Recommender Systems with Implicit Feedback: Challenges, Techniques, and Applications

2018-02-21

Yeon-Chang Lee

Advisor: Sang-Wook Kim

Big Data Science Lab.

Hanyang University

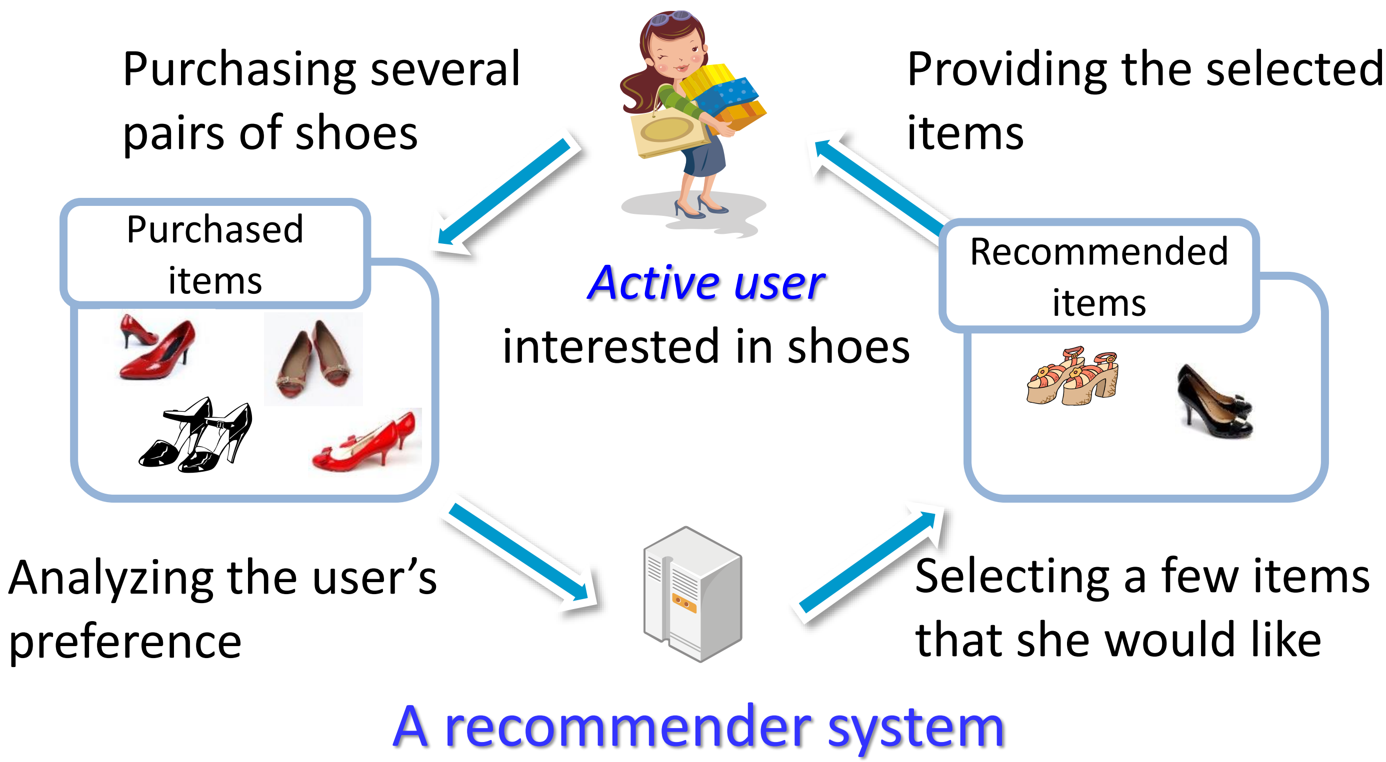
Contents

- gOCCF: Graph-Theoretic One-Class Collaborative Filtering Based on Uninteresting Items (AAAI 18)
  - Background
  - Motivation
  - Proposed Approach
  - Evaluation
  - Conclusions
- Applications of OCCF
  - Job Recommendation (ACM SAC 16, IEEE SMC 17)
  - Paper Recommendation (IEEE SMC 16)
  - TV Show Recommendation (Submitted to IJCAI 18)

gOCCF: Graph-Theoretic One-Class Collaborative Filtering Based on Uninteresting Items

Background: Recommender Systems

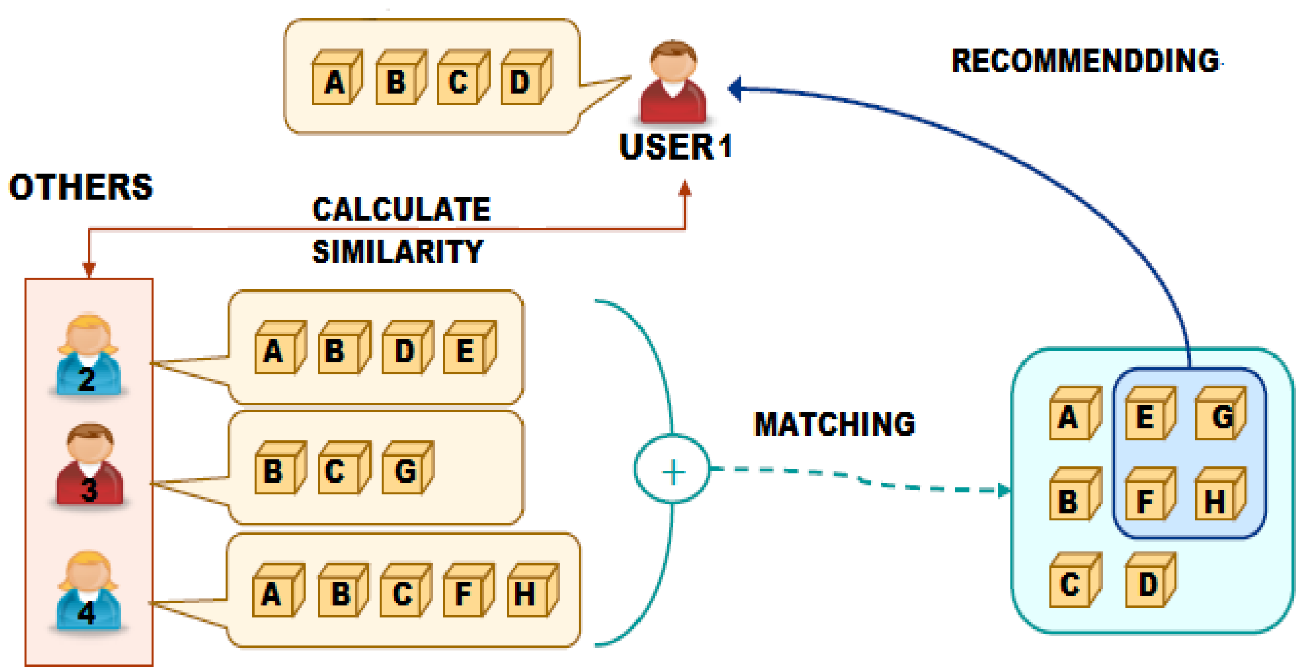
- **Provide a user with a few items that she would like**
  - Using her history of evaluating, purchasing and browsing



Overview of recommender systems

Background: Collaborative Filtering

- One of popular recommendation techniques that uses the similarity between users' past feedback
  - Explicit feedback (*e.g.*, 1-5 stars rating)
  - *Implicit feedback* (*e.g.*, purchase logs, browsing history)



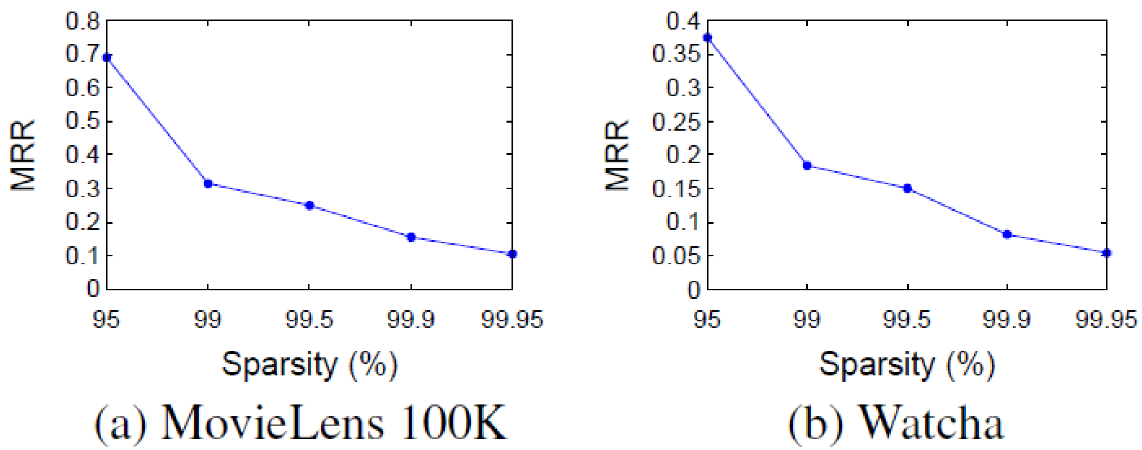
Overview of collaborative filtering

Motivation

- ***One-Class Collaborative Filtering (OCCF)*** **problem**
  - Problem to *handle the implicit feedback*
    - *One-class setting* (clicked or unknown) versus multi-class setting (1-5 stars rating)
  - Challenges
    - *Less information to capture a user's taste* than multi-class setting
    - *More sparse* datasets than in multi-class setting

Motivation

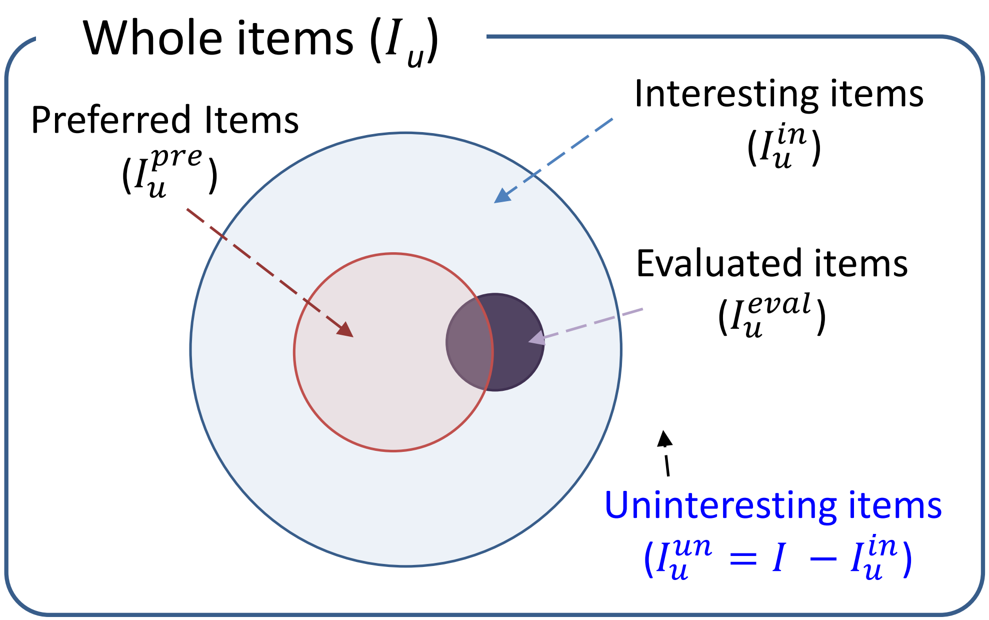
- **Existing OCCF approaches**
  - Common concept
    - View all unrated items as negative preferences
    - *e.g.*, WRMF, BPRMF
  - Challenge
    - *Less effective in dealing with sparse dataset* with many unrated items



