algorithm [CIKM'09, KAIS 2011]
    - 1. Takes similarity between neighbors into account
    - 2. Measures similarities based on maximum neighborhood matching
    - Advantages: more flexible and accurate
  - PageSim algorithm [WWW'06 poster, WI'06]
    - 1. Relaxes 1-hop neighbor-counting to multi-hop by using object feature propagation strategy
    - 2. Takes indirect neighbors into account
    - Advantages: more flexible and accurate, efficient
  - ENS (Extended Neighborhood Structure) model [WI'07]
    - 1. can help neighbor-based methods make better use of neighborhood structure
    - 2. extends 1-hop & 1-directional methods to multi-hop & bi-directional
    - Advantages : accuracy improved

Top-*N* Recommendation Problem

- Top-*N* Recommendation Problem
  - Given the preference information of users, recommend a set of *N* items to a certain user that he might be interested in, based on the items he has selected.
    - E-commerce system example: [Amazon. COM](), customers vs. products.

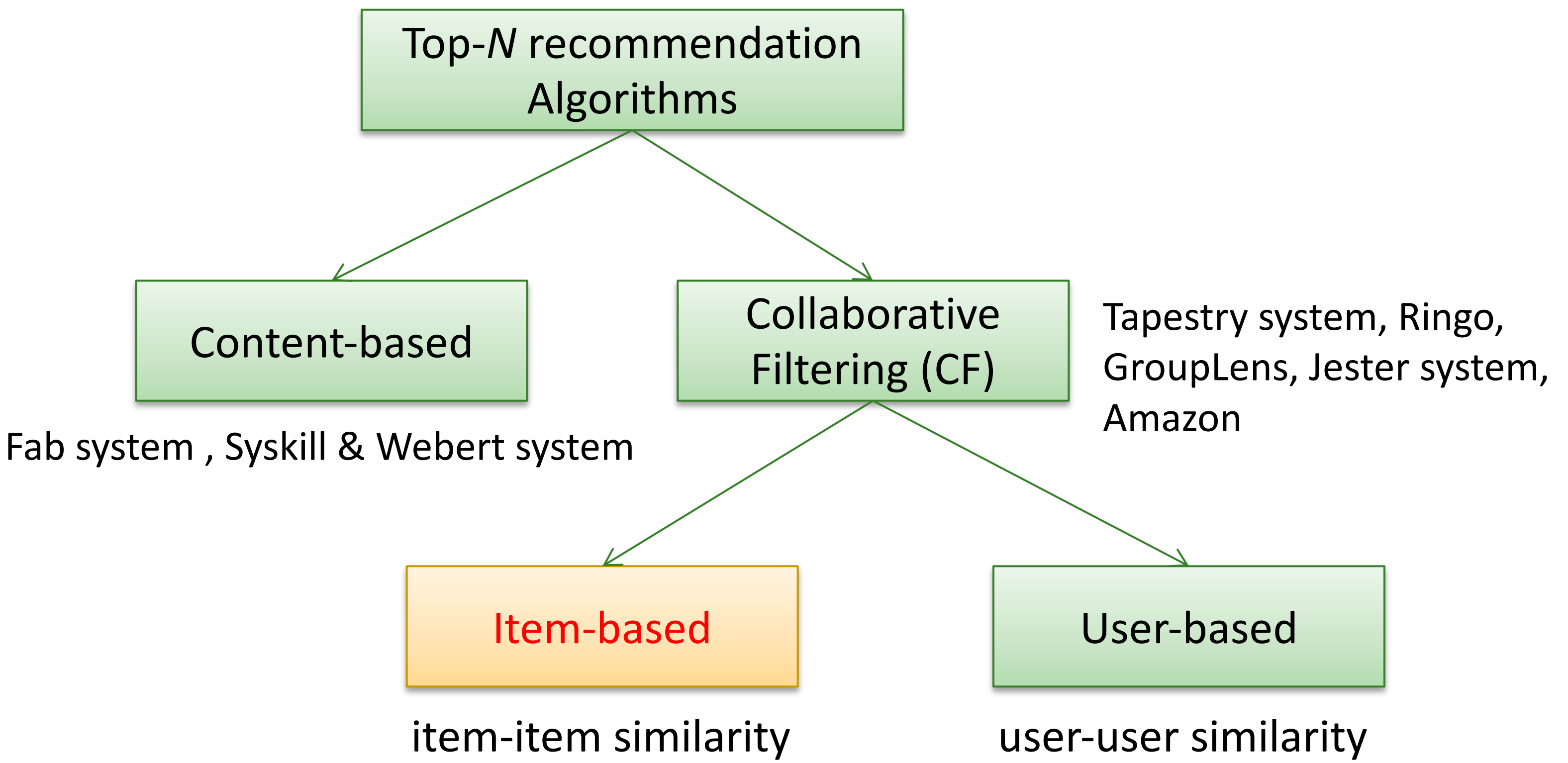
User-Item matrix

	Item 1	Item 2	Item 3	…	Item *m*
User 1	1	0	1		0
User 2	1	1	0		0
…					
User *n*	0	1	0		1
User *n*+1	1	?	1	?	?

Active User

Basket

Top-*N* Recommendation Problem



Top-*N* Recommendation Problem

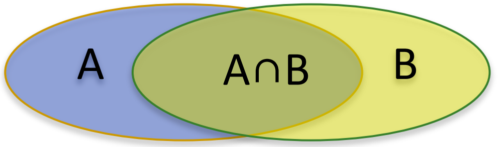
- Classical item-based top-N recommendation algorithms
  - Cosine(COS)-based
  - Conditional-Probability(CP)-based
- Motivation
  - CP-based method considers only the "1-item" probabilities; some useful information may be lost
- Contribution
  - Propose GCP (Generalized Conditional Probability) method, which generalizes CP-based method to a "multi-item"–based version.
  - Advantages: more accurate

Part 1. MatchSim: Similarity Measure Based on Maximum Neighborhood Matching

- 1. Introduction
  - Motivation
  - Contribution
- 2. MatchSim
  - Definition & Computation
  - Complexity & Accelerating Techniques
- 3. Experimental Results
  - Evaluation of Accelerating Techniques
  - Evaluation of MatchSim
- 4. Summary

1. Introduction

- Motivations
  - **Neighbor-counting**: “hard overlapping”, inflexible for large & sparse graphs, poor accuracy
  - **SimRank**: “soft overlapping”, but has a counter-intuitive loophole
- Key Ideas of new solution



  - Consider similarity between neighbors
  - Avoid problem of SimRank by conforming to the “basic intuitions of similarity” [Lin, 1998]

Contributions

- Contributions
  - **Propose** MatchSim
    - based on maximum neighborhood matching
    - flexible and consistent
  - **Prove** the convergence of MatchSim iteration
  - **Design** accelerating techniques
    - Using a *pruning strategy*
    - Adopting an *approximation algorithm.*
  - **Verify** performance on real-world datasets

Neighbor-counting Algorithms

- Intuition: the more *common* neighbors and/or the less *different* neighbors, the more similar

Neighbor-counting Algs.	sim(a,b)
Co-citation	$\|I(a) \cap I(b)\|$, # of common inlinks
Bibliographic coupling	$\|O(a) \cap O(b)\|$, # of common outlinks
Jaccard Measure:	$\frac{\|\Gamma(a) \cap \Gamma(b)\|}{\|\Gamma(a) \cup \Gamma(b)\|}$, Γ can be either I or O.

- Pros: easy to implement & fast
- Cons: inflexible (in large & sparse graphs, the chance that objects have common neighbors is very small.)

SimRank Algorithm

- Intuition: similar pages linked to by similar pages.
- Definition

$$\mathrm{sim}(a,b) = \gamma \cdot \frac{\sum_{u \in I(a)} \sum_{v \in I(b)} \mathrm{sim}(u,v)}{|I(a)| \cdot |I(b)|}, \gamma \in [0,1] \text{ is a constant.}$$

When $|I(a)| \cdot |I(b)| = 0$, $\mathrm{sim}(a,b)=0$ by definition.

- Iterative computation
  - **Initial values**: $\mathrm{sim}(a,b)=1$ if $a=b$, or 0 otherwise.
  - **Iterations**: $\mathrm{sim}(a,b) = \lim_{k \to \infty} \mathrm{sim}_k(a,b)$
- Pros: flexible (considering similarities between neighbors)
- Cons: counter-intuitive

2. MatchSim Algorithm

- Intuition: similar pages have similar neighbors
- Definition:

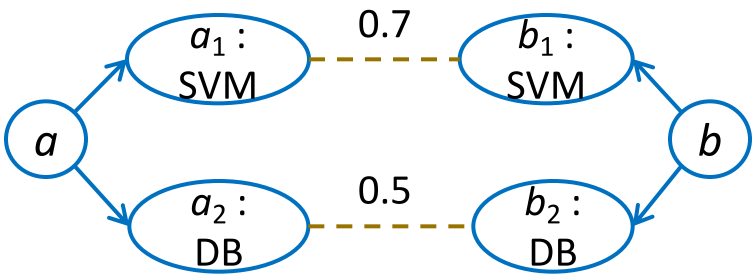
$$\mathrm{sim}(a,b) = \frac{W(a,b)}{\max(|I(a)|\cdot|I(b)|)}, \quad W(a,b) = \sum_{(u,v)\in m^*_{ab}} \mathrm{sim}(u,v)$$

When $|I(a)|\cdot|I(b)|=0$, $\mathrm{sim}(a,b)=0$ by definition

m^*_{ab}: *maximum matching* of similar neighbor-pairs

- Iterative computation (is proved to be convergent)
  - $\mathrm{sim}_0(a,b)=1$ if $a=b$, or 0 otherwise
  - $\mathrm{sim}(a,b) = \lim_{k\to\infty}\mathrm{sim}_k(a,b)$
- Finding maximum matching m^*_{ab}
  - Modeled by *assignment problem*, solved by *Kuhn-Munkers algorithm*.

Examples: SimRank Calculates sim(*a*,*b*)

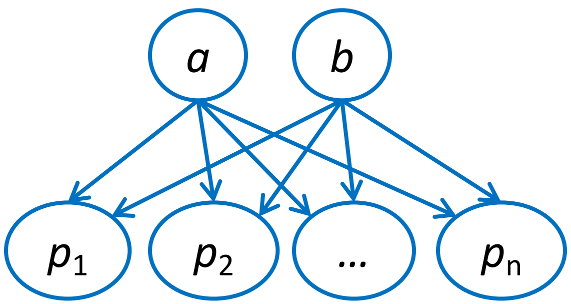


Eg. 1 Only Similar Neighbors
sim(a_1, b_2)=sim(a_2, b_1)=0

Counting	0 (no common neighbors)
SimRank	$\gamma \cdot \sum_{i=1,2} \sum_{j=1,2} sim(a_i, b_j)/4 = 0.3\gamma > 0,$

counter-intuitive!

$$\gamma \times sim(a_2, b_2)/1 = 0.5\gamma,$$



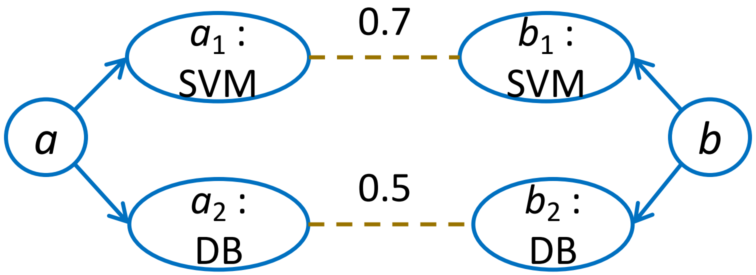
Eg. 2 Many Common Neighbors, sim(p_i, p_j)=0 if $i \neq j$

SimRank	$\gamma \cdot \frac{\sum_{i=1,n} \sum_{j=1,n} sim(p_i, p_j)}{n \cdot n}$

$$= \gamma \cdot \frac{n}{n \cdot n} = 0 (n \to \infty)$$

wrong!

