

Recommender Systems: Concepts, Techniques, and Applications

2020. 4. 9

Sang-Wook Kim

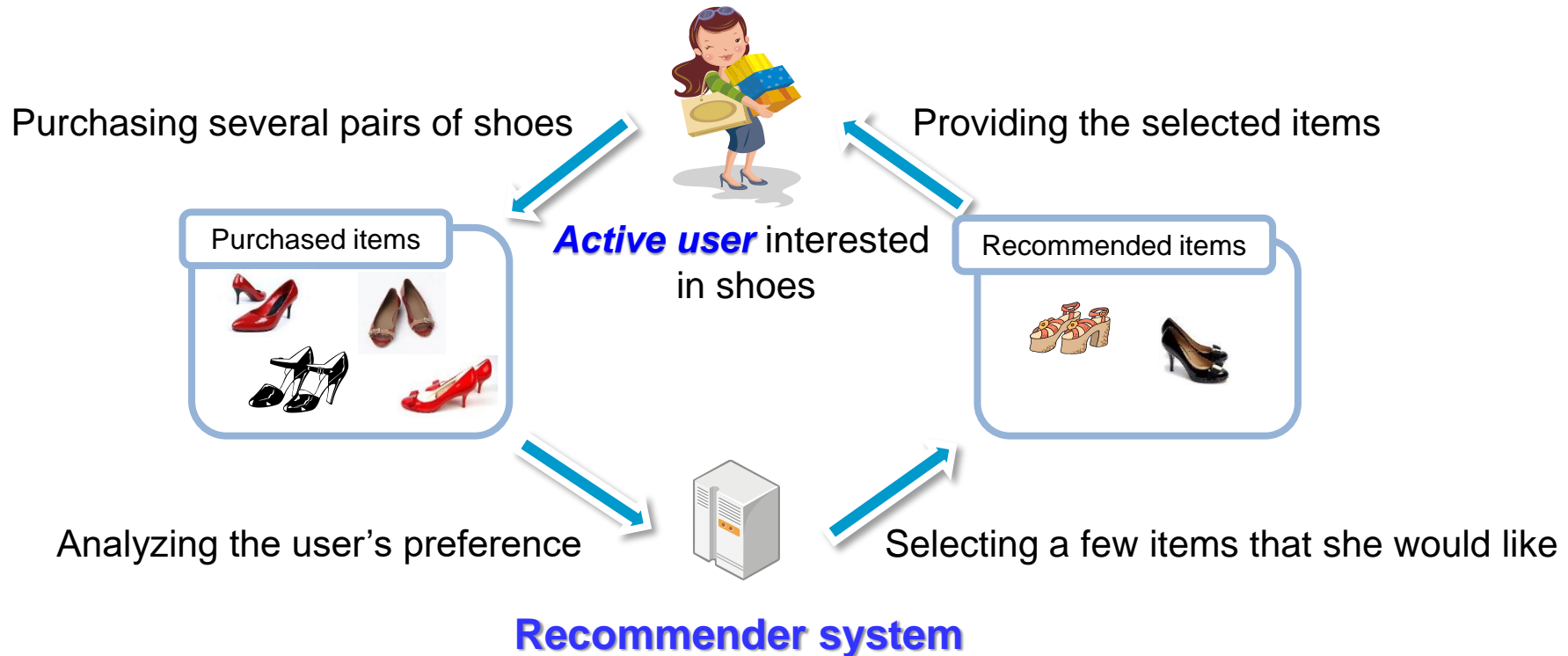
**Department of Computer Science and Engineering
Hanyang University**

- **Background**
 - **Recommendation systems**
 - **Collaborative filtering (CF)**
- Exploiting Uninteresting Items for Effective CF
 - Motivation
 - Proposed method
 - Evaluation
- Extensions
 - To data imputation
 - To one-class settings
- Conclusions











Recommendation Systems

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



Recommended for you, Thomas

| | | | |
|--|---|--|--|
|  <p>Literature & Fiction 80 ITEMS</p> |  <p>Exercise & Fitness Equipment 6 ITEMS</p> |  <p>Health, Fitness & Dieting Books 87 ITEMS</p> |  <p>Tableware 12 ITEMS</p> |
|  <p>Prime Video - Unlimited Streaming for Prime Members</p> |  <p>Coffee, Tea & Espresso 36 ITEMS</p> |  <p>Biographies & Memoirs 17 ITEMS</p> |  <p>Engineering Books 7 ITEMS</p> |

Your Recently Viewed Items and Featured Recommendations

Inspired by your browsing history

Page 1 of 8

| | | | | | | |
|--|--|---|---|---|--|---|
|  <p>Nanyo Gymnastic Rings for Full Body Strength and Crossfit Training ★★★★★ 1,274 \$29.87 Prime</p> |  <p>Champion Sports NCAA NPHS Certified Lacrosse Ball ★★★★★ 673 \$5.00 - \$11.19</p> |  <p>Rumble Roller Beastie Hook - Hand Held Massage Tool - Use With RumbleRoller Beastie... ★★★★★ 13 \$24.95 Prime</p> |  <p>Muscle Roller Ball Set :: Massage Balls for Deep Tissue, Trigger Point & Myofascial Release... ★★★★★ 237 \$14.97 Prime</p> |  <p>Foam Roller, LuxFit Premium High Density Foam Roller - Extra Firm With 1 Year Warranty ★★★★★ 1,421 \$8.75 - \$24.95</p> |  <p>2 X LACROSSE BALLS FOR TRIGGER POINT MASSAGE- Fine-Toned® plus MASSAGE... ★★★★★ 43 \$10.75 Prime</p> |  <p>The Dead Wedge - Deadlift Jack Alternative for Your Gym Bag - Raises loaded barbell & plates... ★★★★★ 355 \$17.95 Prime</p> |
|--|--|---|---|---|--|---|