Accuracy of an OCCF method per sparsity

Motivation

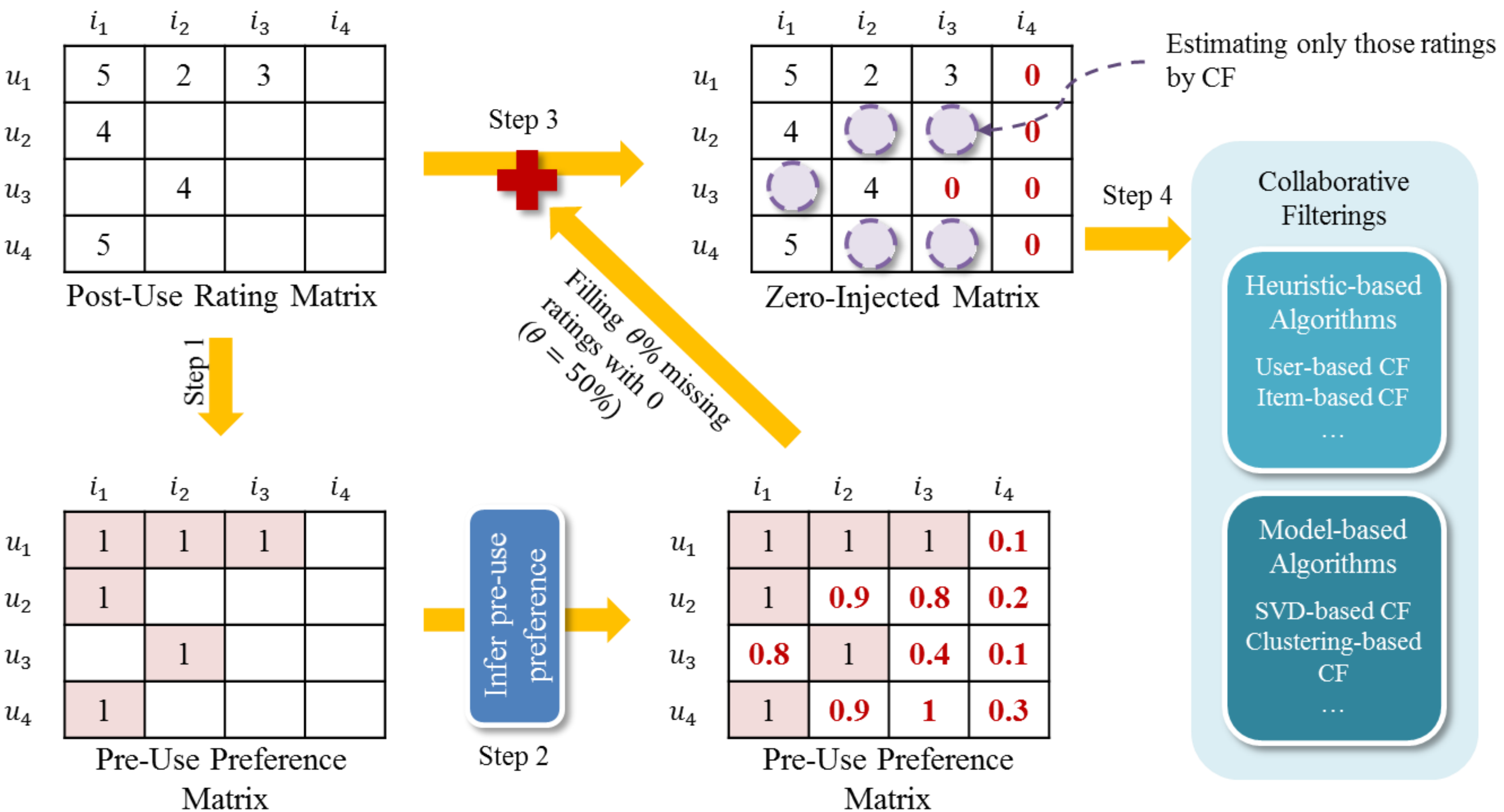
- **Zero-injection [*Hwang et al., ICDE 16*]**
  - Address the sparsity problem successfully in *multi-class setting*
  - Define a novel notion of *'uninteresting items' (U-items, in short) of a user*
    - Items on which the user has "*negative*" preferences



Venn diagram for preferences of items

Motivation

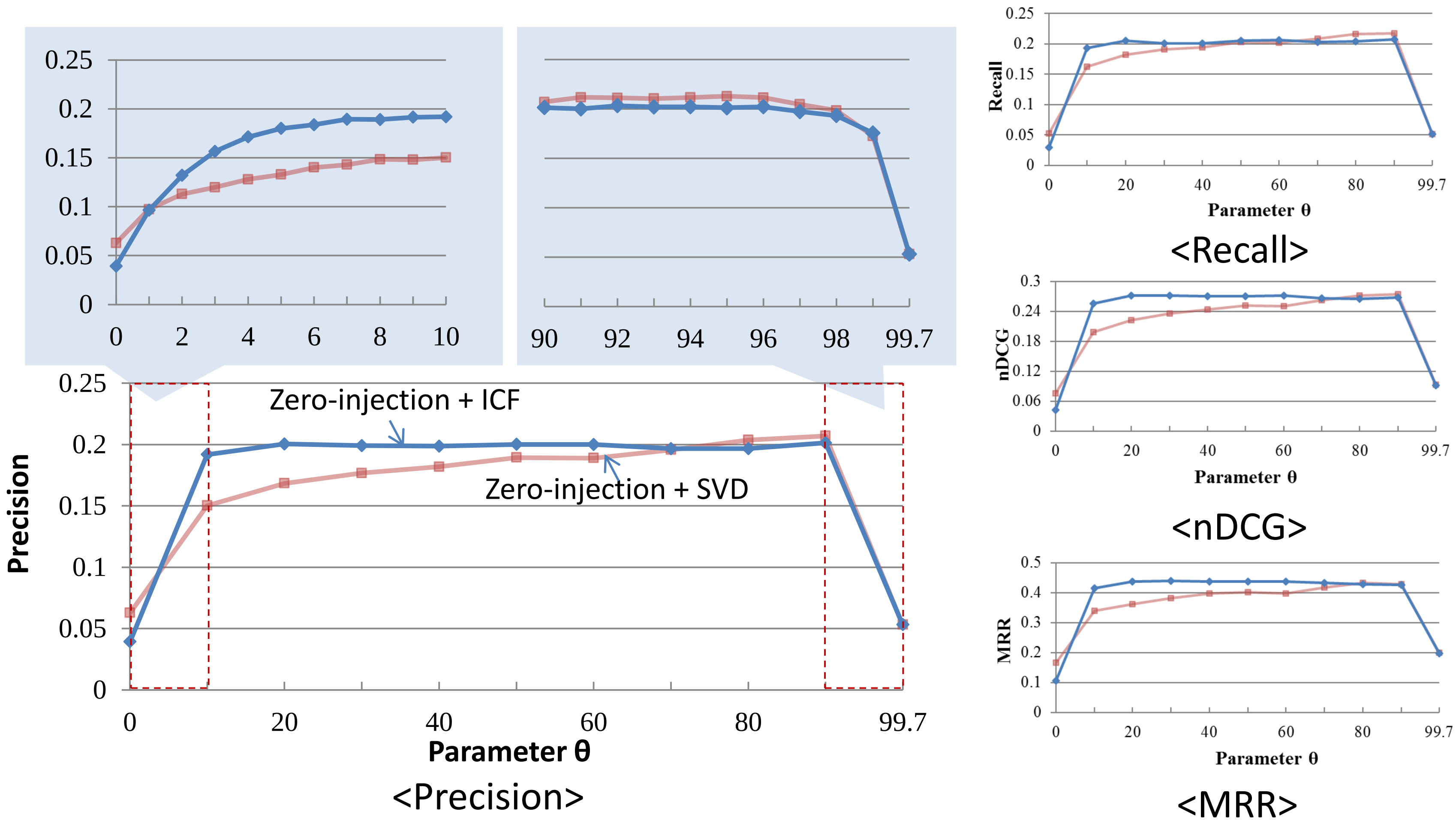
□ Zero-injection [*Hwang et al., ICDE 16*]



Overview of zero-injection

Motivation

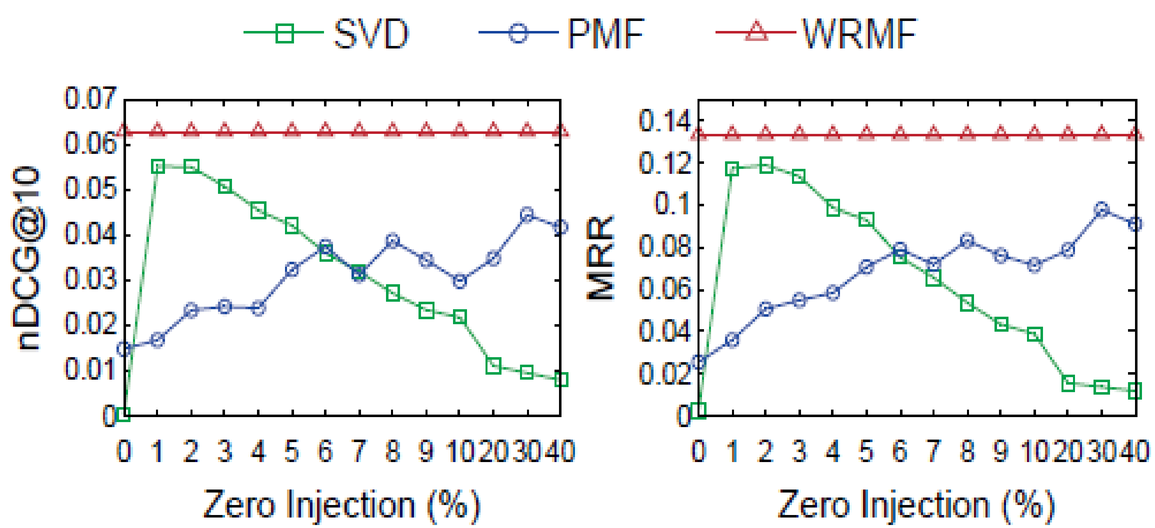
☐ Zero-injection [*Hwang et al., ICDE 16*]



Accuracy of zero-injection

Motivation

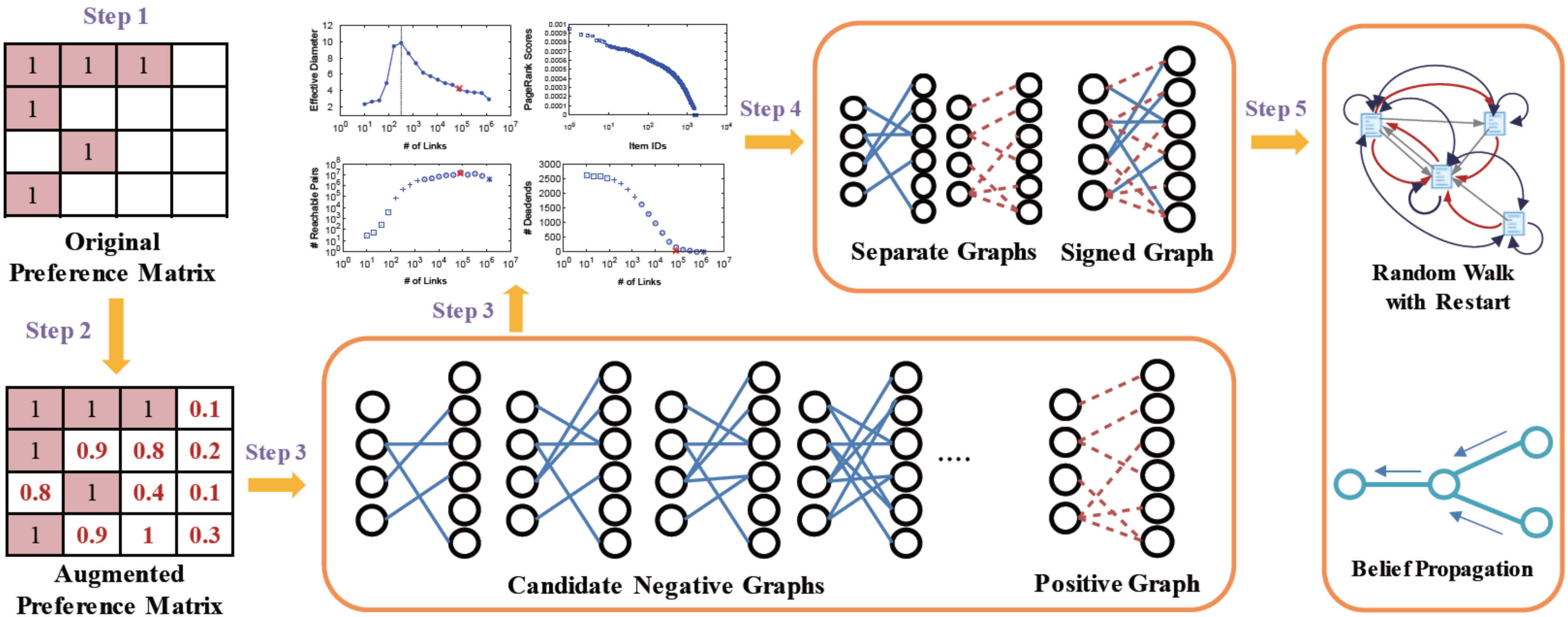
- **Zero-injection [*Hwang et al., ICDE 16*]**
  - A naïve application of zero-injection in *one-class setting*
    - *Lower accuracy* than OCCF approaches
    - *Sensitivity* to the number of U-items



Accuracy in one-class setting: SVD and PMF with varying degree of zero injection and WRMF without zero-injection

Overview of Our Approach: gOCCF

☐ Graph-Theoretic One-Class Collaborative Filtering



Overview of gOCCF

- (S1) Inferring the degree of interestingness
- (S2) Determining the number of U-items without θ
- (S3) Graph Modeling and Recommendation

(S1) How to infer the degree of interestingness

- **Employ a popular OCCF method (WRMF) [*Pan et al., ICDM 08*]**
- **Steps**
  - Treat all unrated items as negative preferences (*i.e.*, value of 0)
  - Assign different weights to quantify the relative contribution
  - Predict the value of by matrix factorization