Examples: MatchSim Calculates sim(a,b)

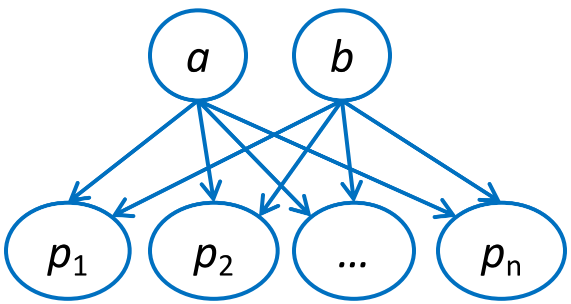


Eg. 1 Only Similar Neighbors
sim(a_1, b_2)=sim(a_2, b_1)=0

Before	$(\mathrm{sim}(a_1, b_1)+\mathrm{sim}(a_2, b_2))/2=0.6$

Reasonable

After	$\mathrm{sim}(a_2, b_2)/1=0.5$



Eg. 2 Many Common Neighbors, sim(p_i, p_j)=0 if $i \neq j$

MatchSim	$\sum_{i=1,n}\mathrm{sim}(p_i, p_j)/n = n/n = 1$ Correct

The maximum matching is (p_i, p_i), i=1, …, n

MatchSim is flexible and consistent.

Accelerating Techniques

- Time complexity: $O(Kn^2L^3)$, $K \approx 15$
- Space complexity: $O(n^2 + L^2)$
  - K: # of iterations, n: # of objects, L: ave. # of neighbors
- 1. Approximate maximum-matching
  - Adopt the *Path Growing Algorithm* (PGA) [Drake 2003]
  - Time complexity reduces to $O(Kn^2L^2)$
- 2. Pruning strategy
  - Prune *unimportant* neighbors to reduce L
  - Adopt PageRank scheme

3. Experimental Results
Datasets, Groundtruth, and Metrics

Dataset	Description	Groundtruth	Metrics
Google Scholar (GS)	Academic articles crawled from Google Scholar by following "cited by" links	"Related Articles" provided by GS	Precision
CiteSeer & Cora	Academic articles classified by topics	Class labels	Precision, Recall, F score

$$GSprec_{A,N}(v) = \frac{|top_{A,N}(v) \cap related_N(v)|}{|top_{A,N}(v)|}.$$

$$precision_{A,N}(v) = \sum_{v \in V} \frac{|top_{A,N}(v) \cap similar(v)|}{|top_{A,N}(v)|}, \quad recall_{A,N}(v) = \sum_{v \in V} \frac{|top_{A,N}(v) \cap similar(v)|}{N},$$

$$Fscore_{A,N}(v) = \sum_{v \in V} (2 \cdot \frac{precision_{A,N}(v) \cdot recall_{A,N}(v)}{precision_{A,N}(v) + recall_{A,N}(v)}).$$

Testing algorithms

- Testing algorithms
  - *CC*: Co-citation,
  - *BC*: Bibliographic Coupling
  - *JM*: Jaccard Measure
  - *SR*: SimRank (γ=0.8)
  - *MS*: MatchSim,
  - MS_{AF}:
    - A – approximate maximum matching,
    - *F* – pruning parameter (maximum number of neighbors)
- Evaluation method
  - Average scores of all objects' results at rank *N* ($1 \leq N \leq 20$)

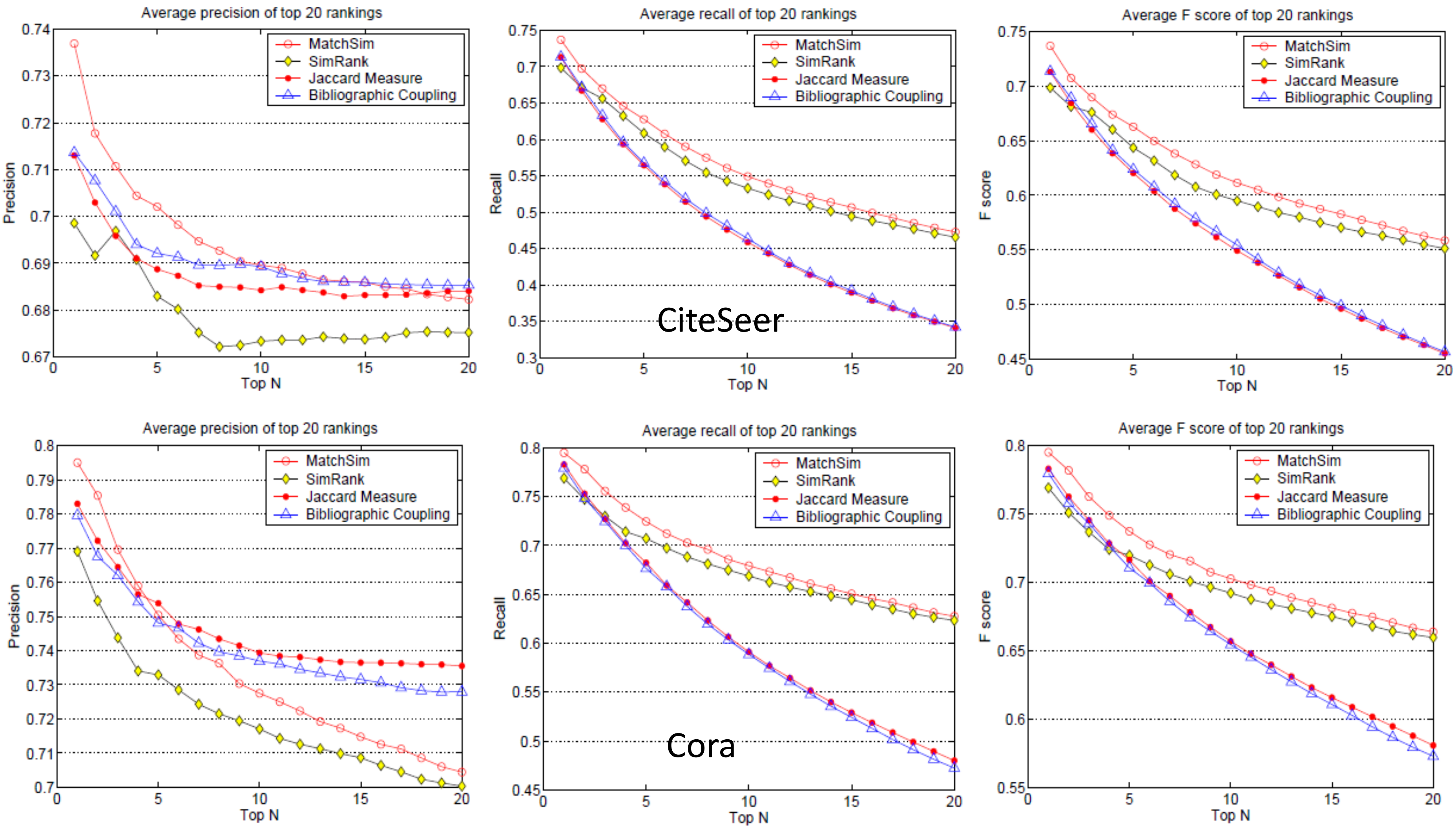
Accelerating Techniques: GS Dataset

F		10	20	30	40	∞
$P(\%)$		7.65	4.07	2.73	1.94	0.00
MS_F	$DA(10^{-2})$	12.44	6.06	3.34	1.42	0.00
	$ROA(\%)$	87.64	94.09	96.78	98.82	100
	$RRT(\%)$	4.81	8.24	11.88	15.86	100
MS_{AF}	$DA(10^{-2})$	11.88	6.00	2.89	1.21	0.94
	$ROA(\%)$	88.10	94.06	97.16	98.90	99.54
	$RRT(\%)$	1.81	2.35	2.76	3.13	6.50

1. *MS* as benchmark
2. Greater *ROA*: more close to *MS*
3. Smaller *RRT*: more time saved

- Observations
  - Pruning parameter *F* ↑, accuracy ↑, running time ↑
  - MS_{AF} uses much less time with small loss of accuracy.
- The best version is MS_{A40}
  - Overall accuracy is 98.9% close to MS.
  - Running time is greatly reduced to 3.13% compared to MS.

Performance on CiteSeer and Cora



4. Summary of Part 1

- Contributions
  - **Propose** MatchSim: neighbor-based similarity measure based on maximum neighborhood matching
  - **Prove** the convergence of MatchSim computation
  - **Design** accelerating techniques including using a *pruning strategy* and an *approximation algorithm*
  - **Verify** performance experimentally on real-world datasets

Part 2. PageSim: Similarity Measure Based on Feature Propagation of Objects

- 1. Introduction
  - Motivations
  - Contributions
- 2. PageSim
  - Feature Propagation & Feature Comparison
  - An Example
- 3. Experimental Results
  - Evaluation of PageSim
- 4. Summary

1. Introduction

- Motivations
  - Neighbor-counting methods only consider direct neighbors.
  - Ignore importance of objects.
- Intuitions
  - Links as recommendations (can propagate to neighbors)
  - Strength of recommendations decrease along links
  - Authoritative objects are more important & trustworthy
- Contributions
  - **Propose** PageSim - a *multi-hop* and *fuzzy* Jaccard Measure
  - **Verify** performance of PageSim experimentally on real-world datasets

2. PageSim

- Key Ideas of PageSim
  - Consider the impacts of *indirect* neighbors
  - Adopt *PR scores* to represent the importance of objects
  - Relax **Jaccard Measure** to a *multi-hop* and *fuzzy* version.
- Two phases in PageSim
  - Phase 1: object feature propagation
  - Phase 2: object feature comparison

Phase 1: Feature Propagation

- Each object has its unique feature information (*PR* scores).
- Feature information of objects are propagated along out-links at decay rate *d*.
- The *PR* scores of *u* that are propagated to *v* is defined by

$$PG(u,v) = \begin{cases} \sum_{p \in PATH(u,v)} \frac{d \cdot PR(u)}{\prod_{w \in p, w \neq v} |O(w)|}, & v \neq u, \\ PR(u) & v = u, \end{cases}$$

- **Note**: if we define *PG*(*u*,*u*) = 0, we get the basic version of PageSim, denoted by $\textbf{PageSim}_{\boldsymbol{B}}$.

Phase 2: Feature Comparison

- Features are saved in *Feature Vectors*.