<http://rejoiner.com/resources/amazon-recommendations-secret-selling-online/>

8:30 naver.com


백화점원도 아몰렛원도 스타일원도

AiTEMS 추천 ^{data} ① 패션 디지털 리빙


코트에 관심 있으세요? >




[현대백화점4관]타미진[TJMS1J...]
139,830원




[현대백화점4관]지오다노] 077917...
107,070원




오버 울 로브코트
104,000원



헤밀턴 싱글 트렌치 코트
76,500원



[크리스마스]글렌체크발미안...
209,300원



버버리체크 맥코트
106,000원

🔄 추천 상품 새로 보기

멘즈패션 훈내나는 멘즈 스타일

단독20%할인



디자이너 폴로버 셔츠 7만 원
대 행사 중

SALE



폴로 랄프로렌 강엄 블랙
원대

8:32 naver.com

백화점원도 아몰렛원도 스타일원도

AiTEMS 추천 ^{data} ① 패션 디지털 리빙

마음에 드는 청바지가 있으신가요? >



캘빈클라인진 남성 그레이 슬림 스...
99,000원



[캘빈클라인진]17년 F / W [남] 스...
159,930원



[캘빈클라인진] 남성 미드블루톤...
59,600원



캘빈클라인진 남성 바디 스키니진 4...
39,000원



캘빈클라인진 남성 Rocker Blue 스...
79,000원



캘빈클라인진 남성 다크블루 스키...
114,000원

🔄 추천 상품 새로 보기


멘즈패션 훈내나는 멘즈 스타일

단독20%할인



디자이너 폴로버 셔츠 7만 원
대 행사 중

SALE



폴로 랄프로렌 강엄 블랙
원대



<https://www.netflix.com/>

9 April 2020

Watcha interface showing movie recommendations. The header includes the Watcha logo, navigation links (박스오피스, 추천영화, 평가블리키), a search bar, and a user profile section.

The main content area displays a grid of movie posters with their respective scores and recommendations:

- RESIDENT EVIL**: Score 3.6. Recommendation: <호성탈출: 진화의 시작>과 비슷해요.
- 8 MILE**: Score 3.5. Recommendation: 좋아하는 태그 OST.
- 이퀄리브리엄**: Score 3.8. Recommendation: <브이 포 벤데타>와 비슷해요.
- SHINING**: Score 3.7. Recommendation: <달 포인트>와 비슷해요.
- 나우시카**: Score 3.7. Recommendation: 예상별점 3.7개 ★★★★★.
- 코러스**: Score 4.0. Recommendation: <꽃피는 봄이 오면>과 비슷해요.
- 퓨리**: Score 3.5. Recommendation: <고지전>과 비슷해요.
- 노킹온 헤븐스도어**: Score 4.1. Recommendation: <인투 더 와일드>와 비슷해요.
- 박하사탕**: Score 3.9. Recommendation: <회려한 휴가>와 비슷해요.
- 달라스 바이어스 클럽**: Score 3.8. Recommendation: <킹스피치>와 비슷해요.
- 4.2**: Recommendation: <4.2>.
- 3.1**: Recommendation: <3.1>.
- 4.3**: Recommendation: <4.3>.
- 3.4**: Recommendation: <3.4>.
- 4.1**: Recommendation: <4.1>.

<https://watcha.net/recommendation>

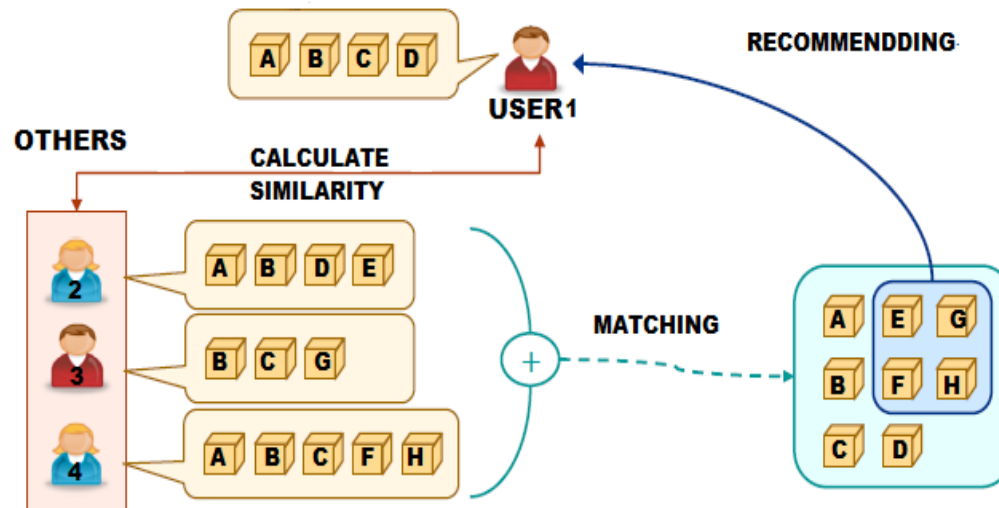
9 April 2020

Classification of Recommendation Systems

- Content-based approach
 - Recommending those items that **have similar contents** to those of the active user's favorite items
- **Collaborative filtering (CF) approach (our focus)**
 - Recommending items **rated high by neighbors** who have preferences similar to that of the active user
- Trust-based approach
 - Recommending items based on **trust relationships among users**
- Hybrid approach
 - Recommending items by **combining** the approaches above

Collaborative Filtering (CF) Approach

- Recommend such items **rated high by neighbors** who have preferences similar to that of the active user
 - Step 1: *Finding a group of users (neighbors)* whose preferences are similar to that of an **active user c**
 - Step 2: *Estimating $r_{c,s}$, the rating of item s for active user c* , based on the ratings given to item s by c 's neighbors
 - Step 3: Recommending *a few items with the ratings estimated high*