1	1	1	
1			
		1	

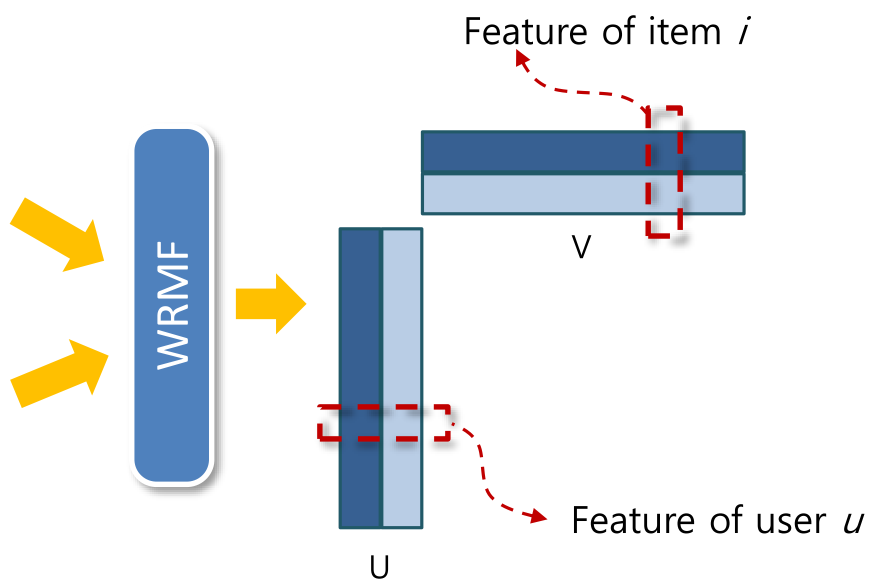
Unary rating matrix

1	1	1	0
1	0	0	0
0	0	1	0

Binary rating matrix

1	1	1	0.6
1	0.2	0.2	0.2
0.2	0.2	1	0.2

Weight matrix



(S1) How to infer the degree of interestingness

Equations

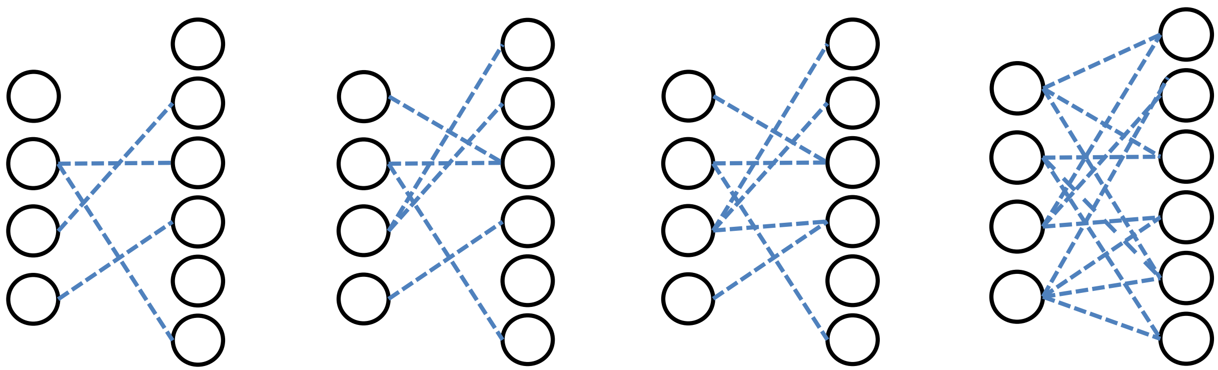
- Objective function
  - $\mathcal{L}(X,Y) = \sum_u \left[\sum_i w_{ui} \left\{\left(p_{ui} - X_u Y_i^T\right)^2 + \lambda\left(\left\|X_{u(\cdot)}\right\|_F^2 + \left\|Y_{i(\cdot)}\right\|_F^2\right)\right\}\right]$
- Updates elements in the matrices X and Y
  - $X_{u(\cdot)} = p_{u(\cdot)}\widetilde{w}_{u(\cdot)}\mathrm{Y}\left\{Y^T\widetilde{w}_{u(\cdot)}Y + \lambda\left(\sum_i w_{ui}\right)L\right\}^{-1}$
    - $\widetilde{w}_{u(\cdot)}$ is a diagonal matrix with elements of $w_{u(\cdot)}$ on the diagonal
    - Matrix L is an identity matrix
  - $Y_{i(\cdot)} = \mathrm{p}_{(\cdot)i}^T\widetilde{w}_{(\cdot)i}\mathrm{X}\left\{X^T\widetilde{w}_{(\cdot)i}X + \lambda\left(\sum_u w_{ui}\right)L\right\}^{-1}$
- Final value
  - $\hat{P} \approx \mathrm{P} = \mathrm{X}Y^T$

(S2) How to determine a right number of U-items

- **Brute-force way**
  - Construct negative graphs G^- by gradually increasing # of U-items
  - Measure the accuracy of each negative graph G^-
  - Choose the best G^- that provides the highest accuracy
- **Challenge:** ***the search space for # of U-items***
  - Large # of U-items
    - There are 1.5M unrated items among which U-items can be chosen in MovieLens
  - Prohibitively expensive to find optimal # of U-items
    - Perform a recommendation algorithm on every G^- for a given dataset
    - *Dataset and algorithm dependent*

(S2) How to determine a right number of U-items

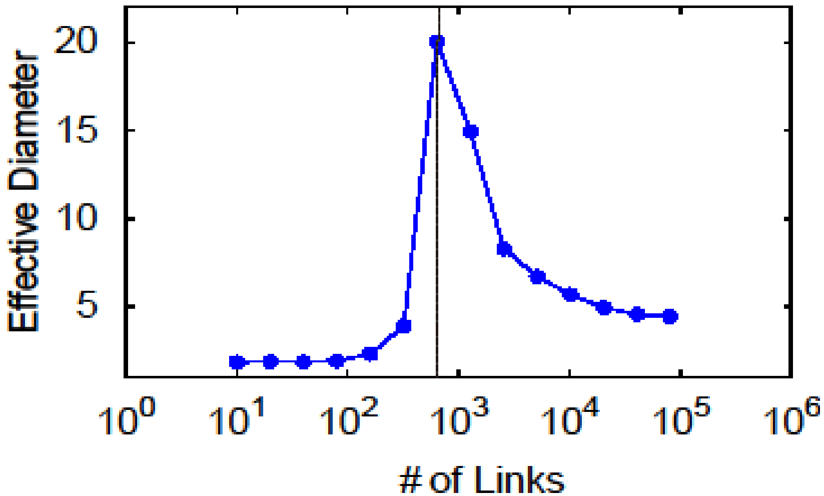
- **Consider k unrated user-item pairs with the lowest degree of interestingness as *negative links***
  - Starting from 10, k doubles until negative links reach 90% of unrated pairs
- **Model them as *a single bipartite (negative) graph* G^-**
- **Examine each G^- by *analyzing graph properties***
  - Topological properties
  - Information propagation



Candidate negative graphs

Topological property

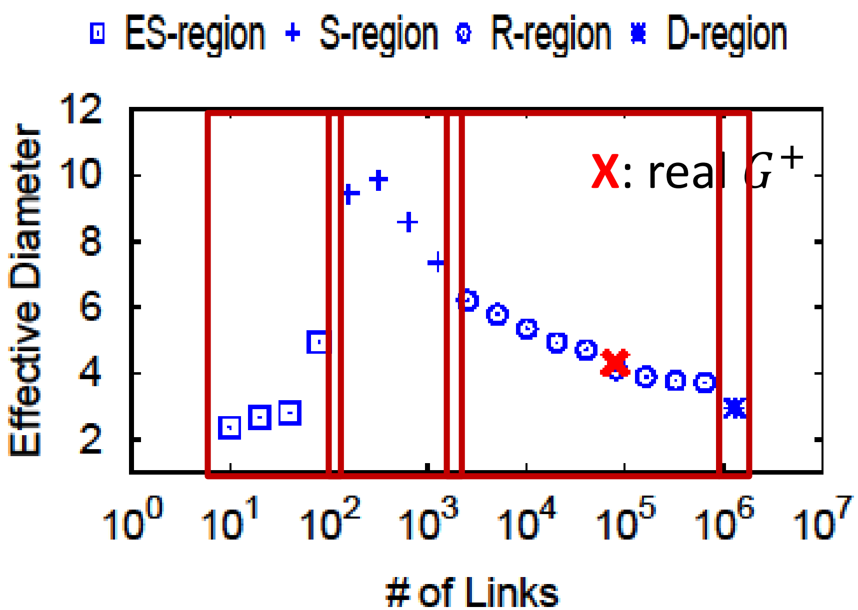
- **Graph shattering theory [*Appel et al., SDM 09*]**
  - Introduces a “*shattering point*”
    - Connectivity of a graph becomes seriously collapsed
    - As links are continuously removed in a random way
  - *ShatterPlot*: visualizing the process of generating the shattering point



ShatterPlot for positive graph G^+ of MovieLens 100K

Topological property

- **Examine the change in topological properties of G^-**
  - ShatterPlot for G^- *has the shattering point*
  - G^- has the most similar topological property to G^+ when *it's number of links equal to that of G^+*

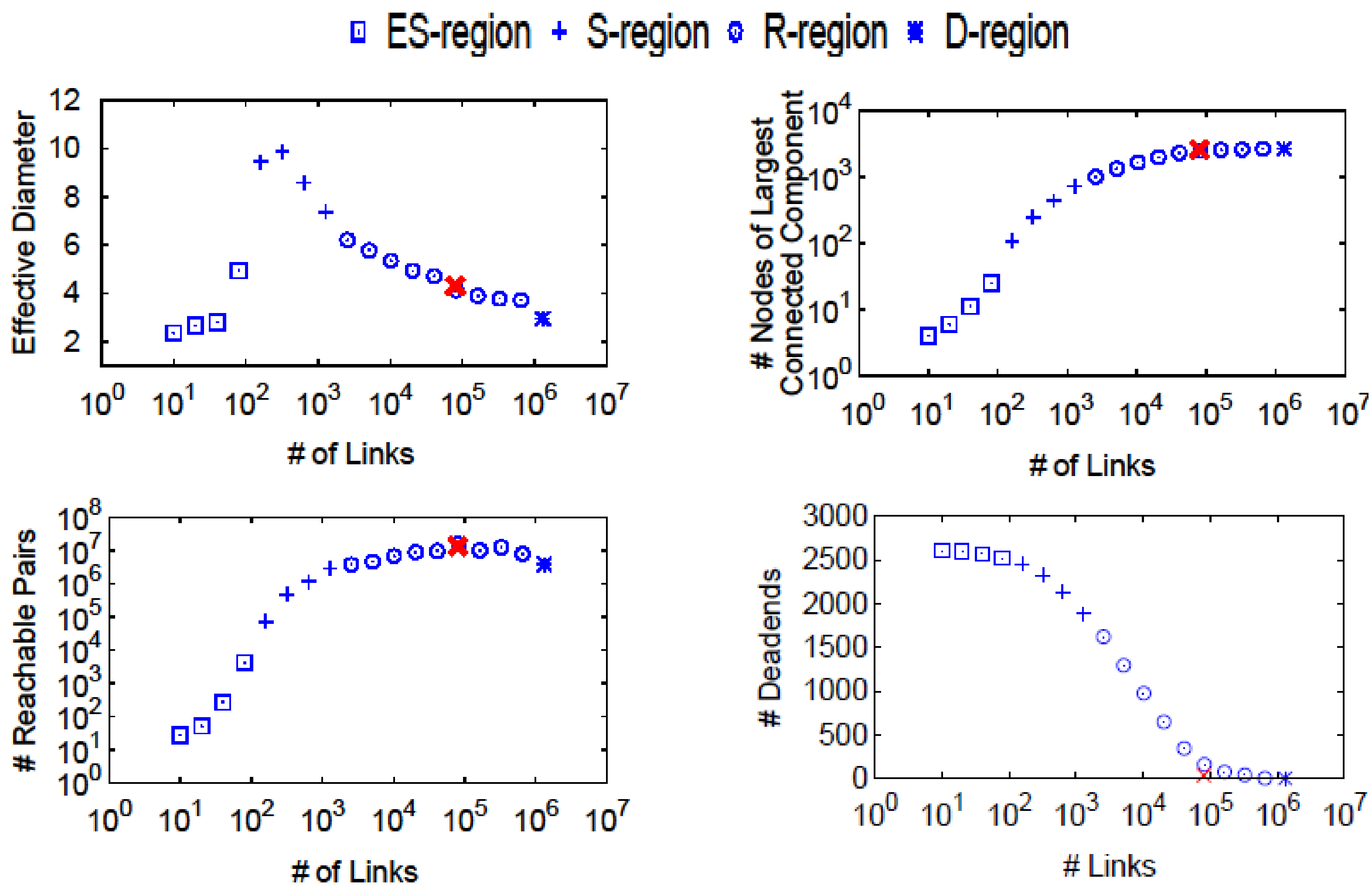


- ES-region: extremely sparse region
- S-region: shattered region
- R-region: real graph region
- D-region: dense region