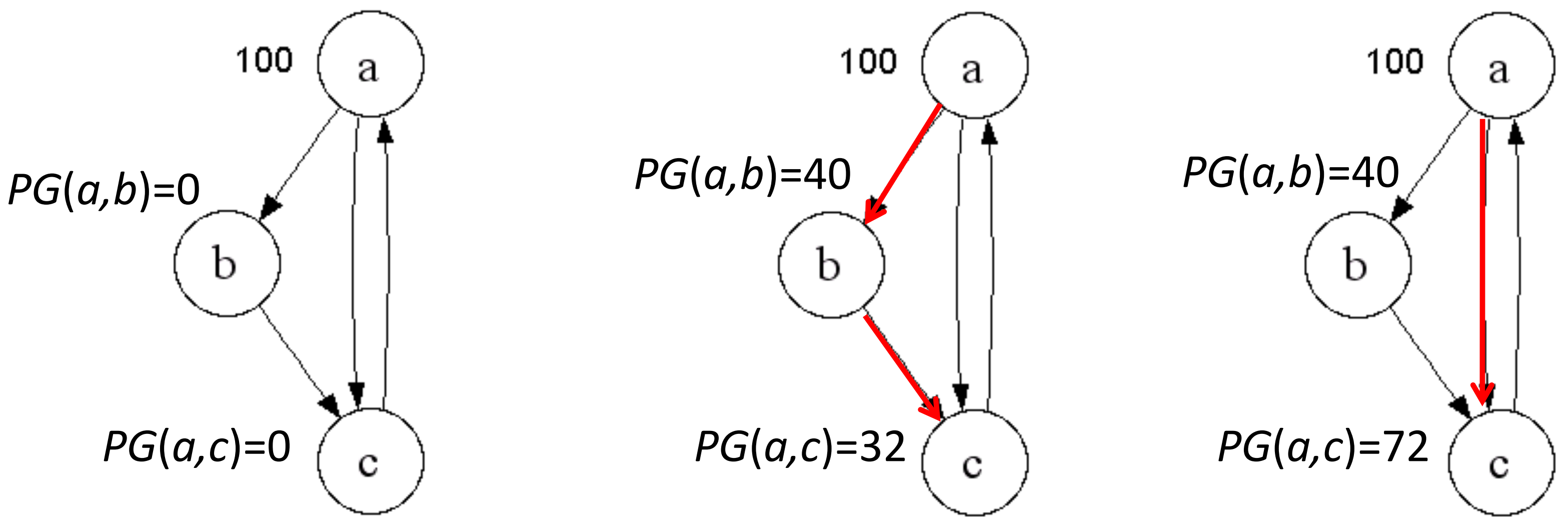
$$\boxed{\overrightarrow{FV}(v) = (PG(v_i, v))^T, i = 1, \cdots, n,}$$

- The PageSim score between objects *u* and *v* is computed by applying Jaccard Measure

$$\boxed{PS(u, v) = \frac{\sum_{i=1}^{n} min(PG(v_i, u), PG(v_i, v))}{\sum_{i=1}^{n} max(PG(v_i, u), PG(v_i, v))}}$$

Example: Feature Propagation Phase

- *PR*(*a*)=100, *PR*(*b*)=55, *PR*(*c*)=102, *d* = 0.8
- A *DFS*-like propagation procedure

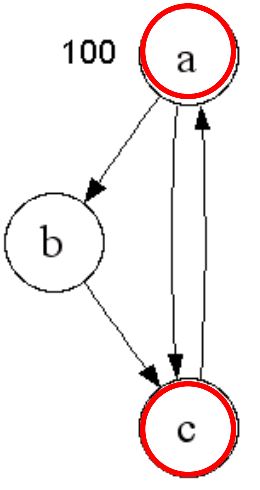


At the beginning | *Path: a →b →c* | *Path: a →c*

Propagating *PR*(*a*)

Example: Feature Comparison Phase

- *PR*(*a*)=100, *PR*(*b*)=55, *PR*(*c*)=102
- Feature vectors
  - *FV*(*a*) = (100, 35, 82)
  - *FV*(*b*) = (40, 55, 33)
  - *FV*(*c*) = (72, 44, 102)
- PageSim scores
  - *PS* (*a*,*b*) = (40+35+33) / (100+55+82) = 0.46
  - *PS* (*a*,*c*) = (72+35+82) / (100+44+102) = 0.77
  - *PS* (*b*,*c*) = (40+44+33) / (72+55+102) = 0.51

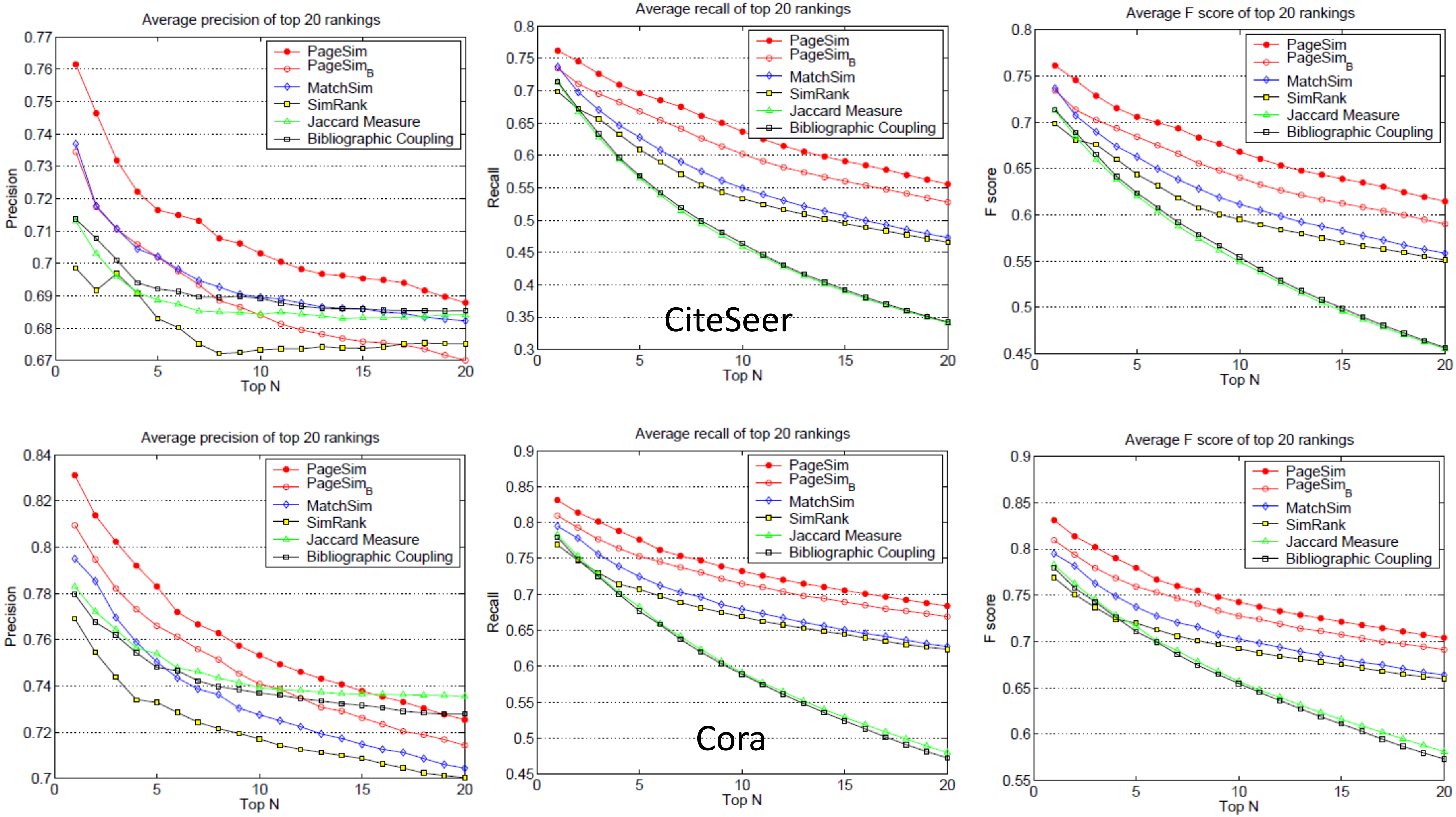


$$PS(u,v) = \frac{\sum_{i=1}^{n} min(PG(v_i,u), PG(v_i,v))}{\sum_{i=1}^{n} max(PG(v_i,u), PG(v_i,v))}$$

3. Experimental Results

- Datasets
  - CiteSeer
  - Cora
- Testing algorithms
  - *CC*: Co-citation
  - *BC*: Bibliographic Coupling
  - *JM*: Jaccard Measure
  - *SR*: SimRank (γ=0.8)
  - *PS*: PageSim (d=0.5, r=3)

Performance on CiteSeer and Cora - 1



PageSim is accurate & flexible.

Performance on CiteSeer and Cora - 2

- Runtime (in second) on CiteSeer and Cora datasets
  - PageSim is efficient.

	BC	CC	JM	SR	MS	PS	PS_B
CiteSeer	171	132	174	1,632	1,680	185	182
Cora	99	97	99	1,515	1,275	116	113

4. Summary of Part 2

- PageSim
  - Taking the *indirect* neighbors into account
  - Feature *propagation* and feature *comparison*
  - A multi-hop and fuzzy version of Jaccard Measure
  - More flexible and accurate
  - Experiments on real-world datasets

Part 3. ENS: Extended Neighborhood Structure Model

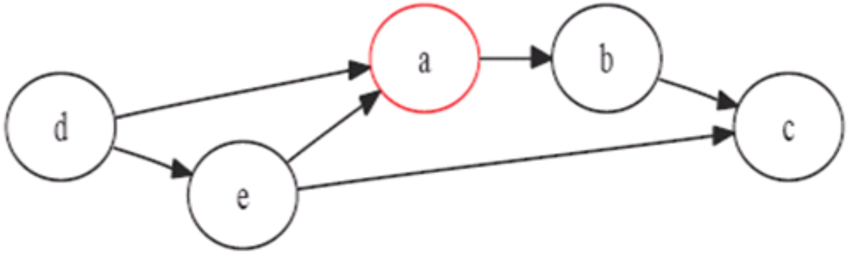
- 1. Introduction
  - Motivation
  - Contribution
- 2. The ENS Model
- 3. Extending Link-based Similarity Measures
  - Neighbor-counting Algorithms
  - PageSim & SimRank
- 4. Experimental Results
- 5. Summary

1. Introduction

- Motivation
  - How to improve accuracy by making better use of the structural information?
- Contributions
  - **Propose** Extended Neighborhood Structure (ENS) model
    - bi-directional
    - multi-hop
  - **Extend** link-based similarity measures base on ENS model
    - more flexible and accurate

2. The ENS Model

- Extended Neighborhood Structure (ENS) model
  - The ENS model
    - bi-direction
      - *in-link* & *out-link*
    - multi-hop
      - *direct* (1-hop) : *d* is *a*'s direct inlinnk neighbor
      - *indirect* (2-hop, 3-hop, etc): *c* is *a*'s indirect outlink neighbor



  - Purpose
    - **Improve** accuracy of link-based similarity measures by helping them make better use of the structural information

3. Extending Link-based Similarity Measures

- Two classical methods (1-directional)
  - **Co-citation**: the more common <u>in-link</u> neighbors, the more similar.
    - $sim(a,b) = |I(a) \cap I(b)|$
  - **Bibliographic coupling**: the more common <u>out-link</u> neighbors, the more similar.
    - $sim(a,b) = |O(a) \cap O(b)|$
- Extended Co-citation and Bibliographic Coupling (ECBC)
  - **ECBC**: The more common neighbors, the more similar.
    - $sim(a,b) = \alpha|I(a) \cap I(b)| + (1-\alpha)|O(a) \cap O(b)|$, bi-directional
      where $\alpha \notin [0,1]$ is a constant.