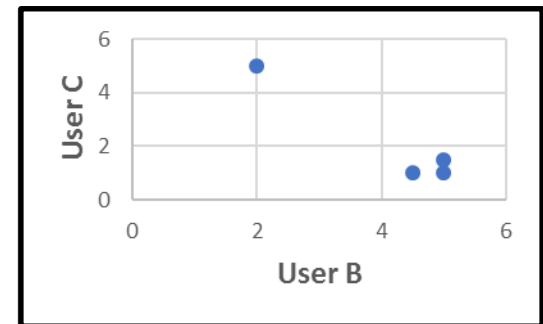
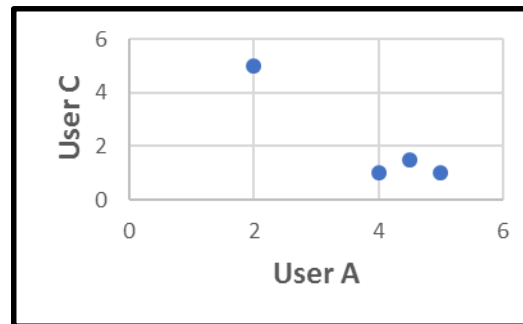
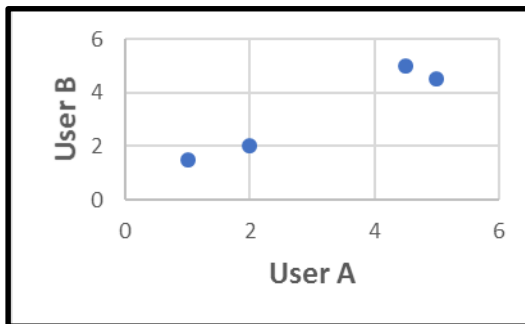


Heuristic-based Method in CF: Step 1

Hanyang Univ.

- Similarity measure for users a and i (used in finding neighbors)
 - Example: three users' ratings for six movies
 - User A = <4.0, 1.0, 4.5, 5.0, 2.0, _>
 - User B = <_, 1.5, 5.0, 4.5, 2.0, 5.0>
 - User C = <1.0, _, 1.5, 1.0, 5.0, 1.0>
 - Pearson correlation coefficient (PCC)

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



Heuristic-based Method in CF: Step 2

- Aggregation of ratings on a target item given by the neighbors

$$r_{c,s} = \text{aggr}_{c' \in \hat{C}} r_{c',s}$$

- $r_{c,s}$: Estimated rating on item s for user c
- \hat{C} : Set of neighbors for user c
- Different methods for aggregation

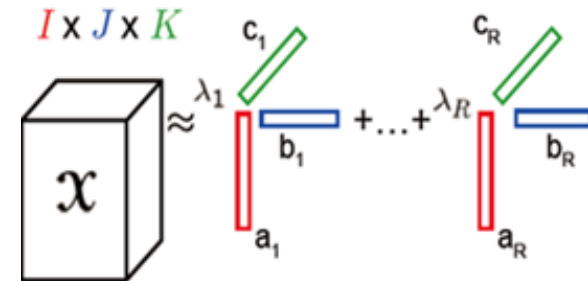
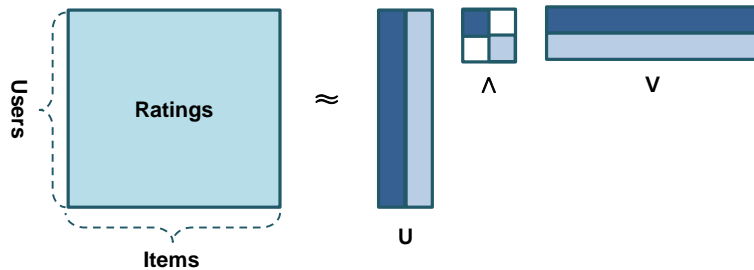
$$(a) r_{c,s} = \frac{1}{N} \sum_{c' \in \hat{C}} r_{c',s}$$

$$(b) r_{c,s} = k \sum_{c' \in \hat{C}} \text{sim}(c, c') \times r_{c',s}$$

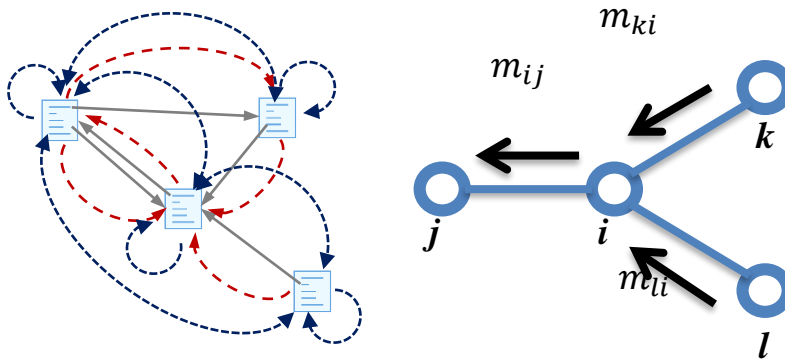
$$(c) r_{c,s} = \bar{r}_c + k \sum_{c' \in \hat{C}} \text{sim}(c, c') \times (r_{c',s} - \bar{r}_{c'})$$