ShatterPlot for negative graph G^- of MovieLens 100K

Topological property

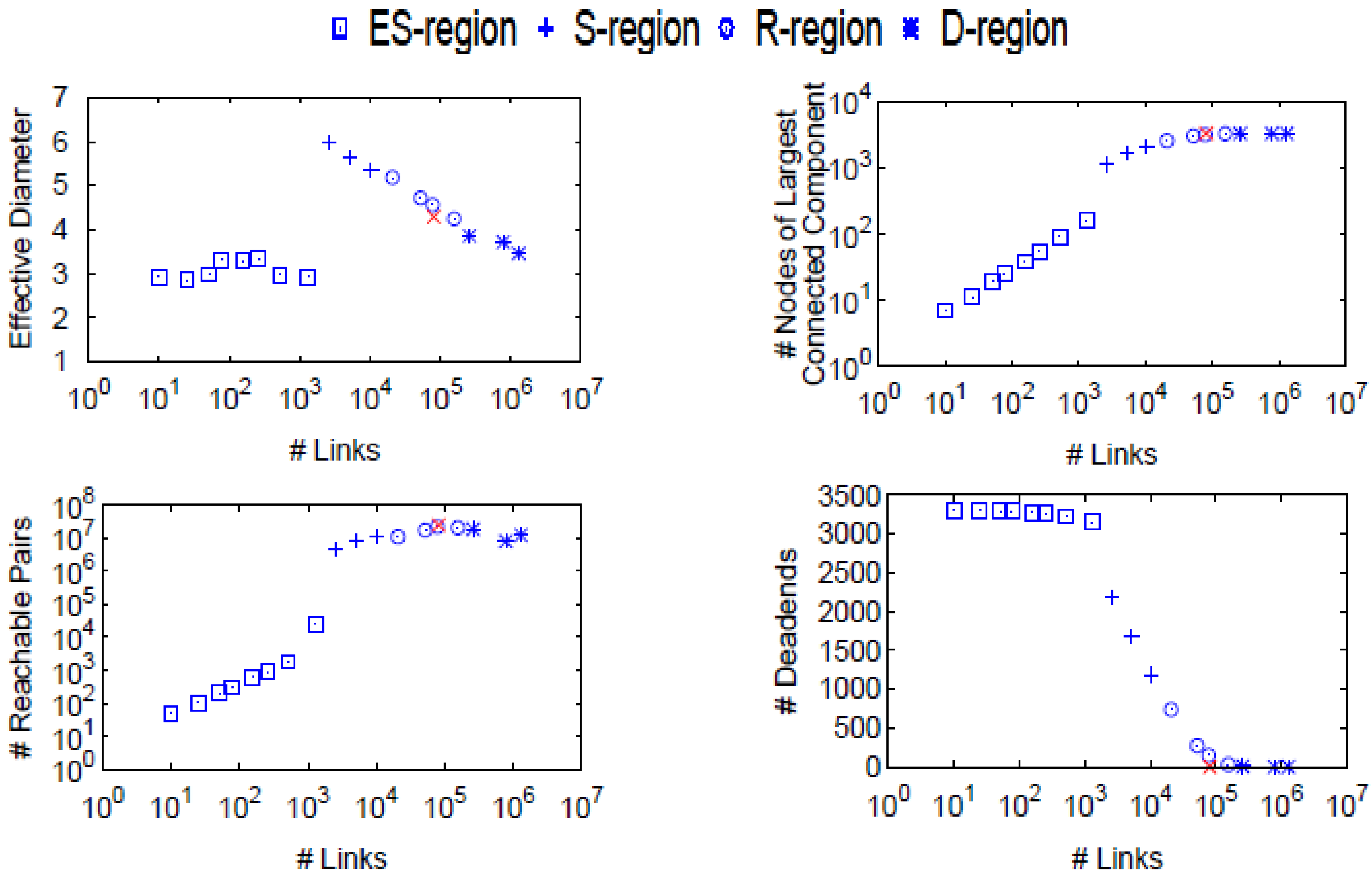
☐ The results with different metrics



ShatterPlot for negative graph G^- of MovieLens 100K

Topological property

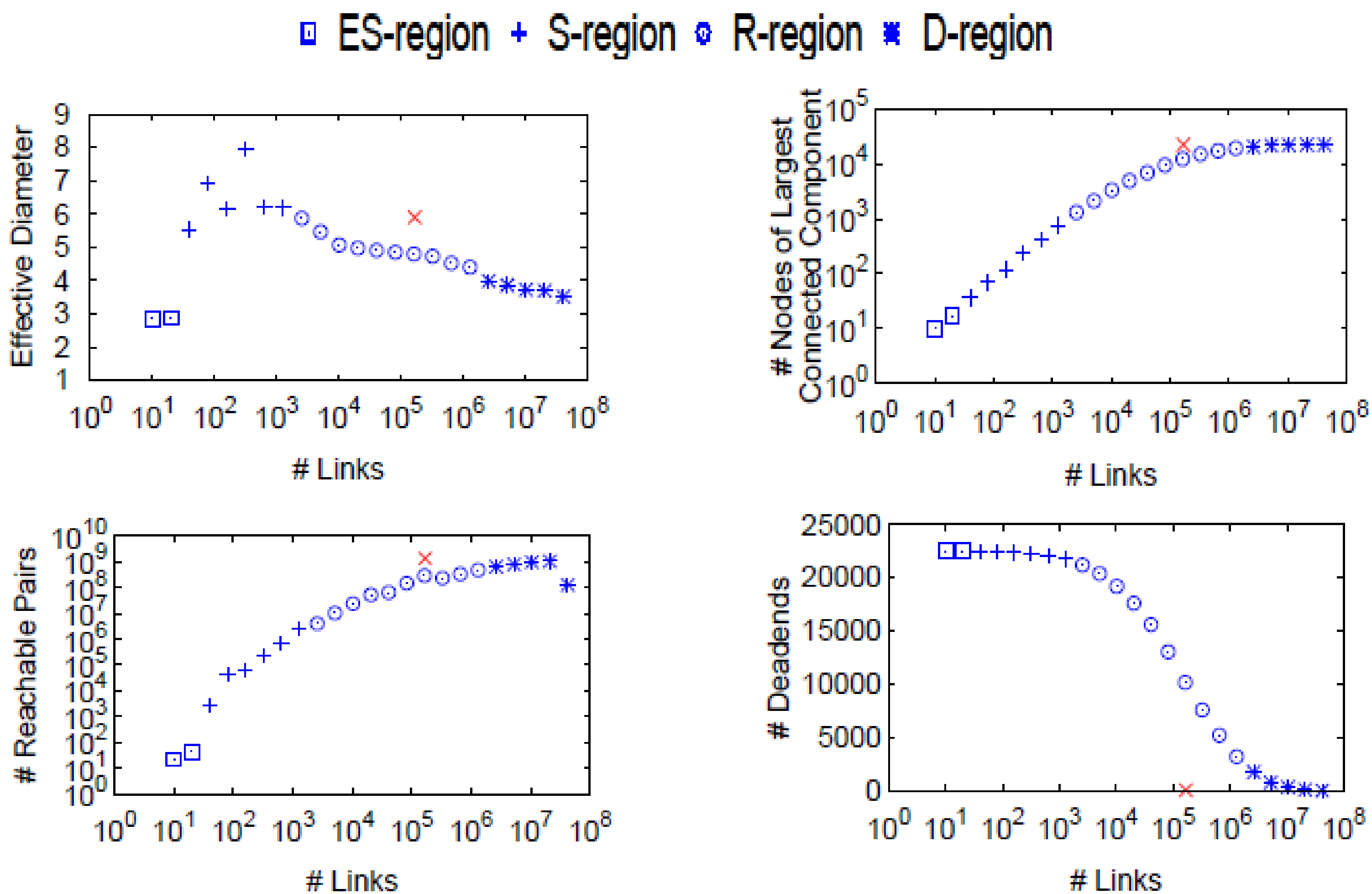
☐ The results with different datasets



ShatterPlot for negative graph G^- of Watcha

Topological property

The results with different datasets



ShatterPlot for negative graph G^- of CiteULike

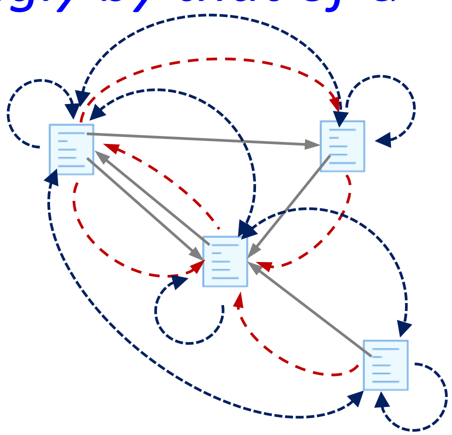
Information propagation

☐ gOCCF is based on graph analysis techniques

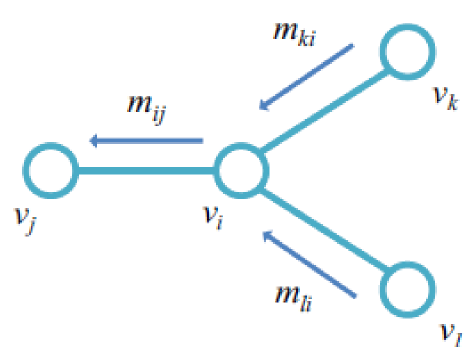
- To analyze the *information propagation in a graph*
- To make a recommendation list based on the analysis result

☐ Effects of employing G^-

- Appear when *information propagation of G^+ can be changed accordingly by that of G^-*



Random Walk with Restart

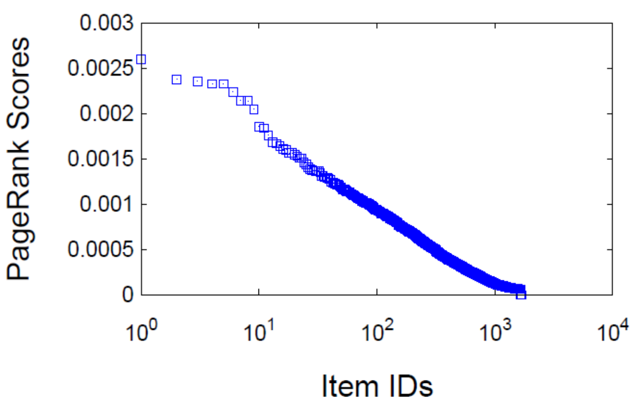


Belief Propagation

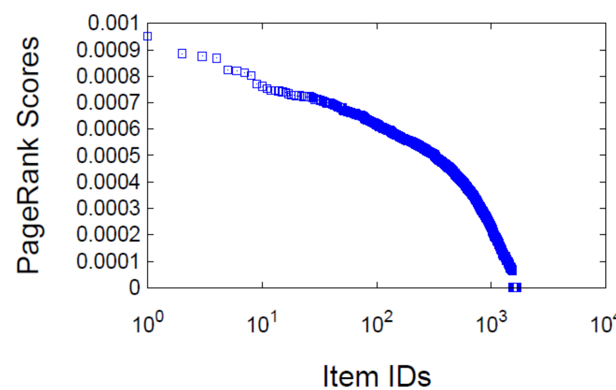
Examples of graph analysis techniques

Information propagation

- **Examine the PageRank scores of G^-**
  - G^- shows a power law-like distribution, similar to real G^+ when *it's number of links equal to that of G^+*
  - When we employ G^-, the existing propagation of G^+ seems to *have significant changes*
    - *Negative scores with various values* influence positive scores of G^+



Real positive graph G^+



R-region

PageRank scores for G^+ and the representative graph in R-region of G^-

Information propagation

☐ The results with different G^-

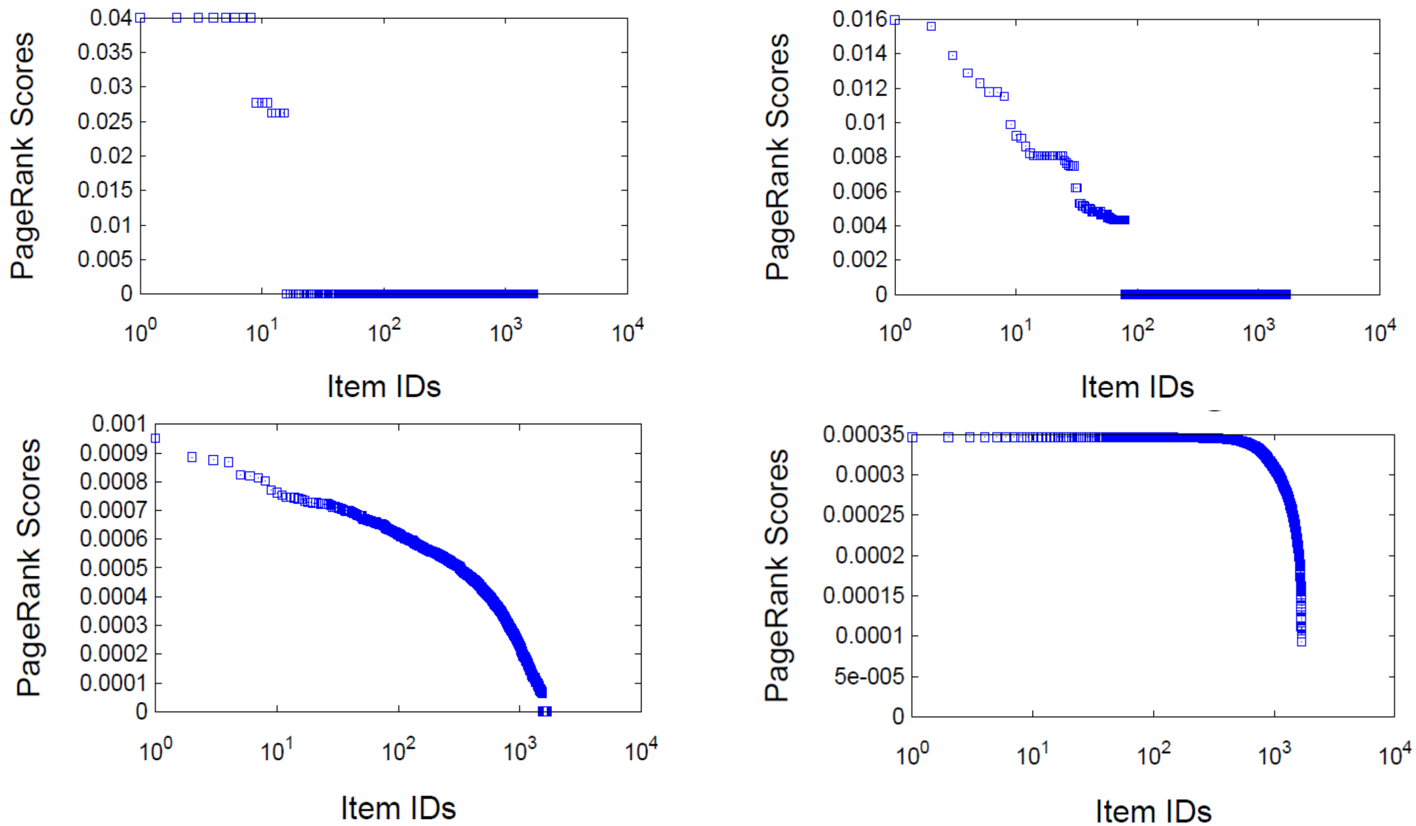
- It is difficult to change the existing propagation of G^+ using other G^-