3. Extending Link-based Similarity Measures

- Extended SimRank

  "two pages are similar if they have similar neighbors"

  - (1) sim(*u*,*u*)=1; (2) sim(*u*,*v*)=0 if |I(*u*)| |I(*v*)| = 0.

    Recursive definition

$$sim(a,b) = \gamma \frac{\sum_{u \in I(a)} \sum_{v \in I(b)} sim(u,v) + \sum_{u \in O(a)} \sum_{v \in O(b)} sim(u,v)}{|I(u)||I(v)| + |O(u)||O(v)|}$$

  - *C* is a constant between 0 and 1.
  - The iteration starts with sim(*u*,*u*)=1, sim(*u*,*v*)=0 if *u*≠ *v*.

$$sim(a,b) = lim_{k \to \infty} sim_k(a,b)$$

3. Extending Link-based Similarity Measures

- PageSim

  "weighted multi-hop" version of Jaccard Measure

  - (a) multi-hop in-link information, and
  - (b) importance of objects.
    - Can be represented by any global scoring system
      - PageRank scores, or
      - Authoritative scores of HITS.

3. Extending Link-based Similarity Measures

- Extended PageSim (EPS)
  - Propagate feature information of objects along in-link hyperlinks at decay rate 1- *d.*
  - Obtain the in-link PS scores.
  - EPS(a,b) = in-link PS(a,b) + out-link PS(a,b).

3. Extending Link-based Similarity Measures

Properties

Table 1: Properties of the Algorithms

Properties	CC	BC	ECBC	SR	ESR	PS	EPS
bi-direction	-	-	+	-	+	-	+
multi-hop	-	-	-	+	+	+	+

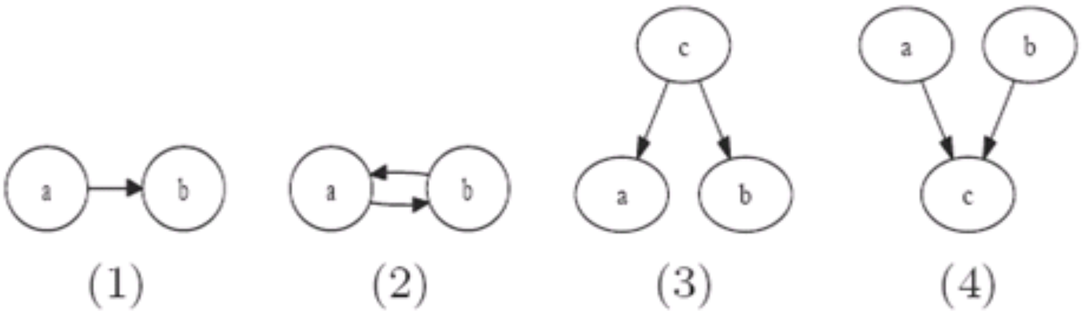
- CC: Co-citation, BC: Bibliographic Coupling
- ECBC: Extended CC and BC
- SR: SimRank, ESR: Extended SR
- PS: PageSim, EPS: Extended PS

Summary

- The extended versions consider more structural information.
- ESR and EPS are bi-directional & multi-hop.

3. Extending Link-based Similarity Measures

- Case study: sim(*a*,*b*)



Case	CC	BC	ECBC	SR	ESR	PS	EPS
1	-	-	-	-	-	+	+
2	-	-	-	-	-	+	+
3	+	-	+	+	+	+	+
4	-	+	+	-	+	-	+

- Summary
  - The extended algorithms are more *flexible*.
  - *EPS* is able to deal with all cases.

4. Experimental Results

- Dataset

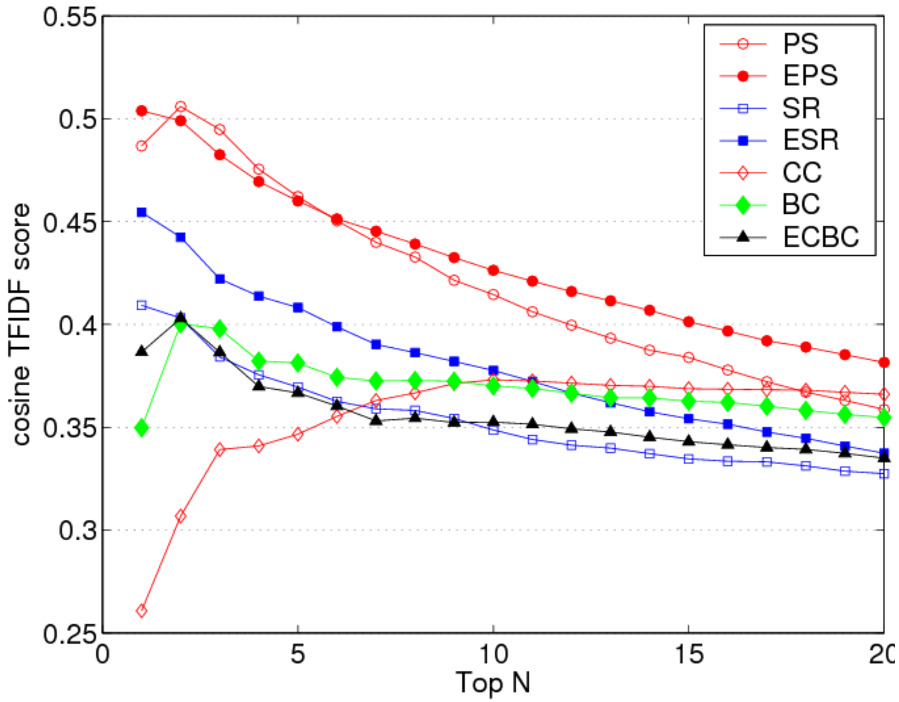
Dataset	Description	Groundtruth	Metrics
CSE Web (CW)	Web pages crawled from http://cse.cuhk.edu.hk	Textual similarity	Cosine TFIDF

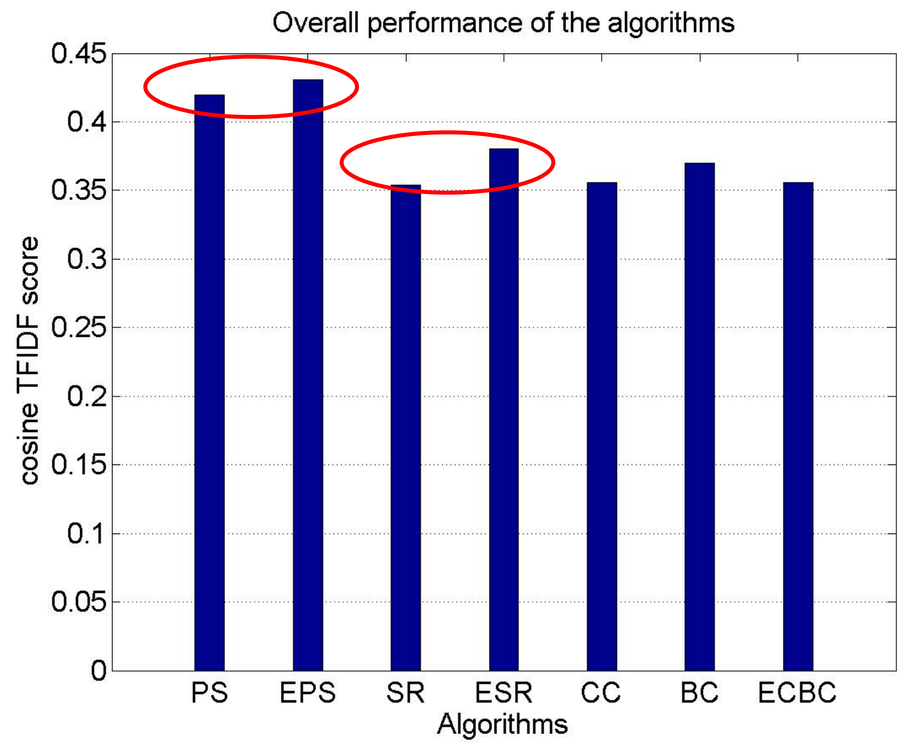
- Evaluation metric

$$cosTFIDF(u,v) = \frac{\sum_{t \in u \cap v} W_{tu} \cdot W_{tv}}{\|u\| \cdot \|v\|},$$

$$\|u\| = \sqrt{\sum_{t \in u} W_{tu}^2} \text{ and } \|v\| = \sqrt{\sum_{t \in v} W_{tv}^2}.$$

Performance Evaluation (CW Dataset)





ECBC	SR	ESR	PS	EPS
$\alpha = 0.5$	$\gamma = 0.8$	$\gamma = 0.8$	$r = 3,\ d = 0.5$	$r = 3,\ d = 0.7$

- *ENS* works well on *PS* and *SR*.
- *ECBC* are worse than *CC* and *BC*.

5. Summary of Part 3

- ENS model
  - bi-directional (inlink and outlink)
  - multi-hop neighborhood structure
- Extend link-based methods
  - PageSim, SimRank, Co-citation, and Bibliographic coupling to EPS, ESR, ECBC algorithms
  - Accuracy improved

Part 4. Top-*N* Recommendation Algorithm Based on Item-Graph

- 1. Introduction
  - Motivations
  - Contributions
- 2. The GCP-based Method
  - Generalized Conditional Probability (GCP) Algorithm
- 3. Experimental Results
- 4. Summary

1. Introduction

- Motivation
  - CP-based method considers only the “1-item” probabilities; some useful information may be lost.
- Contributions
  - Propose GCP (Generalized Conditional Probability) method
  - Advantages: more accurate

1. Introduction

- Notations
  - Item set $I = \{I_1, I_2, \ldots, I_m\}$.
  - User set $U = \{U_1, U_2, \ldots, U_n\}$.
  - User-Item matrix $D = (D_{n,m})$.
  - Basket of the active user $B \in I$.
  - Similarity score of x and y: $\mathrm{sim}(x, y)$.
- Formal definition of top-*N* recommendation problem
  - Given a user-item matrix ***D*** and a set of items ***B*** that have been purchased by the *active user*, identify an ordered set of items ***X*** such that $|\boldsymbol{X}| \leq \boldsymbol{N}$, and $\boldsymbol{X} \cap \boldsymbol{B} = \emptyset$.

1. Introduction

- Two classical item-item similarity measures
  - ***Cosine-based*** (symmetric)

$$\mathrm{sim}(I_i, I_j) = \cos(D_{*,i}, D_{*,j})$$

  - ***Conditional probability(CP)-based*** (asymmetric)

$$\mathrm{sim}(I_i, I_j) = P(I_j \mid I_i) \approx \mathrm{Freq}(I_i\, I_j) \,/\, \mathrm{Freq}(I_i)$$

Freq(X): the number of customers that have purchased the items in the set X.