Machine-Learning Based CF Techniques

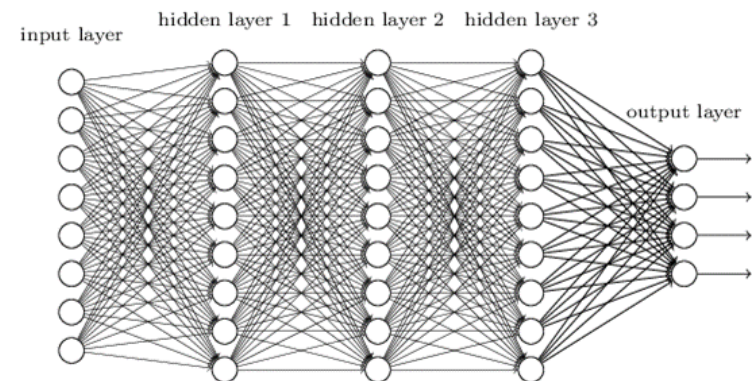
- Matrix/Tensor Factorization



- Social Network Analysis



- Deep Learning



Matrix Factorization (MF)

$$\begin{array}{c} \text{User} \left\{ \begin{array}{c} \begin{array}{c} \text{Item} \\ \begin{array}{ccccc} \boxed{5} & 2 & \text{blue} & \dots & 1 \\ 4 & \text{blue} & 4 & \dots & \text{blue} \\ 3 & 3 & \text{blue} & \dots & \text{blue} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 5 & \text{blue} & \text{blue} & \dots & 3 \end{array} \\ \mathbf{R} \end{array} \end{array} \approx \begin{array}{c} \text{User} \left\{ \begin{array}{c} \begin{array}{c} f \\ \begin{array}{ccc} \boxed{2.7} & \dots & \boxed{0.5} \\ 0.9 & \dots & 0.3 \\ 0.7 & \dots & 1.5 \\ \vdots & \ddots & \vdots \\ 1.9 & \dots & 0.2 \end{array} \end{array} \end{array} \times \begin{array}{c} f \left\{ \begin{array}{c} \text{Item} \\ \begin{array}{ccccc} \boxed{0.3} & 0.7 & 1.5 & \dots & 1.2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1.6 & 0.5 & 0.1 & \dots & 2.5 \end{array} \\ \mathbf{V}^T \end{array} \end{array}\end{array}$$

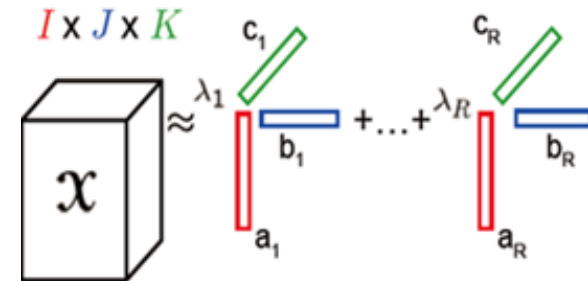
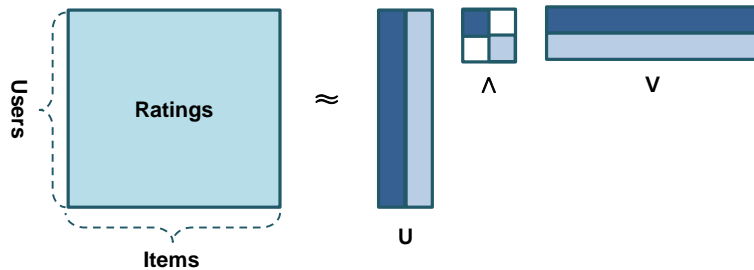
Matrix Factorization (MF)

$$\begin{array}{c} \text{User} \\ \left[\begin{array}{ccc} 2.7 & \dots & 0.5 \\ \boxed{0.9} & \dots & \boxed{0.3} \\ 0.7 & \dots & 1.5 \\ \vdots & \ddots & \vdots \\ 1.9 & \dots & 0.2 \end{array} \right] \end{array} \times \begin{array}{c} \text{Item} \\ \left[\begin{array}{ccccc} 0.3 & \boxed{0.7} & 1.5 & \dots & 1.2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1.6 & \boxed{0.5} & 0.1 & \dots & 2.5 \end{array} \right] \end{array} = \begin{array}{c} \text{User} \\ \left[\begin{array}{ccccc} 4.9 & 1.5 & \boxed{3.5} & \dots & 1.2 \\ 3.5 & \boxed{2.4} & 4.0 & \dots & \boxed{1.9} \\ 3.0 & 3.5 & \boxed{4.5} & \dots & \boxed{0.5} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 4.8 & \boxed{1.3} & \boxed{3.4} & \dots & 3.2 \end{array} \right] \end{array}$$

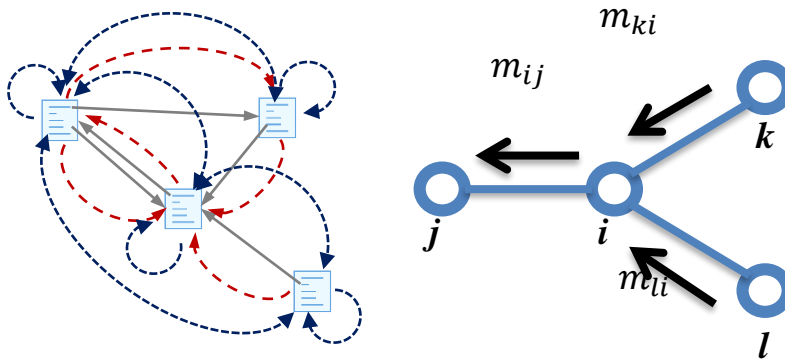
$U \qquad V^T \qquad \hat{R}$

Machine-Learning Based CF Techniques

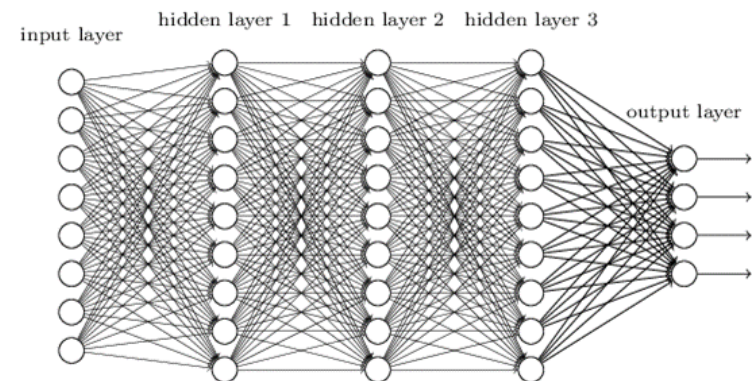
- Matrix/Tensor Factorization



- Social Network Analysis



- Deep Learning



- Background
 - Recommendation systems
 - Collaborative filtering (CF)
- **Exploiting Uninteresting Items for Effective CF**
 - **Motivation**
 - **Proposed method**
 - **Evaluation**
- Extensions
 - To data imputation
 - To one-class settings
- Conclusions