ES-region

S-region

R-region

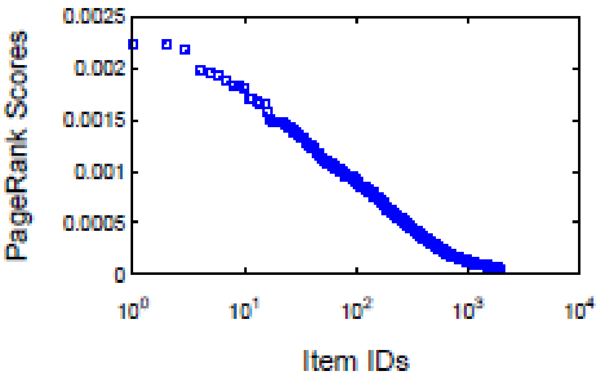
D-region



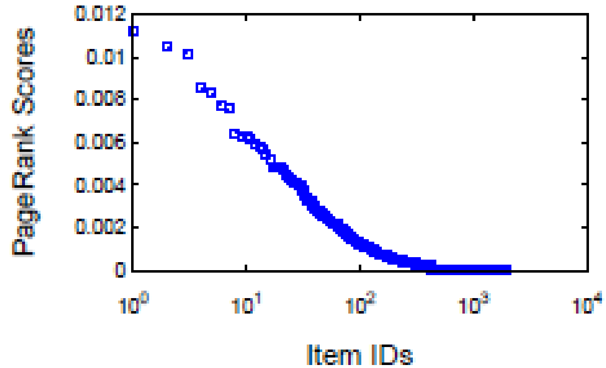
PageRank scores for the representative graphs in other regions of G^-

The results with different datasets

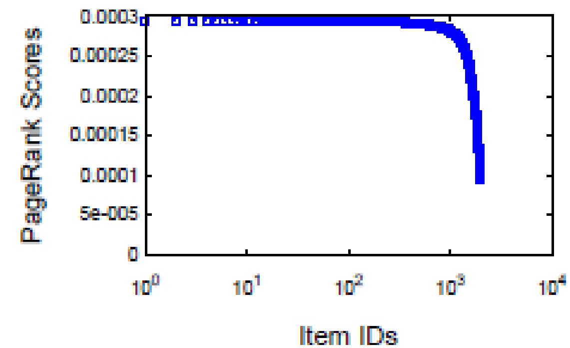
- Watcha



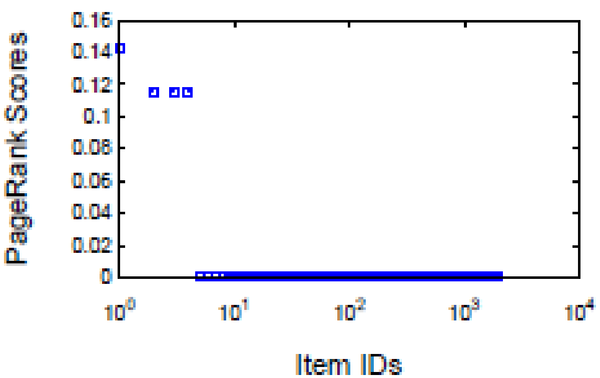
(a) Real Positive Graph



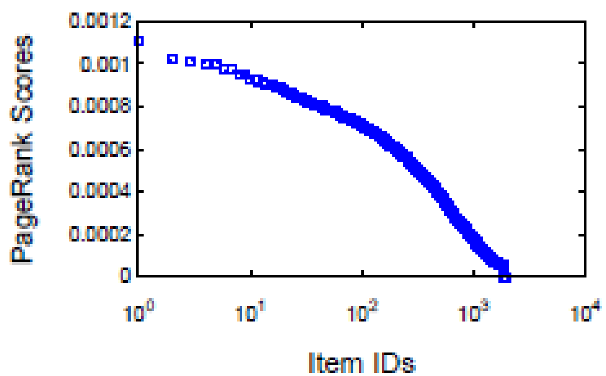
(c) S-region



(e) D-region



(b) ES-region



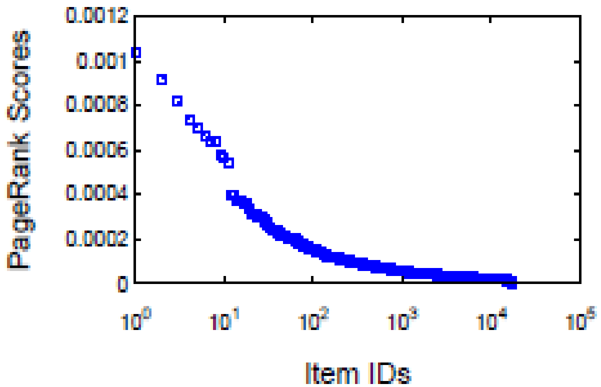
(d) R-region

PageRank scores for G^+ and the representative graphs of G^-

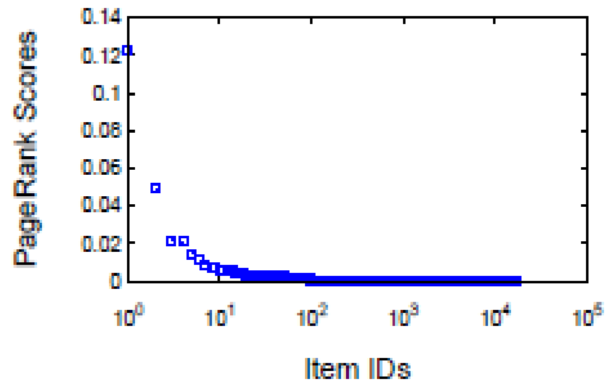
Information propagation

☐ The results with different datasets

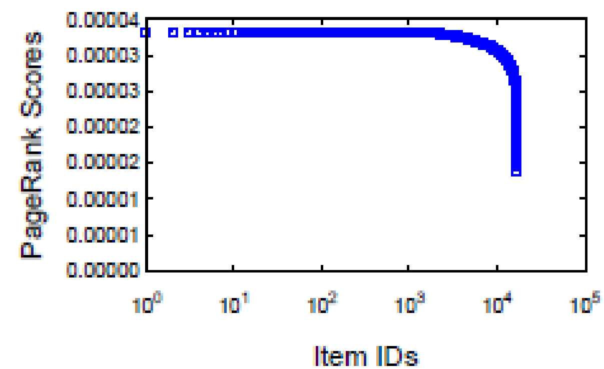
- CiteULike



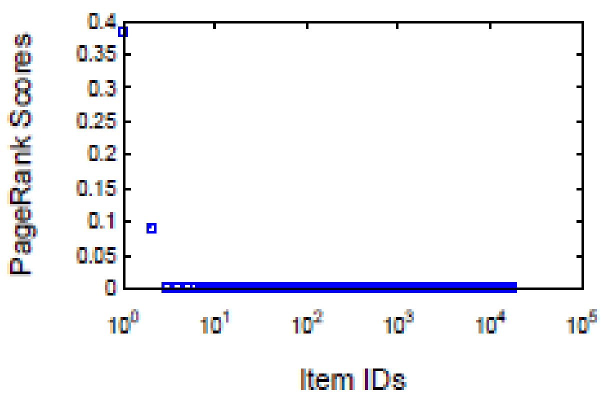
Real positive graph G^+



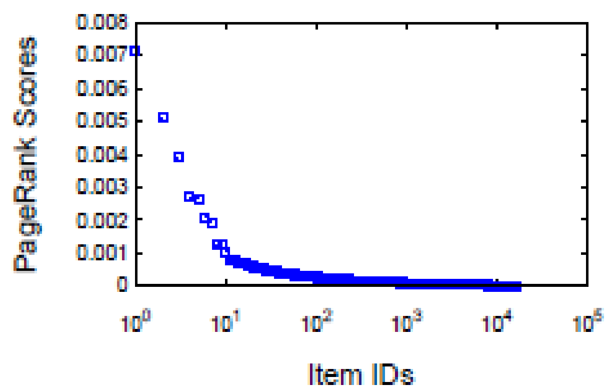
S-region



D-region



ES-region

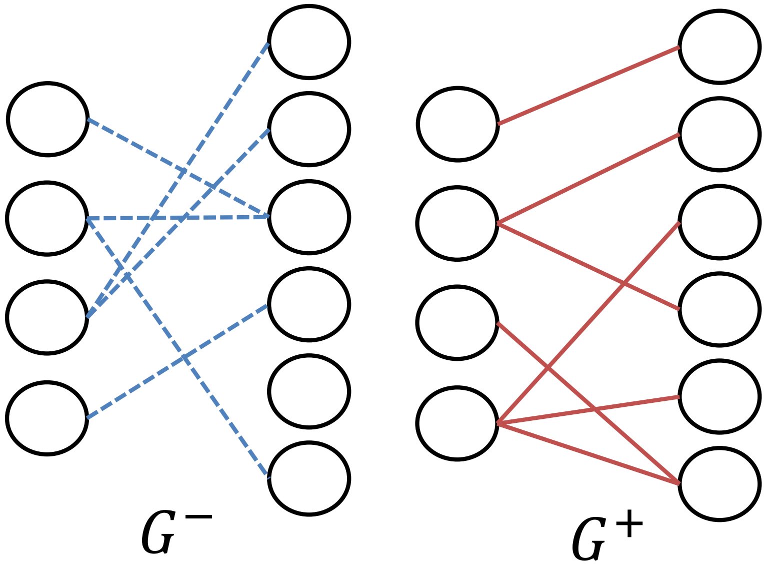


R-region

PageRank scores for G^+ and the representative graphs of G^-

(S3) How to recommend top-*N* items

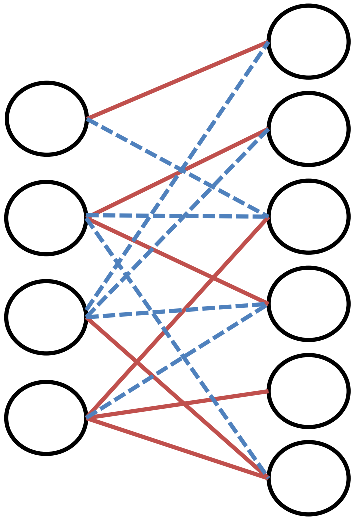
- **Transform a dataset in one-class setting to one in *binary-class setting with both U-items and I-items***
- **Model binary-class information as undirected graphs**
  - *Two separate graphs*
    - Employ two variants of random walk with restart and belief propagation
      - SeparateRWR
      - SeparateBP



Separate graphs

(S3) How to recommend top-*N* items

- **Transform a dataset in one-class setting to one in *binary-class setting with both U-items and I-items***
- **Model binary-class information as undirected graphs**
  - *A single signed graph*
    - Employ a variant of belief propagation
      - SignedBP



Signed graph $G^{\pm}$

SeparateRWR

- Perform RWR separately on each graph consisting of only one type of links
- Equations
  - $\vec{r} = \vec{r}^{(+)} - \vec{r}^{(-)}$
    - $\vec{r}^{(+)} = \alpha \overline{w}^{(+)} \vec{r}^{(+)} + (1-\alpha)\vec{t}$
      - $\vec{r}^{(+)}$: positive ranking vectors of all nodes
      - $\overline{w}^{(+)}$: normalized weight matrix built based on positive links
    - $\vec{r}^{(-)} = \alpha \overline{w}^{(-)} \vec{r}^{(-)} + (1-\alpha)\vec{t}$
      - $\vec{r}^{(-)}$: negative ranking vectors of all nodes
      - $\overline{w}^{(-)}$: normalized weight matrix built based on negative links
      - α: a damping factor
      - $\vec{t}$: a personalization vector representing the target nodes to be restarted