- Recommendation strength (ranking score) of item x is

$$RS(x) = \sum_{b \in B} \mathrm{sim}(b, x)$$

2. The GCP-based Method

- The GCP-based recommendation algorithm
  - Define RS(x) by the sum of all "multi-item"-based conditional probabilities

$$\mathrm{GCP}(x|B) = \sum_{S \in B} \mathrm{P}(x|S) \approx \sum_{S \in B} (\mathrm{Freq}(xS) \,/\, \mathrm{Freq}(S))$$

  - Exponential problem: # of S = $2^{|B|}$
  - Approximate GCP

$$\mathrm{GCP}_d(x|B) = \sum_{S \in B,\ |S| \leq d} \mathrm{P}(x|S)$$

3. Experimental Results

- Dataset
  - The MovieLens (http://www.grouplens.org/data)
    - Multi-valued ratings indicating how much each user liked a particular movie or not
    - Treat the ratings as an indication that the users have seen the movies (nonzero) or not (zero)

# of Users	# of Items	Density[1]	Average Basket Size
943	1682	6.31%	106.04

[1]Density: the percentage of nonzero entries in the user-item matrix.

Evaluation

- Evaluation design
  - Split the dataset into a *training* and *test* set by
    - randomly selecting one rated movie of each user to be part of the test set,
    - use the remaining rated movies for training.
  - Cosine(COS)-based, CP-based, GCP-based methods, 10-runs average.
- Evaluation metrics
  - Hit-Rate (HR)

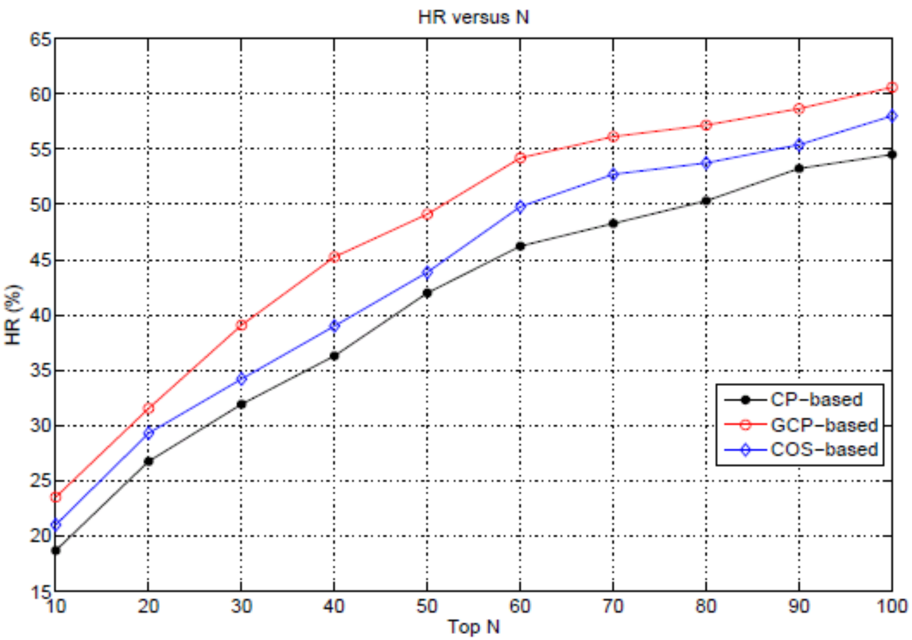
$$HR = \# \text{ of hits} / n$$

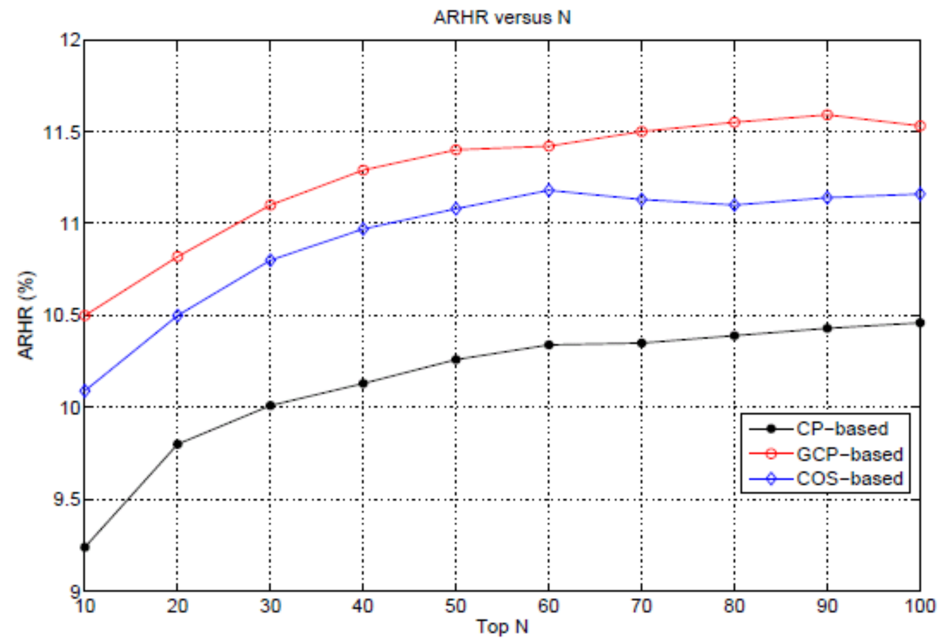
  - Average Reciprocal Hit-Rate (ARHR)

$$ARHR = \left(\sum_{i=1,h} 1/p_i\right) / n$$

\# of hits: the number of items in the test set that were also in the top-*N* lists. h is the number of hits that occurred at positions $p_1, p_2, \ldots, p_h$ within the top-*N* lists (i.e., $1 \le p_i \le N$).

Performance Evaluation





- In GCP method, d = 2

4. Summary of Part 4

- Conclusion
  - Top-*N* recommendation problem & item-centric algorithms
    - Cosine-based, conditional probability-based
- Contribution
  - Generalized Conditional Probability-based top-*N* recommendation algorithm
    - A “multi-item”-based generalization of CP

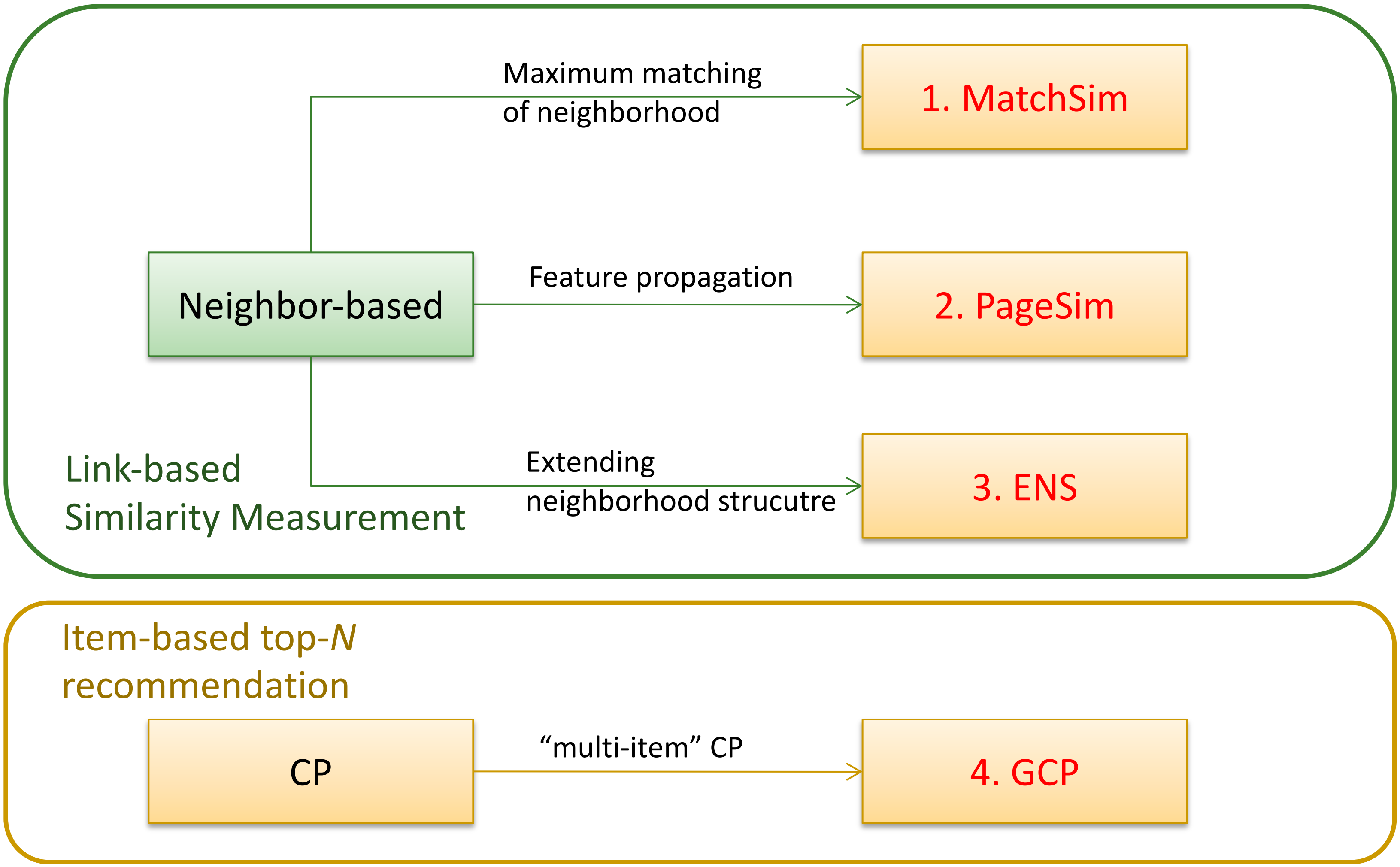
Conclusion

- Technical contributions
  - Two neighbor-based similarity measures
    - MatchSim & PageSim
  - The ENS model and extend link-based similarity measures
  - The GCP-based top-*N* recommendation algorithm
  - Accelerating techniques
- Theoretical contributions
  - Complexity analysis
  - Proof of converge
- Practical contributions
  - ScholarMate: a social network for researchers
  - eGrants: proposal-expert recommendation

Future Work

- Link-based similarity measurement
  - Weight/popularity of objects/links
  - Embedding semantic information on links
- Top-N recommendation
  - Link-based similarity measurement techniques for item-item or user-user similarity computation
    - User-item bipartite graph
    - Item-item correlation graph

Relationships of The Four Parts

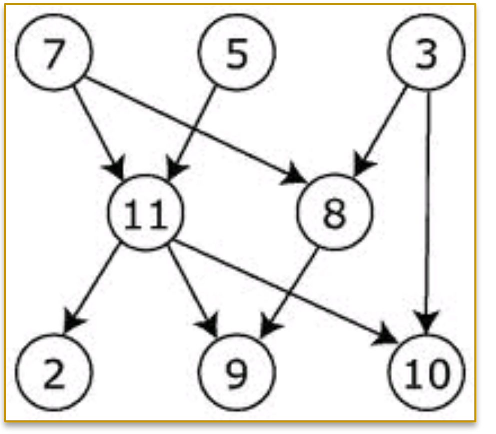


Publication List

1. **Z. Lin**, M. Lyu, and I. King, "MatchSim: A Novel Similarity Measurement Based on Maximum Neighborhood Matching", *Knowl. Inf. Syst.*, 1-26, 2010.
2. **Z. Lin**, M. Lyu, and I. King, "MatchSim: Web Pages Similarity Measurement with Maximum Matching", Conference on Information and Knowledge Management, 1613-1616, 2009.
3. X. Liu, **Z. Lin**, H. Wang, "Two Novel Methods for Time Series Segmentation", *IEEE Trans. on Knowledge and Data Eng*., 20(1616-1626):12, December 2008.
4. **Z. Lin**, M. Lyu, and I. King, "Extending Link-based Algorithms for Similar Web Pages", IEEE/WIC/ACM International Conference on Web Intelligence, 263-266, 2007.
5. **Z. Lin**, I. King, and M. Lyu, "PageSim: A Novel Link-based Similarity Measure", IEEE/WIC/ACM International Conference on Web Intelligence, 687-693, 2006.
6. **Z. Lin**, I. King, and M. Lyu, "PageSim: A Novel Link-Based Measure of Web Page Similarity", International Conference on World Wide Web, poster session, 1019-1020, 2006.