“Told You I Didn’t Like It”: Exploiting Uninteresting Items for Effective Collaborative Filtering

(IEEE ICDE’16 / IEEE TKDE’19)



- CF approaches focus on *only the ratings given by users*
 - Data sparsity problem: most users have evaluated *only a few items*
 - There are only a few ratings in a rating matrix ($< 4\%$)
- CF approaches suffer from *low accuracy and coverage*

| | | Items | | | | | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|----------|----------|
| | | i_1 | i_2 | i_3 | i_4 | i_5 | i_6 | i_7 | i_8 | i_9 | i_{10} | i_{11} | i_{12} |
| Users | u_1 | | 1 | | 1 | | 5 | 3 | | | 5 | | |
| | u_2 | 1 | 3 | | 4 | | 4 | | | | 2 | 4 | |
| | u_3 | | | | | | | | | | | | 5 |
| | u_4 | | 3 | | 4 | | 4 | | 4 | 3 | | | |
| | u_5 | | | 4 | | | | | | | | | |
| | u_6 | 1 | | | 3 | 5 | 1 | | 4 | | | 3 | |

A rating on i_1
given by user u_6

A Rating Matrix

Our Idea: Using Unrated Items

- Exploit **unrated** items
 - A fraction of rated items in a rating matrix R is extremely small ($< 4\%$)
 - Exploiting the vast number of “unrated” items in R can lead to a significant improvement in CF approaches

| | | Items | | | | | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|----------|----------|
| | | i_1 | i_2 | i_3 | i_4 | i_5 | i_6 | i_7 | i_8 | i_9 | i_{10} | i_{11} | i_{12} |
| Users | u_1 | | 1 | | 1 | | 5 | 3 | | | 5 | | |
| | u_2 | 1 | 3 | | 4 | | 4 | | | | 2 | 4 | |
| | u_3 | | | | | | | | | | | | 5 |
| | u_4 | | 3 | | 4 | | 4 | | 4 | 3 | | | |
| | u_5 | | | 4 | | | | | | | | | |
| | u_6 | 1 | | | 3 | 5 | 1 | | 4 | | | 3 | |

unrated items

Our Idea: Uninteresting Items (ICDE'16 / TKDE'19)

Hanyang Univ.



- Unrated items
 - Users were not aware of their existence
 - Candidates for recommendation
 - Users knew but did not like and thus did not purchase
 - *Uninteresting items*
- Uninteresting items (of a user)
 - Items on which the user has “negative” preferences



Users' Preferences: Two New Notions

- **Pre-use preferences**

- User's impression on items **before** purchasing and using them
- Determined **via meta data** of items (known before purchasing)

- **Post-use preferences**

- User's impression on items **after** purchasing and using them
- Determined **via real content** of items (unknown before purchasing)



Pre-Use/Post-Use Preferences and Ratings

- A user has high pre-use preference for Movie #1 and Movie #2
 - The user likes Movie #1 but is disappointed at Movie #2
- A user has low pre-use preference for Movie #3

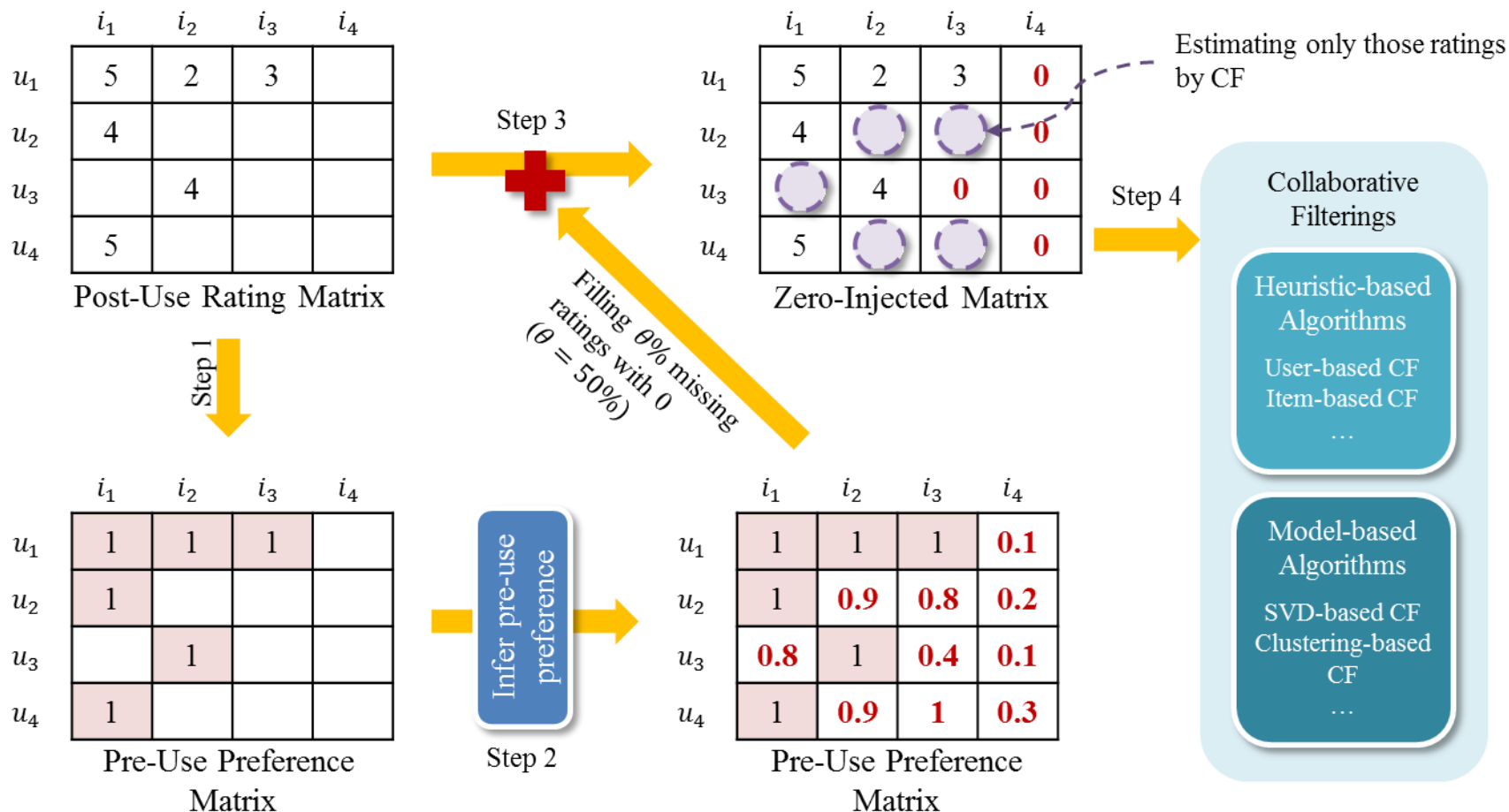
| Movie | Pre-use preference | Post-use preference | Rating |
|---------|--------------------|---------------------|---------|
| Movie#1 | High | High | 5 |
| Movie#2 | High | Low | 1 |
| Movie#3 | Not high | Unknown | Unrated |