SeparateBP

- Perform BP separately on each graph consisting of only one type of links
- Equations
  - Message passing
    - $m_{ij}^{(+)}(\sigma) \leftarrow \sum_{\sigma} \phi_i^{(+)}(\sigma')\varphi_{ij}^{(+)}(\sigma',\sigma) \prod_{k \in N(i)^{(+)} \setminus \mathrm{j}} m_{ki}^{(+)}(\sigma)$
    - $m_{ij}^{(-)}(\sigma) \leftarrow \sum_{\sigma} \phi_i^{(-)}(\sigma')\varphi_{ij}^{(-)}(\sigma',\sigma) \prod_{k \in N(i)^{(-)} \setminus \mathrm{j}} m_{ki}^{(-)}(\sigma)$
  - Belief score
    - $b_i^{(+)}(\sigma) = k \prod_{j \in N(i)^{(+)}} m_{ji}^{(+)}(\sigma)$
    - $b_i^{(-)}(\sigma') = k \prod_{j \in N(i)^{(-)}} m_{ji}^{(-)}(\sigma')$
  - Final score
    - $b_i(\sigma) = b_i^{(+)}(\sigma) - b_i^{(-)}(\sigma')$

SignedBP

- **Perform BP on a signed graph that contains both positive and negative links together**
- **Equations**
  - Message passing
    - $m_{ij}(\sigma) \leftarrow \begin{cases} \sum_{\sigma} \phi_i(\sigma')\varphi_{ij}^{(+)}(\sigma',\sigma) \prod_{k\in N(i)\backslash \mathrm{j}} m_{ki}(\sigma') \\ \sum_{\sigma} \phi_i(\sigma')\varphi_{ij}^{(-)}(\sigma',\sigma) \prod_{k\in N(i)\backslash \mathrm{j}} m_{ki}(\sigma') \end{cases}$
  - Belief score
    - $b_i(\sigma) = k \prod_{j\in N(i)} m_{ji}(\sigma)$

Empirical validation

Datasets: MovieLens 100K, Watcha*, CiteULike

Dataset statistics

Datasets	# users	# items	# user-item pair	Sparsity
Movielens 100K	943	1,682	100,000	93.69%
Watcha	1,391	1,927	100,000	96.98%
CiteULike	5,551	16,980	210,504	99.82%

* Watcha is the largest Korean movie rating system

Metrics: Precision, Recall, nDCG, MRR, HLU

Competing Methods

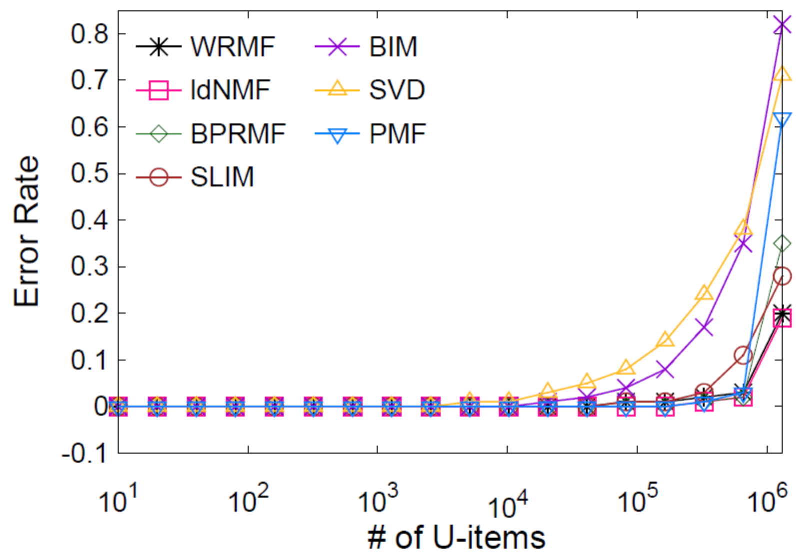
- Baseline: MostPopular
- Graph-based methods: RWR, BP
- Zero-injection methods: SVD_ZI, PMF_ZI
- OCCF methods: WRMF, ldNMF, SLIM, BPRMF, GBPRMF

Questions to Be Answered

- **Q1:** Is the degree of interestingness of the items inferred by the WRMF useful for recommendation?
- **Q2:** Are the U-items selected by our U-items decision method useful for recommendation?
- **Q3:** Is it useful to recommend additional U-items to existing graph-based recommendations?
- **Q4:** Does our proposed OCCF methods provide more accurate recommendations than existing OCCF methods with a very sparse dataset?

Question 1

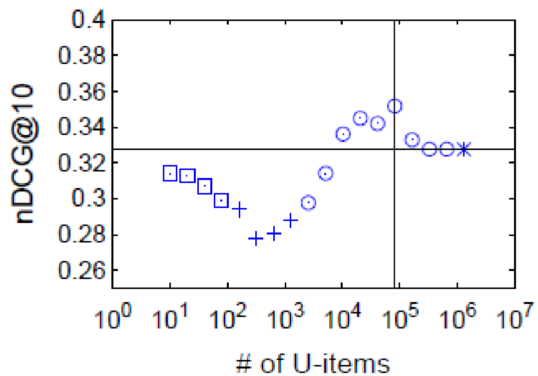
- **Accuracy metric: user *u*'s error rate**
  - $err_u^{\theta} = \frac{|I_u^{un}(\theta) \cap I_u^{test}|}{|I_u^{test}|}$
  - How many *rated items* (in the test set) are selected as uninteresting items (*i.e.*, mis-classified) for *u*
- **WRMF and IdNMF is the most effective**



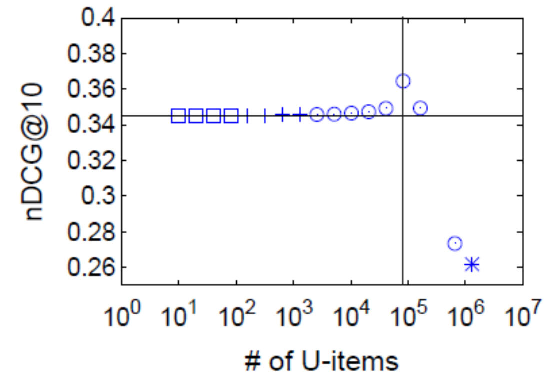
Error rates of methods for finding U-items (MovieLens)

Question 2

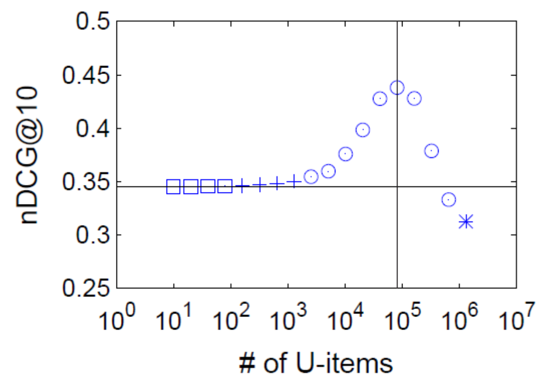
- **gOCCF provides the best accuracy** ***when having the same number of negative links as that of positive links***



SeparateRWR



SeparateBP



SignedBP

Accuracy according to # of U-items (MovieLens)

Question 3

- ☐ ***Utilizing U-items in gOCCF*** **is effective in improving the accuracy significantly**

Accuracy before and after exploiting U-items (MovieLens)

Metrics		BP					RWR		
		Origin	Separate BP	Gain	SignedBP	Gain	Origin	Separate RWR	Gain
Precision	@5	0.353	0.369	4.8%	0.438	24.1%	0.318	0.340	7.0%
	@10	0.289	0.309	7.2%	0.370	28.2%	0.277	0.302	9.2%
	@20	0.233	0.256	9.9%	0.301	29.1%	0.232	0.257	11.1%
	@50	0.168	0.191	13.6%	0.239	42.4%	0.169	0.194	14.8%
Recall	@5	0.098	0.105	7.2%	0.134	36.6%	0.100	0.105	4.9%
	@10	0.148	0.164	10.9%	0.210	42.1%	0.158	0.171	8.2%
	@20	0.221	0.255	15.7%	0.315	42.8%	0.244	0.270	11.0%
	@50	0.359	0.389	8.4%	0.445	23.9%	0.391	0.414	5.9%
nDCG	@5	0.383	0.398	3.9%	0.473	23.5%	0.348	0.368	5.8%
	@10	0.345	0.365	5.7%	0.438	27.0%	0.328	0.352	7.3%
	@20	0.328	0.356	8.5%	0.426	30.1%	0.325	0.354	8.8%
	@50	0.343	0.369	7.6%	0.438	27.9%	0.351	0.378	7.7%
MRR		0.592	0.604	1.9%	0.679	14.6%	0.570	0.584	2.5%

Question 3

Accuracy before and after exploiting U-items (Watcha)

Metrics		BP					RWR		
		Origin	Separate BP	Gain	SignedBP	Gain	Origin	Separate RWR	Gain
Precision	@5	0.130	0.142	9.3%	0.180	38.7%	0.116	0.124	7.1%
	@10	0.109	0.124	13.2%	0.151	38.3%	0.102	0.113	10.8%
	@20	0.094	0.108	14.2%	0.125	32.7%	0.089	0.099	11.5%
	@50	0.084	0.098	16.0%	0.118	39.9%	0.078	0.088	13.6%
Recall	@5	0.061	0.067	9.5%	0.086	41.6%	0.057	0.060	4.9%
	@10	0.100	0.113	13.4%	0.142	41.9%	0.097	0.107	9.6%
	@20	0.168	0.194	15.3%	0.230	37.0%	0.165	0.182	10.6%
	@50	0.226	0.248	9.6%	0.282	24.8%	0.217	0.233	7.3%
nDCG	@5	0.137	0.152	11.3%	0.195	42.5%	0.122	0.131	7.7%
	@10	0.134	0.152	13.2%	0.190	41.8%	0.124	0.136	9.9%
	@20	0.153	0.174	14.0%	0.212	38.7%	0.144	0.159	10.1%
	@50	0.174	0.194	11.6%	0.232	33.5%	0.163	0.177	8.7%
MRR		0.297	0.329	10.8%	0.391	32.0%	0.275	0.295	7.1%

Question 3

Accuracy before and after exploiting U-items (CiteULike)

Metrics		BP					RWR		
		Origin	Separate BP	Gain	SignedBP	Gain	Origin	Separate RWR	Gain
Precision	@5	0.108	0.141	30.3%	0.111	2.5%	0.128	0.158	24.0%
	@10	0.088	0.112	26.9%	0.091	2.9%	0.101	0.122	21.0%
	@20	0.070	0.083	17.6%	0.072	2.4%	0.078	0.090	15.8%
	@50	0.059	0.075	26.6%	0.061	3.3%	0.065	0.079	23.0%
Recall	@5	0.082	0.109	33.7%	0.083	1.4%	0.117	0.137	16.7%
	@10	0.128	0.162	26.4%	0.131	2.4%	0.175	0.199	13.5%
	@20	0.192	0.226	17.6%	0.196	2.1%	0.251	0.275	9.5%
	@50	0.241	0.275	14.2%	0.245	1.6%	0.307	0.388	26.1%
nDCG	@5	0.127	0.168	32.1%	0.130	2.1%	0.156	0.190	21.9%
	@10	0.135	0.175	29.9%	0.138	2.3%	0.171	0.202	18.2%
	@20	0.154	0.192	24.7%	0.157	2.0%	0.196	0.225	14.5%
	@50	0.170	0.209	22.9%	0.173	1.8%	0.215	0.261	21.6%
MRR		0.243	0.295	21.0%	0.247	1.3%	0.286	0.326	14.0%

Question 4

☐ Overall accuracy

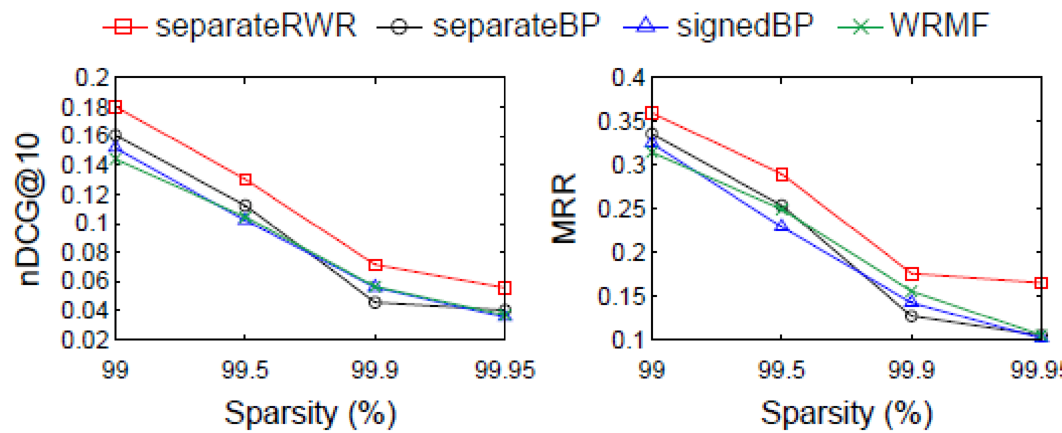
Accuracy of competing methods and gOCCF methods