Appendix 1: Intuitions of Similarity

- Basic intuitions of similarity
  - S1. The more commonality, the more similar
  - S2. The more differences, the less similar
  - S3. The maximum similarity is reached when objects are identical
- Basic notations
  - G=(V, E), |V| = *n*: a direct graph of size *n*
  - I(*a*) / O(*a*): in-link / out-link neighbors of object *a*
  - sim(*a*,*b*): similarity score of objects *a* and *b*



- Example graphs
  - Web graph: V – web pages, E – hyperlinks
  - Citation graph: V – scientific articles, E – citations

Appendix 2: Part 1. Statistics of Datasets

	CW	GS	CiteSeer	Cora	
Type of Objects	web page	paper	paper	paper	
Type of Links	hyperlink	citation	citation	citation	
# of Objects	22,615	20,000	2,110	2,485	
# of Links	120,947	87,717	3,757	5,209	
Inlinks/Outlinks per Object	5.3	4.4	1.8	2.1	
inlink dangling nodes (%)	0%	57.7%	39.4%	42.3%	No inlinks
outlink dangling nodes (%)	14.7%	0.06%	24.7%	16.4%	No outlinks

- Dangling nodes are caused by *incompleteness* of datasets.
- Too many dangling nodes can reduce quality of results.
  - For CW dataset, use *inlinks* as default input
  - For others, use *outlinks* as default input

- Distributions of Articles in CiteSeer and Cora Datasets

CiteSeer	# of papers	Cora	# of papers
Agents	463	Case_Based	285
AI	115	Genetic_Algorithms	406
DB	388	Neural_Networks	726
IR	304	Probabilistic_Methods	379
ML	532	Reinforcement_Learning	214
HCI	308	Rule_Learning	131
		Rule_Theory	344
Total	2,110	Total	2,485

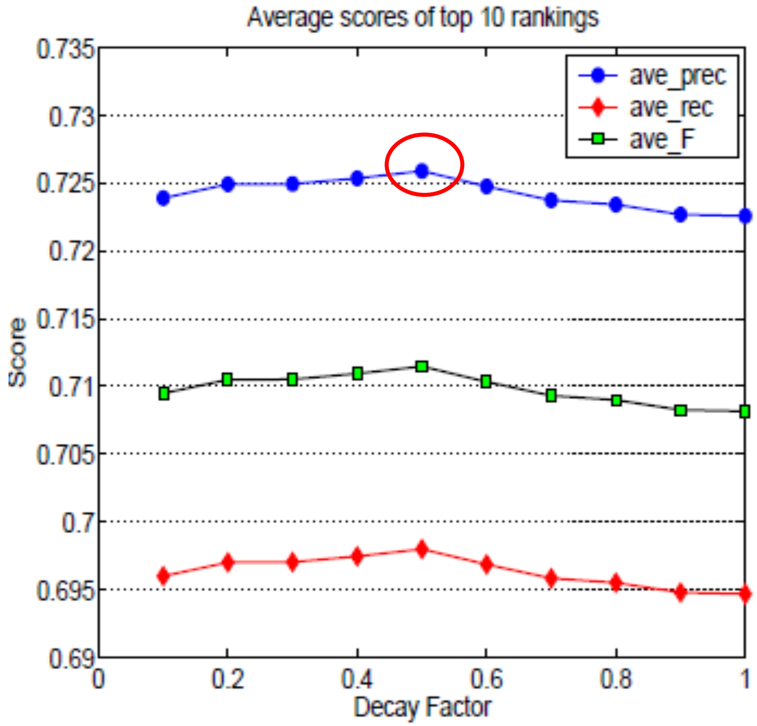
- Testing algorithms
  - *CC*: Co-citation, *BC*: Bibliographic Coupling, *JM*: Jaccard Measure,
  - *SR*: SimRank (γ=0.8), *MS*: MatchSim,
  - MS_{AF}: Approximate MatchSim, *F* – pruning number

Appendix 3: Part 1. Performance on CiteSeer and Cora

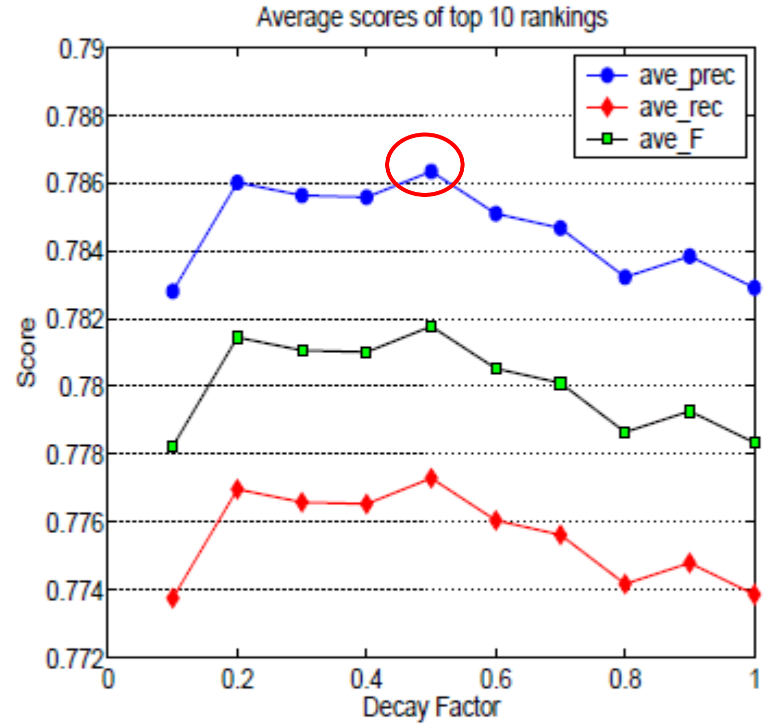
- Running time
  - MatchSim and SimRank are less efficient

	BC	CC	JM	SR	MS
CiteSeer	171	132	174	1,632	1,680
Cora	99	97	99	1,515	1,275

Appendix 4: Part 2. Impact of Decay Factor *d*



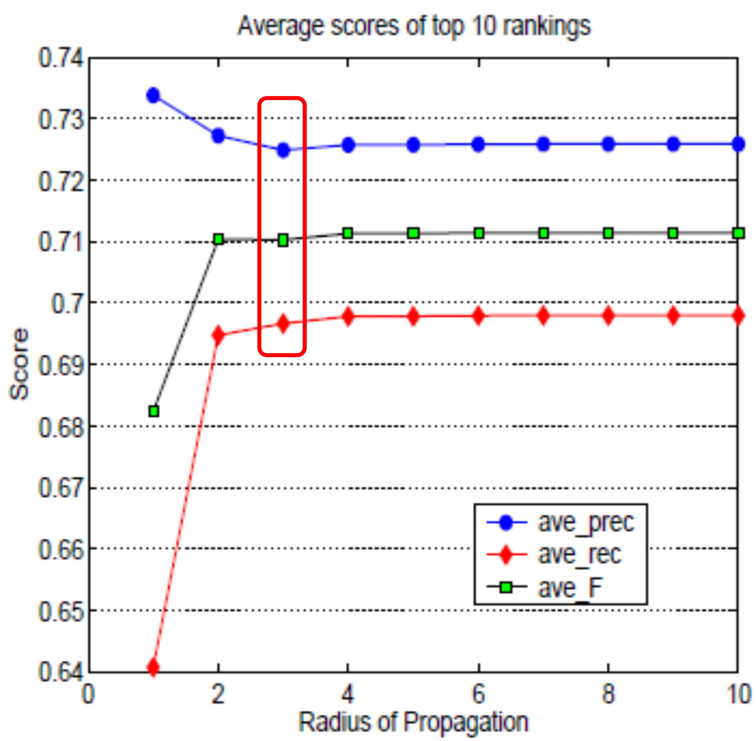
(a) Results on the CiteSeer dataset



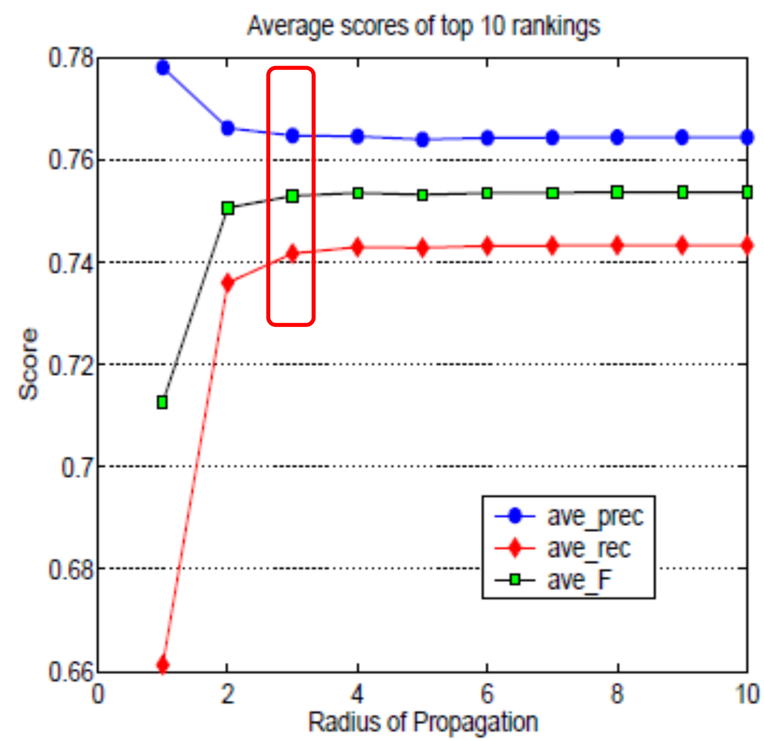
(b) Results on Cora dataset

- (1) the impact of decay factor *d* is not very significant.
- (2) *d* = 0.5 is the best setting for *d* on both datasets.