- In this case
 - Uninteresting items: Movie #3
 - Interesting items: Movie #1 (preferred) and Movie #2 (not preferred)

Pre-Use Preference and Uninteresting Items

- Challenge: To identify **uninteresting items** among unrated items
- A user's uninteresting items
 - Her pre-use preferences on them are **relatively low**
- How to know a user's pre-use preferences
 - For all “rated” items: **highest pre-use preferences**
 - Otherwise, users would not have bought them in the first place
 - For unrated items: **pre-use preferences need to be inferred**
 - Based on pre-use preferences on rated items

Zero-Injection (IEEE ICDE'16 / IEEE TKDE'19)

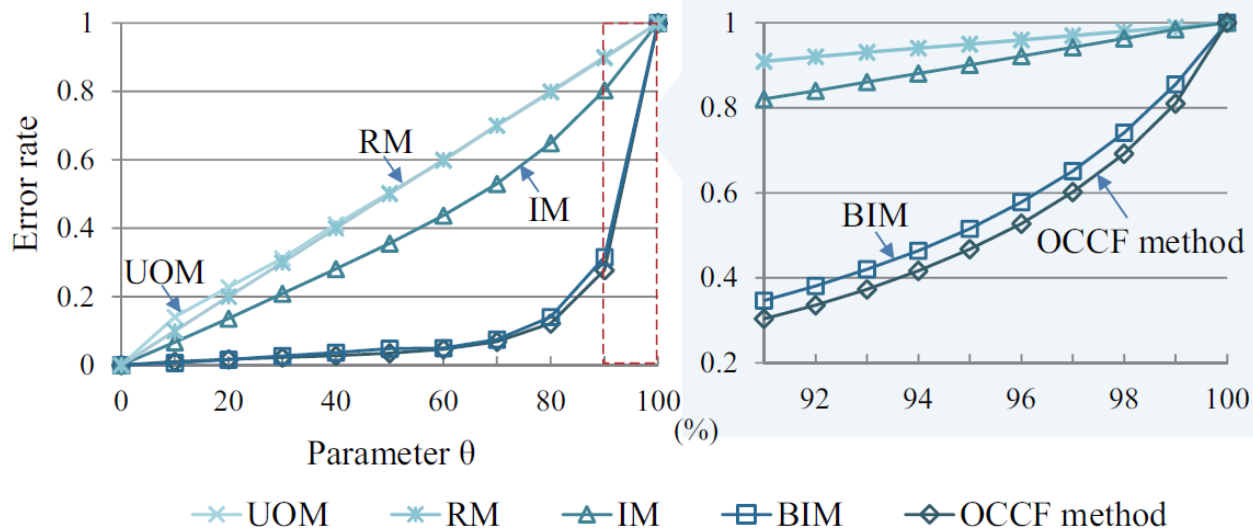


Exploiting Uninteresting items

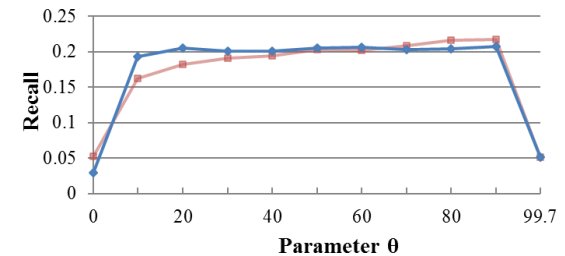
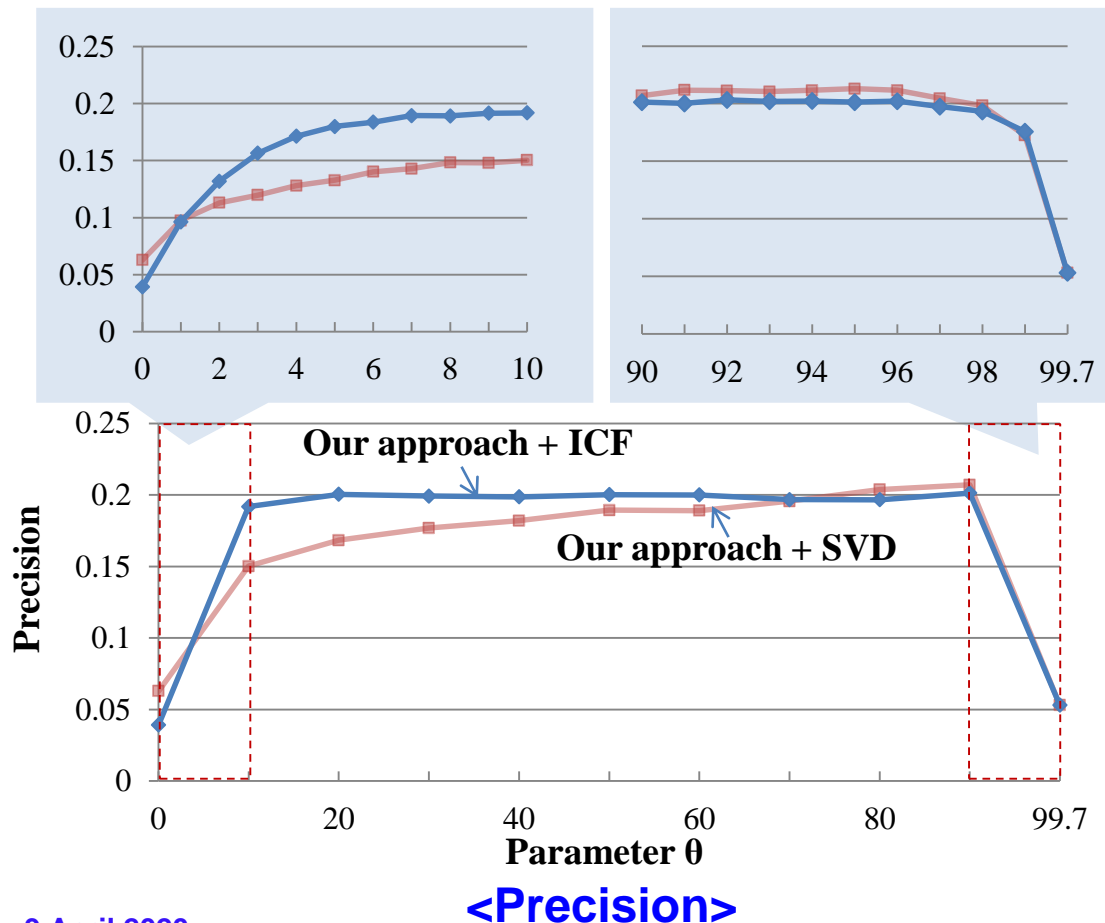