Metrics	MovieLens									
	MostPopular	SVD_ZI	PMF_ZI	WRMF	BPRMF	GBPRMF	SLIM	**separateRWR**	**separateBP**	**signedBP**
P@10	0.222	0.365	0.343	**0.393**	0.368	0.385	0.337	0.302	0.309	0.370
R@10	0.113	0.203	0.198	**0.222**	0.215	0.220	0.194	0.171	0.164	0.210
nDCG@10	0.250	0.424	0.389	**0.460**	0.435	0.454	0.396	0.352	0.365	0.438
MRR	0.453	0.652	0.589	0.689	0.679	**0.692**	0.630	0.584	0.604	0.679
HLU	30.600	51.971	42.921	56.132	54.731	**56.747**	49.097	43.894	47.195	55.980
	Watcha									
	MostPopular	SVD_ZI	PMF_ZI	WRMF	BPRMF	GBPRMF	SLIM	**separateRWR**	**separateBP**	**signedBP**
P@10	0.090	0.141	0.137	0.146	0.120	0.121	0.123	0.113	0.124	**0.151**
R@10	0.085	0.134	0.136	0.140	0.110	0.115	0.117	0.107	0.113	**0.142**
nDCG@10	0.108	0.170	0.165	0.182	0.181	0.150	0.152	0.136	0.152	**0.190**
MRR	0.253	0.346	0.335	0.375	0.376	0.305	0.328	0.295	0.329	**0.391**
HLU	11.860	18.181	16.884	21.077	21.929	17.299	17.453	14.288	17.448	**23.012**
	CiteULike									
	MostPopular	SVD_ZI	PMF_ZI	WRMF	BPRMF	GBPRMF	SLIM	**separateRWR**	**separateBP**	**signedBP**
P@10	0.012	0.043	0.034	0.045	0.092	0.049	-	**0.122**	0.112	0.091
R@10	0.029	0.044	0.037	0.049	0.140	0.078	-	**0.199**	0.162	0.131
nDCG@10	0.023	0.055	0.044	0.062	0.136	0.066	-	**0.202**	0.175	0.138
MRR	0.050	0.117	0.073	0.133	0.240	0.132	-	**0.326**	0.295	0.247
HLU	1.527	5.899	4.489	7.198	12.754	4.005	-	**21.257**	19.323	14.680

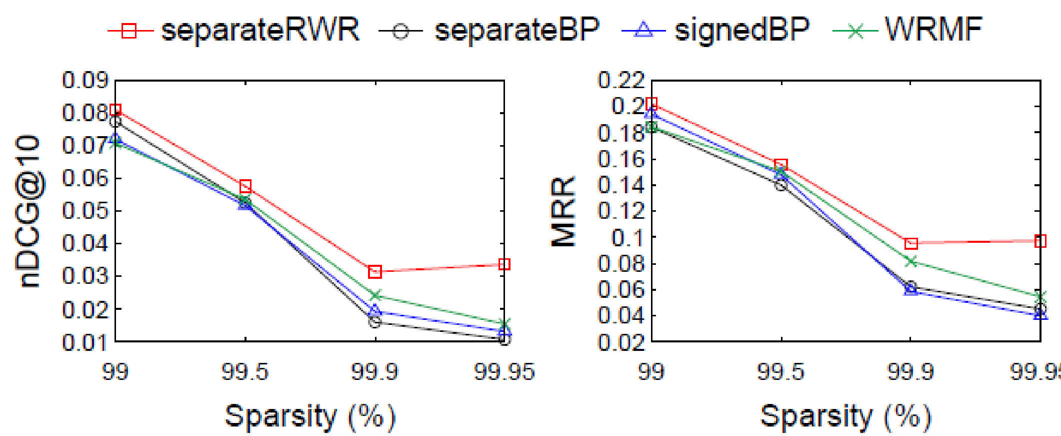
Question 4

☐ Accuracy per density

- MovieLens 100K



- Watcha



Accuracy of gOCCF methods and WRMF per sparsity

Conclusions

- **To propose *a novel graph-theoretic OCCF approach (gOCCF)* for one-class setting**

- **Strengths**
  - *Easy to implement* (almost 100 lines)
  - *Parameter free*
  - *Significantly improved accuracy*
    - Up to 48% (nDCG@10), 36% (MRR), and 67% (HLU)

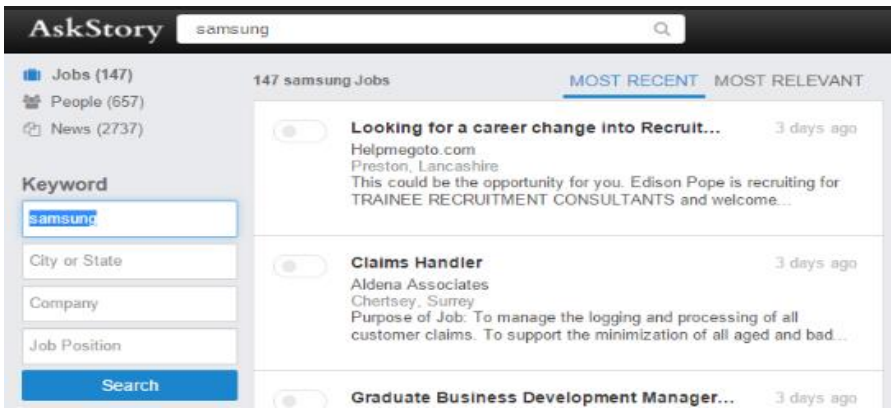
- **Datasets and implementations (coming soon)**
  - https://goo.gl/sfiawn

Applications of OCCF

JobRec (ACM SAC 16, IEEE SMC 17)

Motivation

- E-recruitment sites
  - Important places to both job seekers and recruiters
  - Examples: *Reed, Indeed, AskStory*

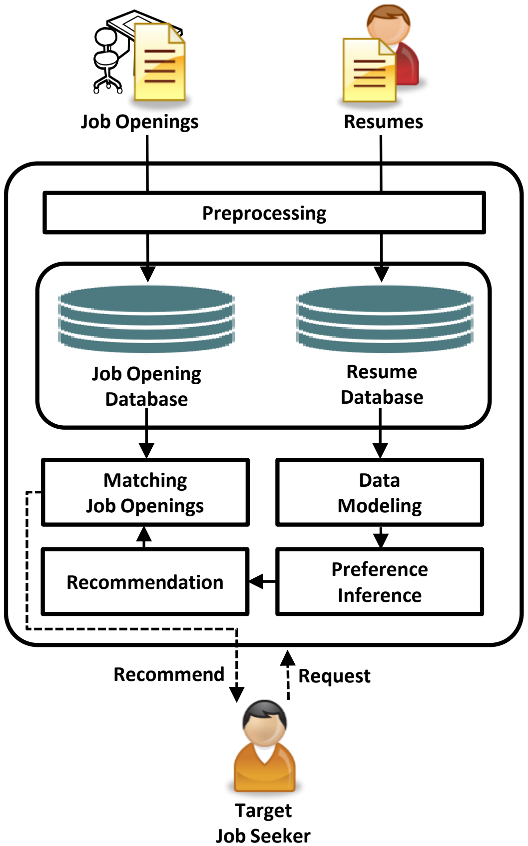


- Limitation of current *AskStory* services
  - Only provides a function of *keyword-based searches*
  - Very *time-consuming task* for a job seeker to find job openings
- Our proposal
  - *A hybrid approach* exploiting content and collaborative dataset

JobRec (ACM SAC 16, IEEE SMC 17)

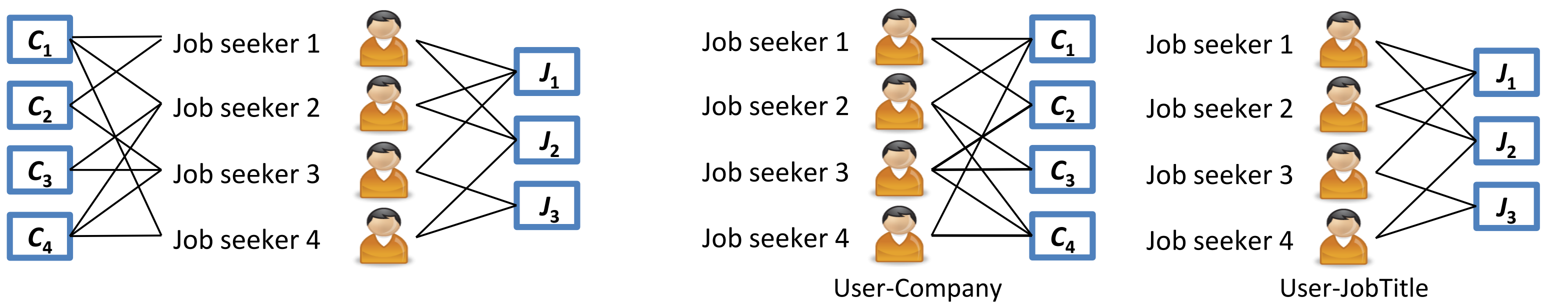
Our approach

- Overview
  - Input: a resume of a target job seeker
  - Output: a list of top-*N* job openings
- Datasets
  - *Resume* posted by job seekers
    - Consists of a series of job records representing her/his career experiences
    - A job record represents a set of a company name, a job title, a working period, and a description
  - *Job opening* posted by recruiters
    - Consists of descriptions for a company and a job title



JobRec (ACM SAC 16, IEEE SMC 17)

- Our approach
  - Overall process
    - Data preprocessing and storage
    - Extracting content and collaborative information
    - *Graph modeling* for the information extracted
      - CF graph

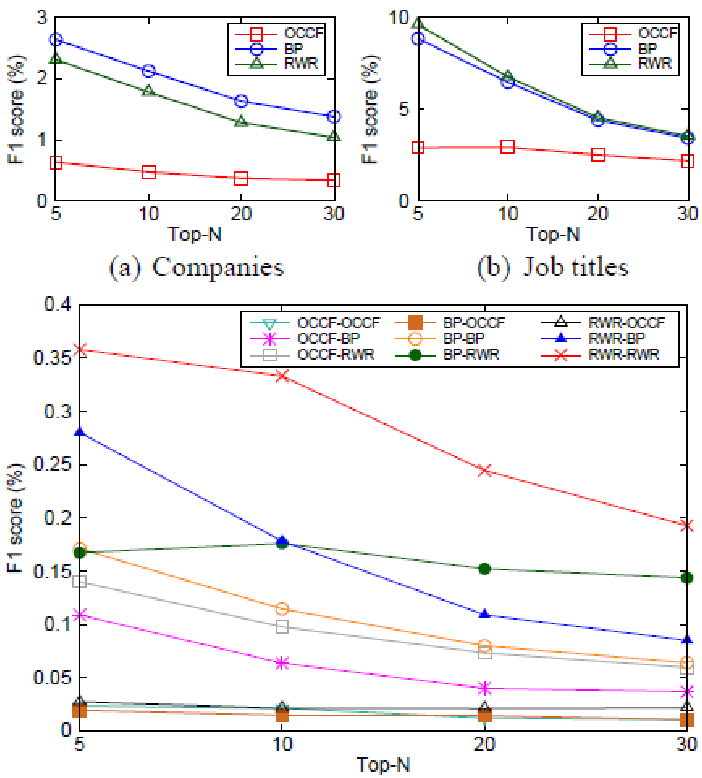


      - Hybrid graph
        - Assign *content-based weights* on the original links of the CF graph
        - Add new links based on *content-based similarities* to the CF graph
    - Analyzing the graph by using graph analysis algorithms
    - Recommending the most suitable *N* job openings to a user

JobRec (ACM SAC 16, IEEE SMC 17)

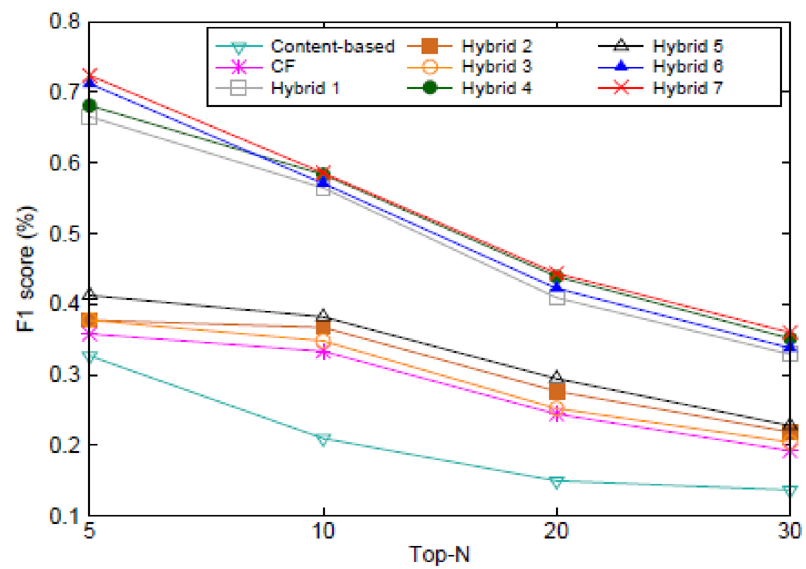
☐ Results

- *RWR algorithm* outperformed other algorithms consistently
- *Hybrid approach* outperformed other approaches consistently