Appendix 5: Part 2.
Impact of Radius *r* on Effectiveness



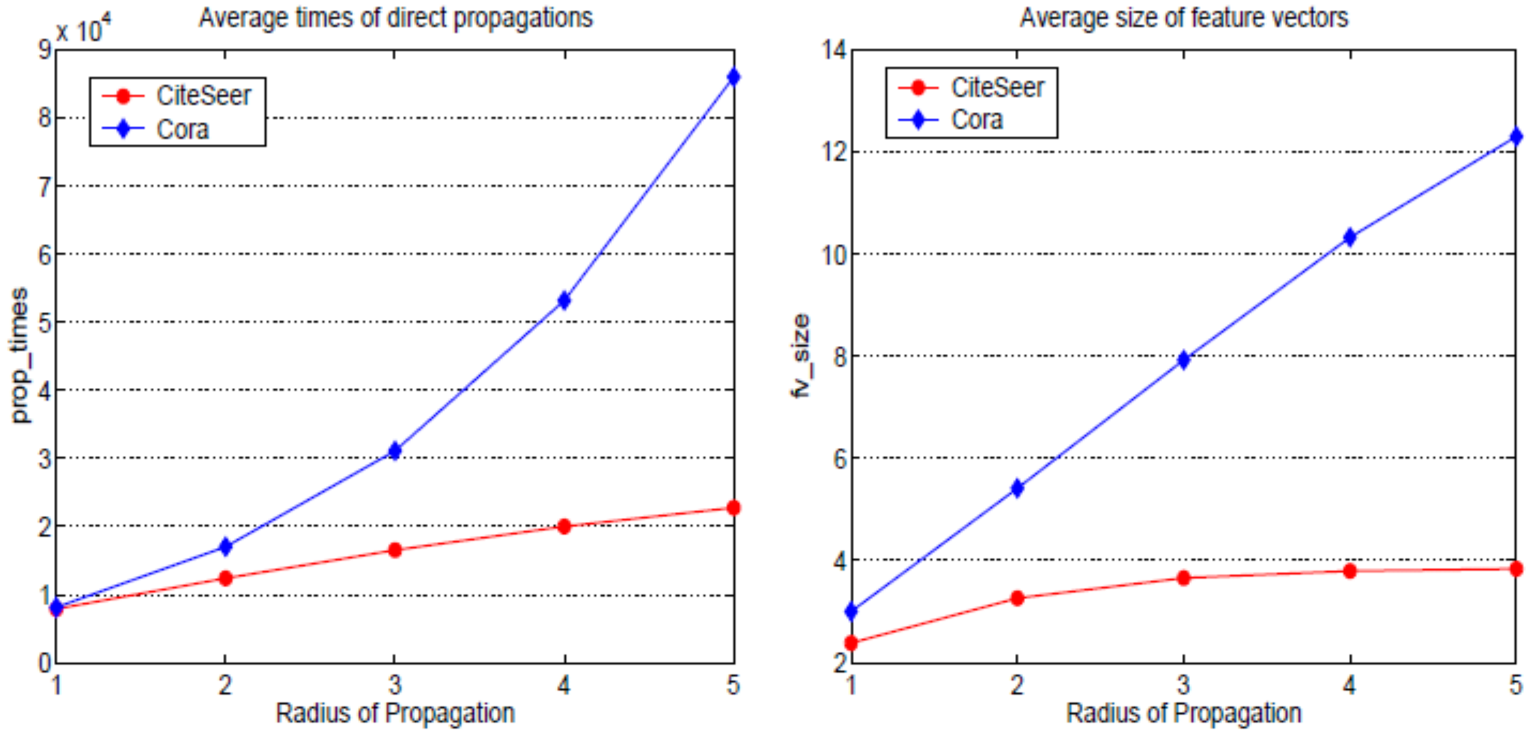
(a) Results on the CiteSeer dataset



(b) Results on Cora dataset

- (1) accuracy does not increase with *r*.
- (2) *r* = 3 is the best setting for *r* on both datasets.

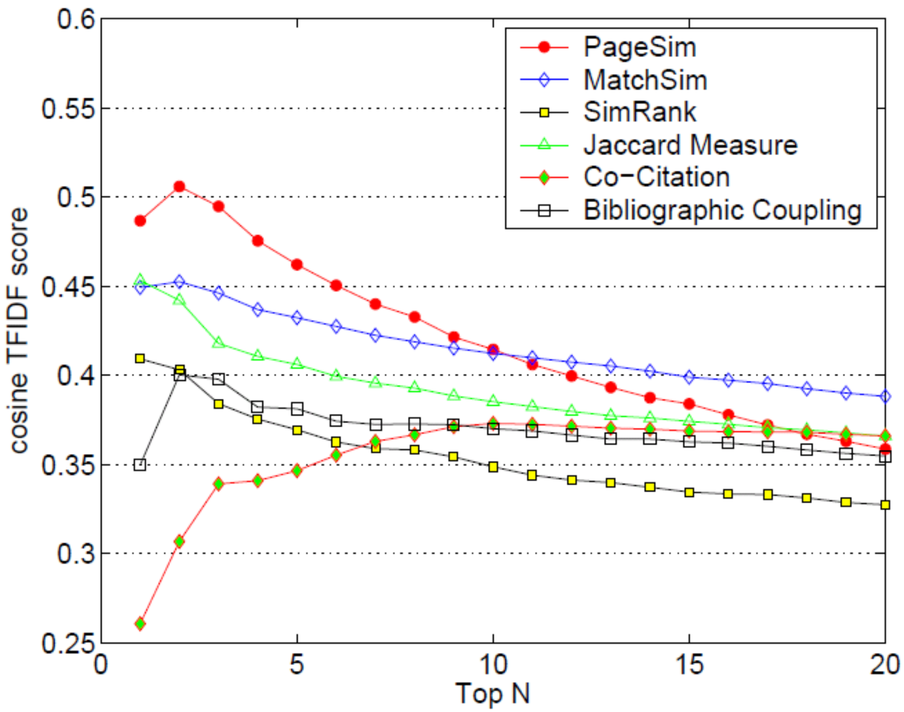
Appendix 6: Part 2.
Impact of Radius *r* on Efficiency



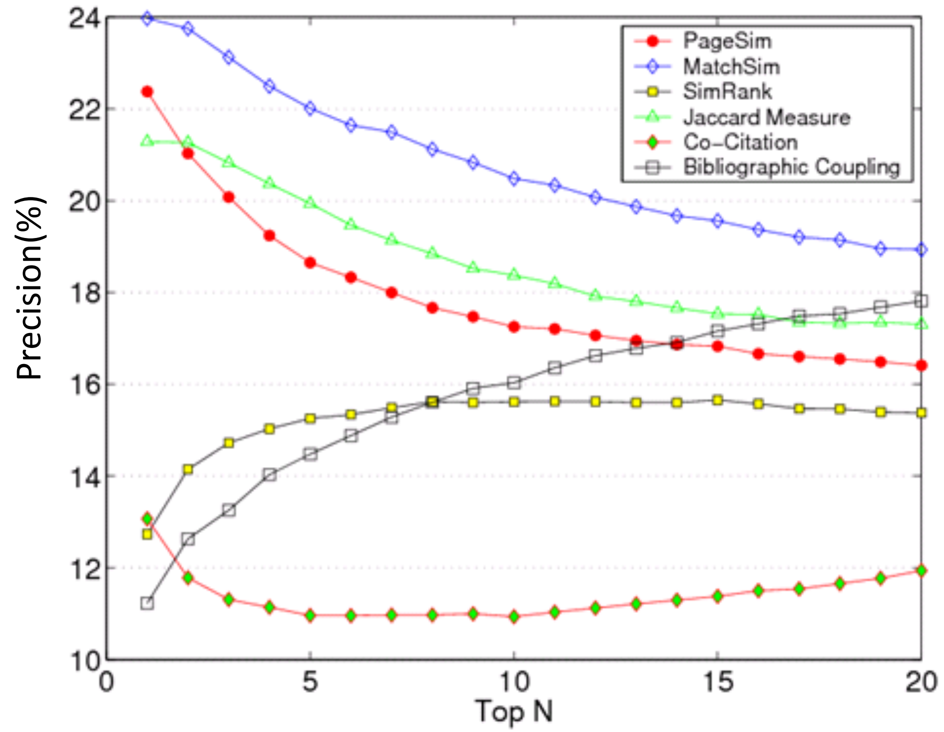
(a) Times of propagation *prop_times* (b) Number of returned objects *ret_num*

- *Prop_times:* the average times of propagations performed in phase 1.
- Radius *r* ↑, running time ↑. Therefore, we choose *r* = 3.