- Two strategies with uninteresting items
 - Strategy 1: to exclude uninteresting items from the final recommendation list
 - Strategy 2: to exploit both uninteresting items and ratings in CF

Accuracy in Finding Uninteresting items

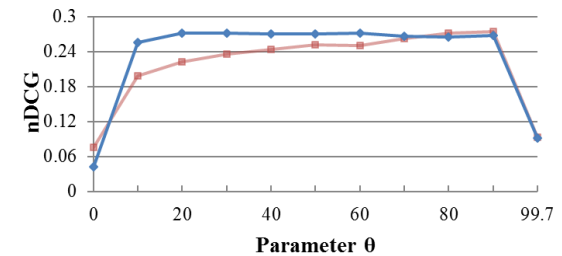
- Error rate comparison of five inference methods
 - A user u 's error rate (with 5 cross validation)
 - How many *rated items* (in the test set) are selected as uninteresting items (*i.e.*, mis-classified) for u
 - $err_u^\theta = \frac{|I_u^{un}(\theta) \cap I_u^{test}|}{|I_u^{test}|}$
 - The OCCF method is the most effective



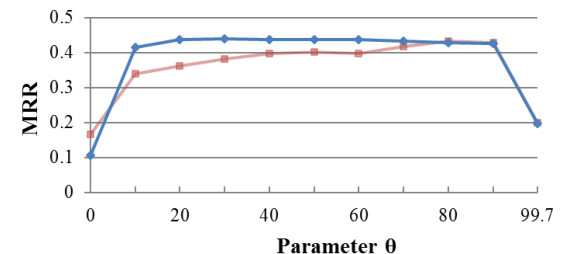
- Accuracy of ICF and SVD equipped with our zero-injection under varying parameter θ



<Recall>



<nDCG>



<MRR>

- Accuracy of four CF methods equipped with zero-injection ($\theta = 90\%$)