(c) Job openings

Different algorithms

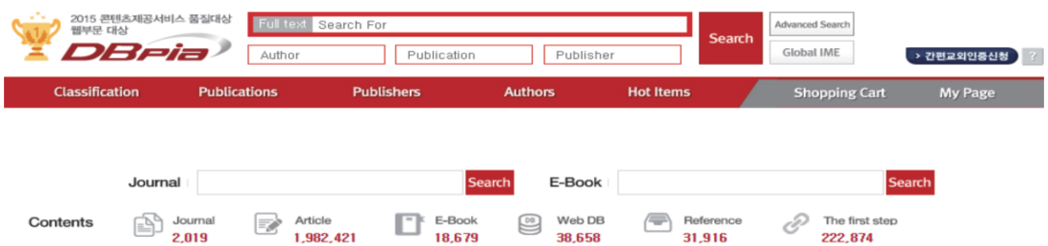


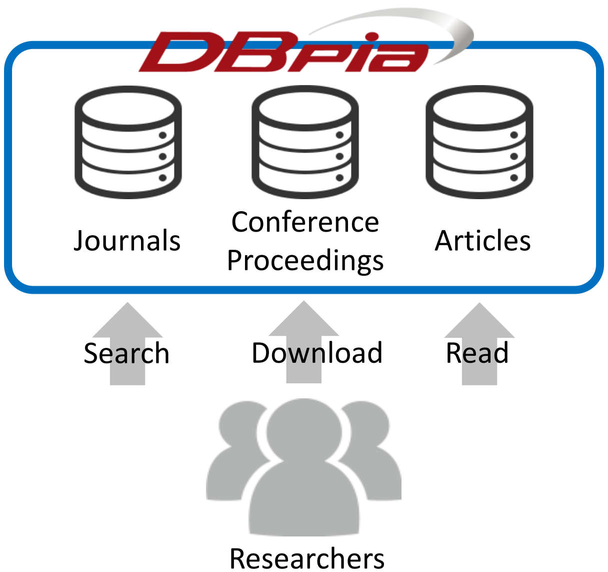
Different recommendation approaches

PaperRec (IEEE SMC 16)

Motivation

- Digital-bibliography service provider
  - Finding relevant literature to understand a trend of their research topic
  - Examples: *Google scholar, DBLP, DBpia*



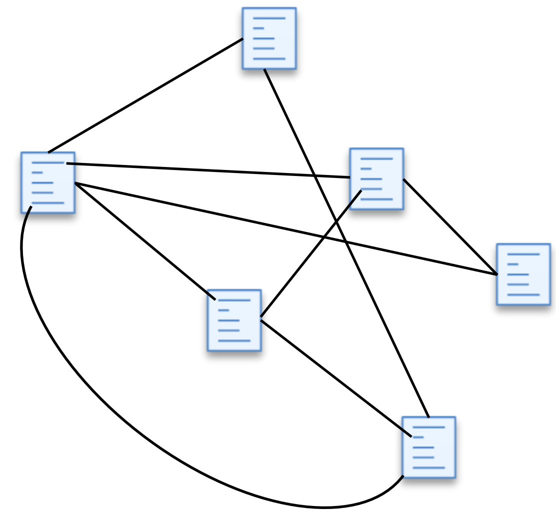


- Limitation of current *DBpia* services
  - Only provides *search functions*
  - Very *time-consuming task* for a researcher to find relevant papers
- Our proposal
  - *A hybrid approach* exploiting content and collaborative dataset

PaperRec (IEEE SMC 16)

☐ Our approach

- Four datasets used for recommendation
  - *Content information of each paper*
    - Consists of the title, the abstract, the body, and the keywords of the paper
  - *Citation relationships between papers*
    - A citation relationship of a paper represents a list of other papers cited by the paper
  - *Researcher-to-paper history*
    - For a researcher, it consists of information on other papers downloaded by the researcher
  - *Paper-to-paper history*
    - For a paper, it consists of information for other papers downloaded together with the paper by researchers



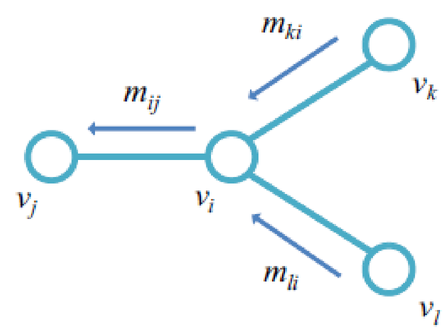
PaperRec (IEEE SMC 16)

Our approach

- Idea
  - To exploit both of content-based similarity and graph-based network
- Proposed Approach
  - Content-based approach
    - Based on content-based similarity between papers
    - Supplemented by information of similar users
  - Graph-based approach
    - Graph modeling using citation relationships between papers
    - To employ belief propagation (BP) algorithm that probabilistically determines a target user's preference on an item
  - *Hybrid approach*
    - To combine the results obtained from content-based or graph-based approaches

$$f_{hybrid} = w * f_{content} + (1 - w) * f_{graph}$$

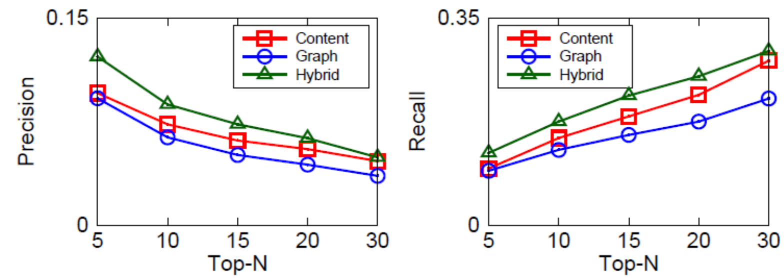
- w: weight of the content-based approach
- $(1 - w)$: weight of the graph-based approach



PaperRec (IEEE SMC 16)

Results

- *Utilizing both of content and graph information together* provides accuracy higher than those using two information sources independently



Different recommendation approaches

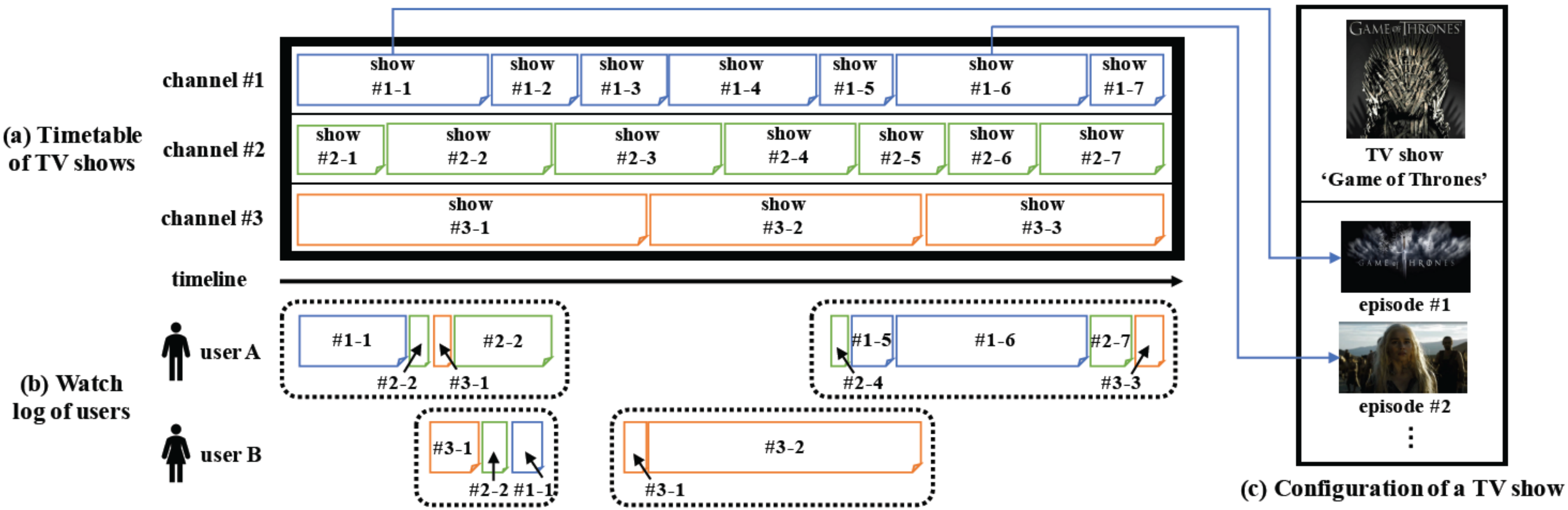
DBpia Insight

- http://insight.dbpia.co.kr/

TVshowRec (Submitted to IJCAI 18)

Motivation

- *A large number of channels* (>= 100 channels)
- Very *time-consuming task* for a user to find TV shows that they prefer



Data in a TV show recommendation

TVshowRec (Submitted to IJCAI 18)

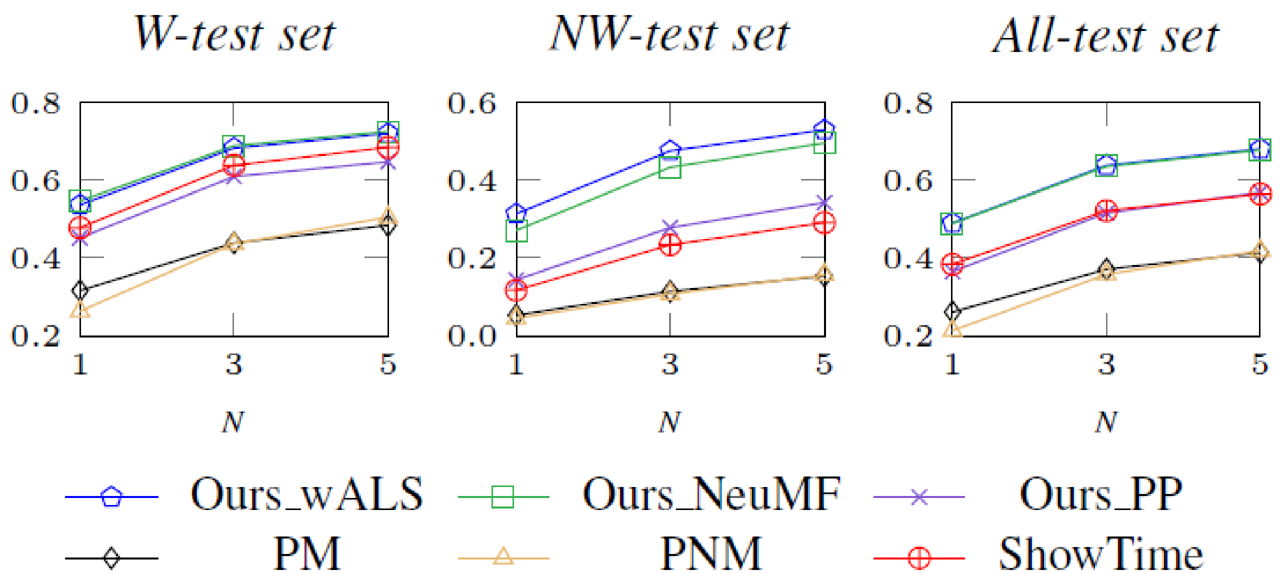
Our approach

- Core concepts
  - A user u's '*watching interval*'
    - range between the starting and ending times of u's watching TV shows
  - A user u's '*watchable interval*' for an episode
    - A certain timeframe that u can watch the episode within
    - An overlapped interval between u's 'watching interval' and the episode's broadcasting interval
  - A user u's '*watchable episodes*'
    - a set of u's determined by her 'watching interval'
    - episodes that u can give feedback to
- Idea
  - A user u's preference on an episode can be inferred through the analysis on *her feedback given in the range of her watchable interval for the episode*

TVshowRec (Submitted to IJCAI 18)

☐ Results

- Our proposed framework provides *the highest accuracy in all test sets and with all metrics*



Final recommendation accuracy in terms of nDCG

References

- gOCCF
  - Yeon-Chang Lee, Sang-Wook Kim, Dongwon Lee, "**gOCCF: Graph-Theoretic One-Class Collaborative Filtering Based on Uninteresting Items**", In AAAI 2018
- JobRec
  - Yujin Lee*, Yeon-Chang Lee*, Jiwon Hong, Sang-Wook Kim, "**Exploiting Job Transition Patterns for Effective Job Recommendation**", In IEEE SMC 2017 (*co-first authors with equal contribution)
  - Yeon-Chang Lee, Jiwon Hong, and Sang-Wook Kim, "**Job Recommendation in AskStory: Experiences, Methods, and Evaluation**," In ACM SAC 2016
  - Yeon-Chang Lee, Jiwon Hong, Sang-Wook Kim, Sheng Gao, and Ji-Yong Hwang, "**On Recommending Job Openings**," In ACM HT 2015
- PaperRec
  - Yeon-Chang Lee, Jungwan Yeom, Kiburm Song, Jiwoon Ha, Kichun Lee, Jangho Yeo, and Sang-Wook Kim, "**Recommendation of Research Papers in DBpia: A Hybrid Approach Exploiting Content and Collaborative Data**," In IEEE SMC 2016
- TVshowRec
  - Kyung-Jae Cho, Yeon-Chang Lee, Kyungsik Han, Sang-Wook Kim, "**TV Show Recommendation using Watchable Episodes**," In IJCAI 2018 (Submitted)