Appendix 7: Part 2.
Performance on CW and GS Datasets



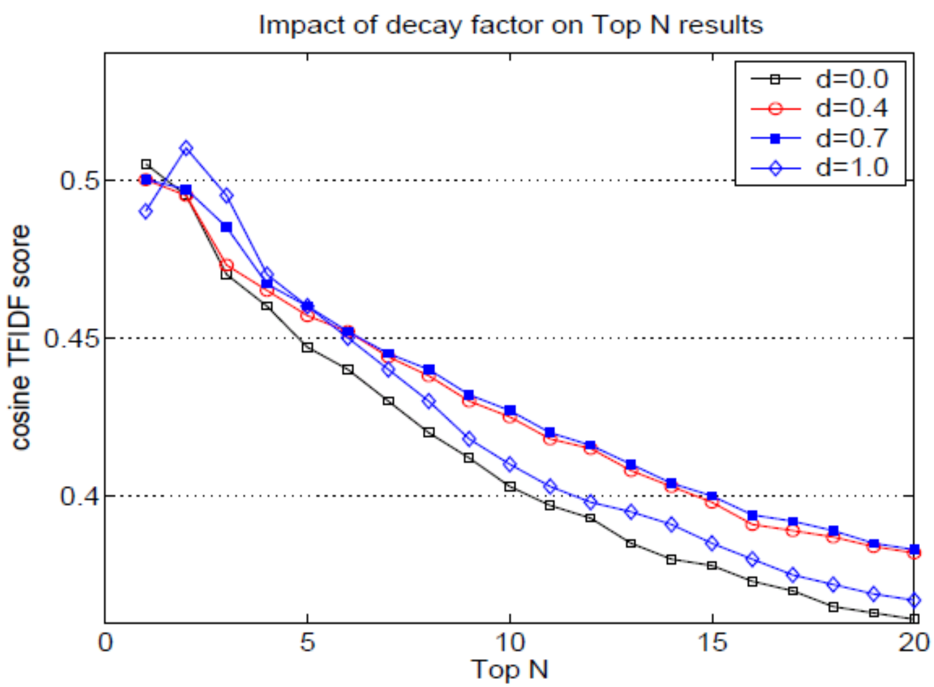
CW dataset

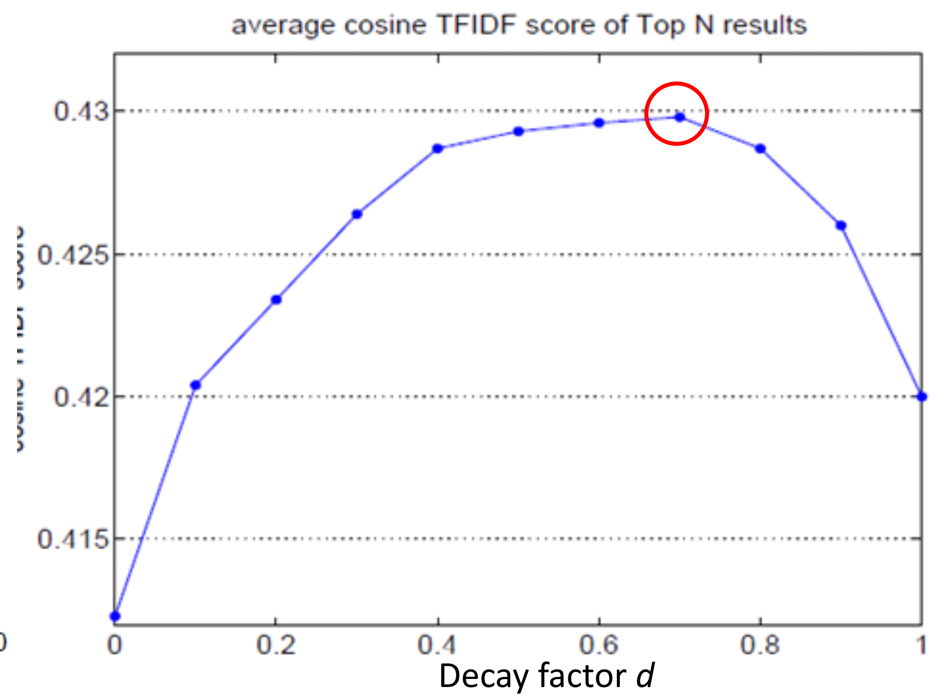


GS dataset

- PageSim works well on CW, but worse than MatchSim.
- JM works better than PageSim on GS, Google Scholar may gives more weights to direct neighbors.

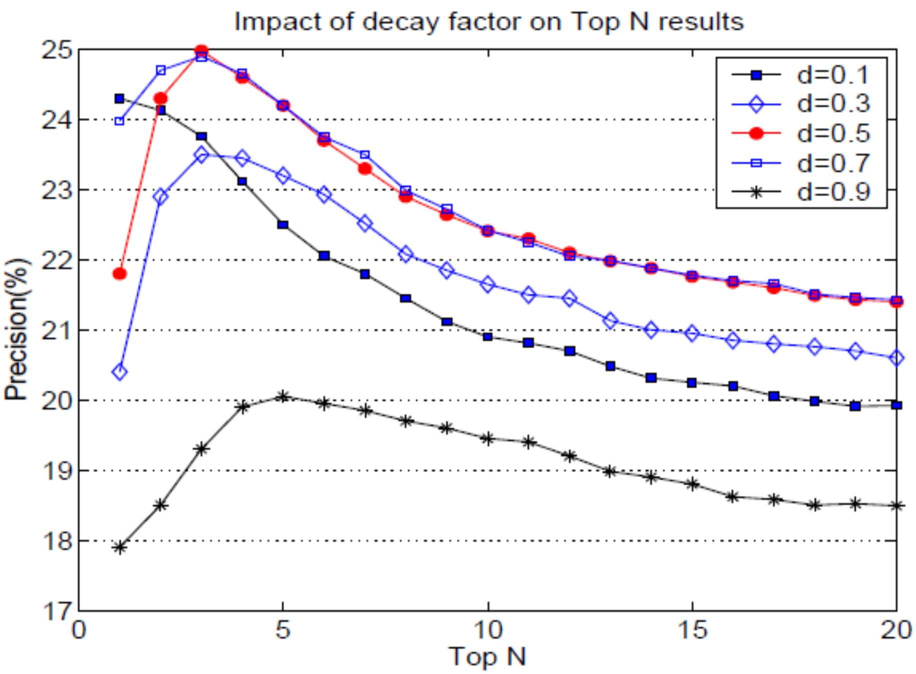
Appendix 8: Part 3.
Experiments: Decay Factor *d* of EPS(CW Dataset)

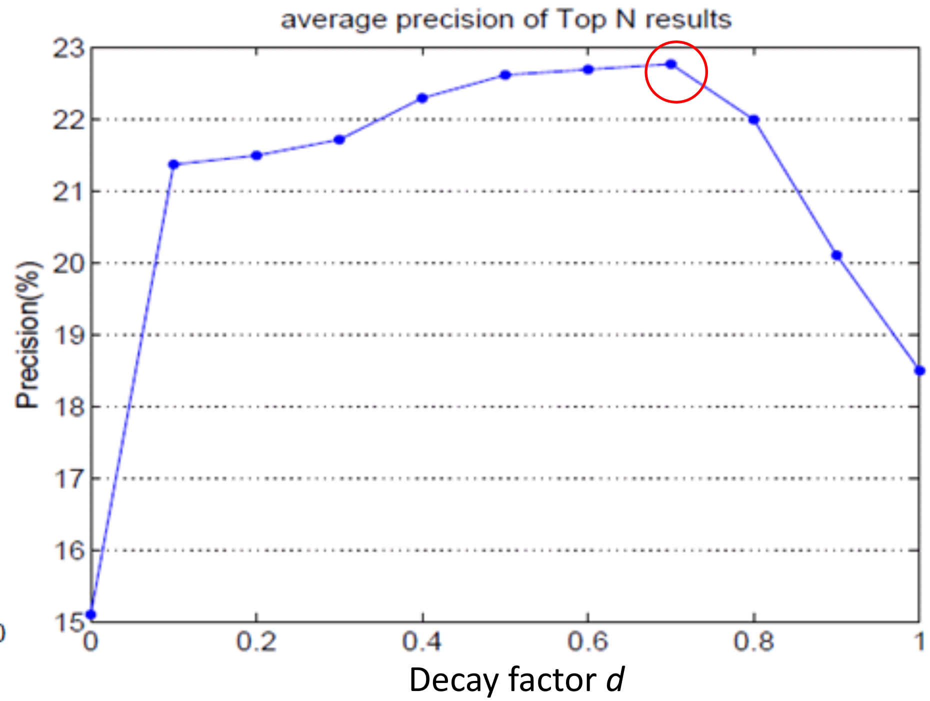




- (a) Optimal setting: *d* = 0.7
- (b) *d* = 1 corresponds to the original PageSim → *EPS* outperforms *PS*

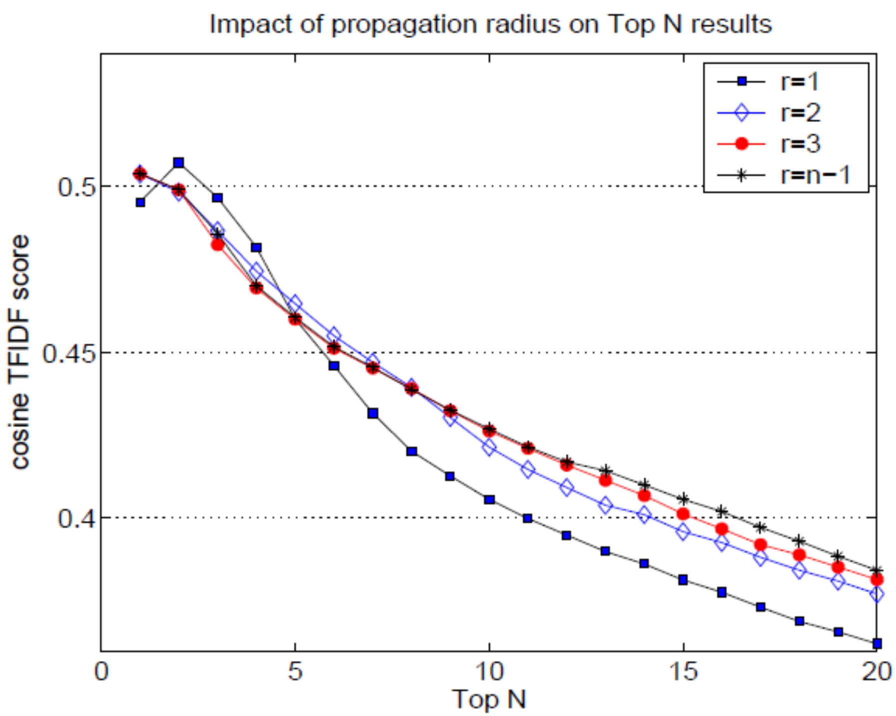
Appendix 9: Part 3. Decay Factor *d* of EPS(GS Dataset)



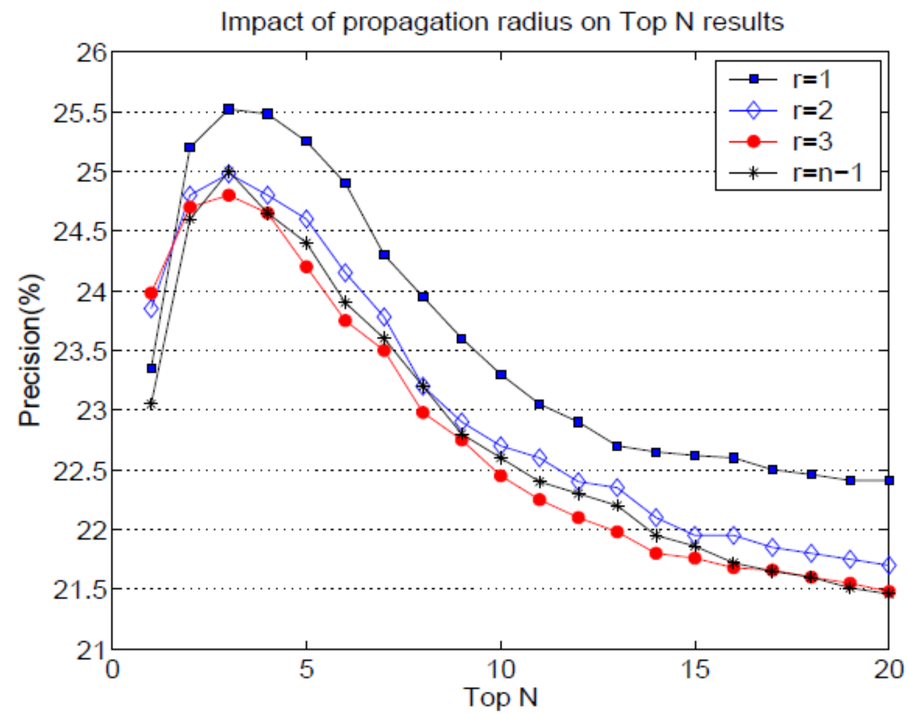


- (a) Optimal setting: *d* = 0.7
- (b) *d* = 1 corresponds to the original PageSim → *EPS* outporms *PS*

Appendix 10: Part 3. Propagation Radius *r* of EPS



CW dataset



GS dataset

- Optimal setting: *r* = 3 for CW and *r* = 1 for GS

Appendix 11. Part 4. Preliminary Experimental Results

- Item-Graph of the MovieLens dataset
  - Vertices correspond to the items;
  - Edges correspond to co-watches;
  - Weights of edges correspond to the times of co-watches.

Table 2: The characteristics of the Item-Graph

# of vertices	Average Neighbor	Average Weight
1682	773.67	13.43