| Metric | | ICF | | | SVD | | | SVD++ | | | PureSVD | | |
|-----------|-----|----------|-------|--------|----------|-------|--------|----------|-------|--------|----------|-------|-------|
| | | Original | Ours | Gain | Original | Ours | Gain | Original | Ours | Gain | Original | Ours | Gain |
| Precision | @5 | 0.039 | 0.201 | 413.8% | 0.063 | 0.207 | 229.7% | 0.076 | 0.193 | 153.3% | 0.100 | 0.106 | 16.7% |
| | @10 | 0.041 | 0.161 | 292.6% | 0.056 | 0.166 | 196.9% | 0.069 | 0.154 | 123.9% | 0.082 | 0.089 | 19.1% |
| | @15 | 0.040 | 0.137 | 243.7% | 0.053 | 0.142 | 169.9% | 0.063 | 0.134 | 112.0% | 0.071 | 0.078 | 11.3% |
| | @20 | 0.039 | 0.121 | 211.7% | 0.048 | 0.125 | 159.1% | 0.058 | 0.118 | 102.3% | 0.063 | 0.071 | 13.7% |
| Recall | @5 | 0.030 | 0.207 | 600.3% | 0.052 | 0.218 | 316.0% | 0.063 | 0.194 | 209.6% | 0.112 | 0.120 | 16.9% |
| | @10 | 0.059 | 0.305 | 412.7% | 0.089 | 0.325 | 265.9% | 0.109 | 0.288 | 163.1% | 0.175 | 0.191 | 19.3% |
| | @15 | 0.085 | 0.375 | 341.4% | 0.121 | 0.394 | 226.3% | 0.150 | 0.361 | 141.2% | 0.220 | 0.245 | 11.4% |
| | @20 | 0.111 | 0.428 | 285.4% | 0.144 | 0.450 | 213.5% | 0.184 | 0.415 | 125.3% | 0.254 | 0.293 | 15.4% |
| nDCG | @5 | 0.043 | 0.268 | 527.9% | 0.076 | 0.274 | 260.7% | 0.087 | 0.256 | 196.0% | 0.135 | 0.143 | 16.0% |
| | @10 | 0.053 | 0.285 | 436.0% | 0.084 | 0.297 | 252.0% | 0.099 | 0.272 | 175.6% | 0.151 | 0.162 | 17.6% |
| | @15 | 0.062 | 0.303 | 390.7% | 0.094 | 0.315 | 234.8% | 0.110 | 0.291 | 163.7% | 0.164 | 0.178 | 18.9% |
| | @20 | 0.071 | 0.319 | 351.7% | 0.101 | 0.332 | 227.3% | 0.121 | 0.306 | 153.5% | 0.174 | 0.193 | 10.9% |
| MRR | | 0.106 | 0.426 | 303.0% | 0.165 | 0.428 | 159.2% | 0.181 | 0.416 | 129.3% | 0.262 | 0.274 | 14.7% |

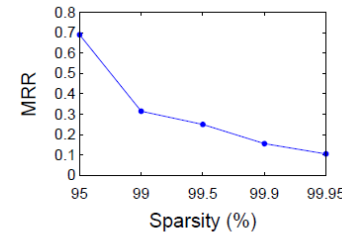
- Background
 - Recommendation systems
 - Collaborative filtering (CF)
- Exploiting Uninteresting Items for Effective CF
 - Motivation
 - Proposed method
 - Evaluation
- **Extensions**
 - **To data imputation**
 - **To one-class settings**
- Conclusions



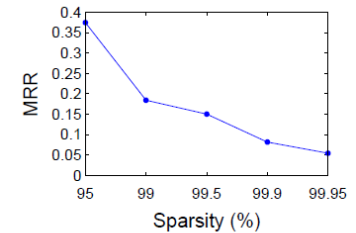
gOCCF: Graph-Theoretic One-Class Collaborative Filtering Based on Uninteresting Items (AAAI'18)

- **One-Class** Collaborative Filtering (OCCF)
 - To handle implicit feedback (*i.e.*, one-class setting, rather than ratings setting)
 - Example: Click, bookmark, and purchase
 - Challenges
 - *Less information to capture a user's taste* than in ratings setting
 - *Sparser datasets* than in ratings setting

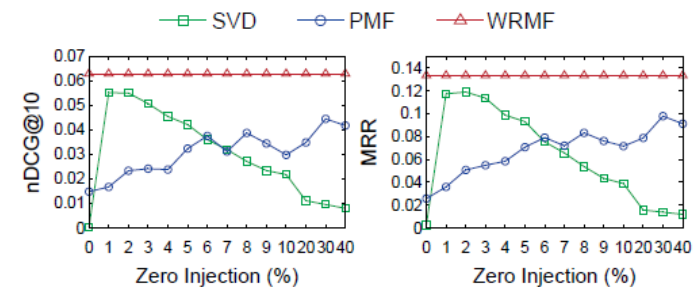
- Existing OCCF approaches
 - Less effective in dealing with sparser datasets with many unrated items
- A naïve application of zero-injection in one-class setting
 - Lower accuracy than existing OCCF approaches
 - Sensitivity to the number of uninteresting items



(a) MovieLens 100K



(b) Watcha

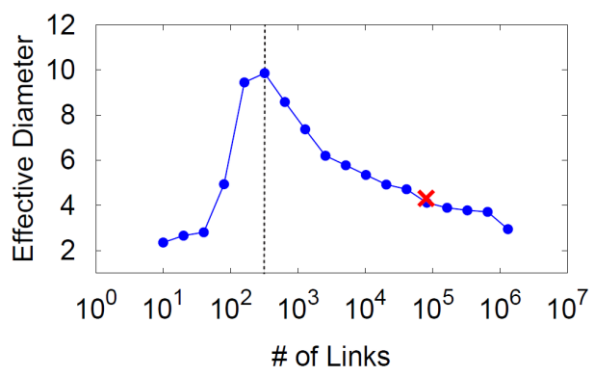


Our Idea: Exploiting Graph Properties (AAAI'18)

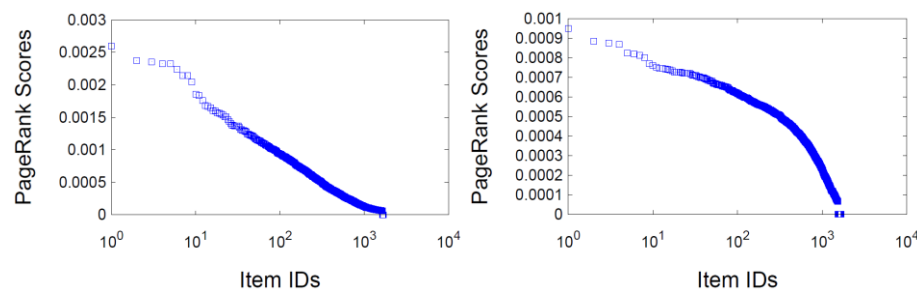


Hanyang Univ.

- Challenge: To determine *a right number of uninteresting items* with the best accuracy
- Analyze properties of negative graphs constructed by **user-uninteresting item pairs**
 - Topological properties
 - Use *graph shattering theory*
 - Information propagation
 - Use PageRank scores



(1) ShatterPlot

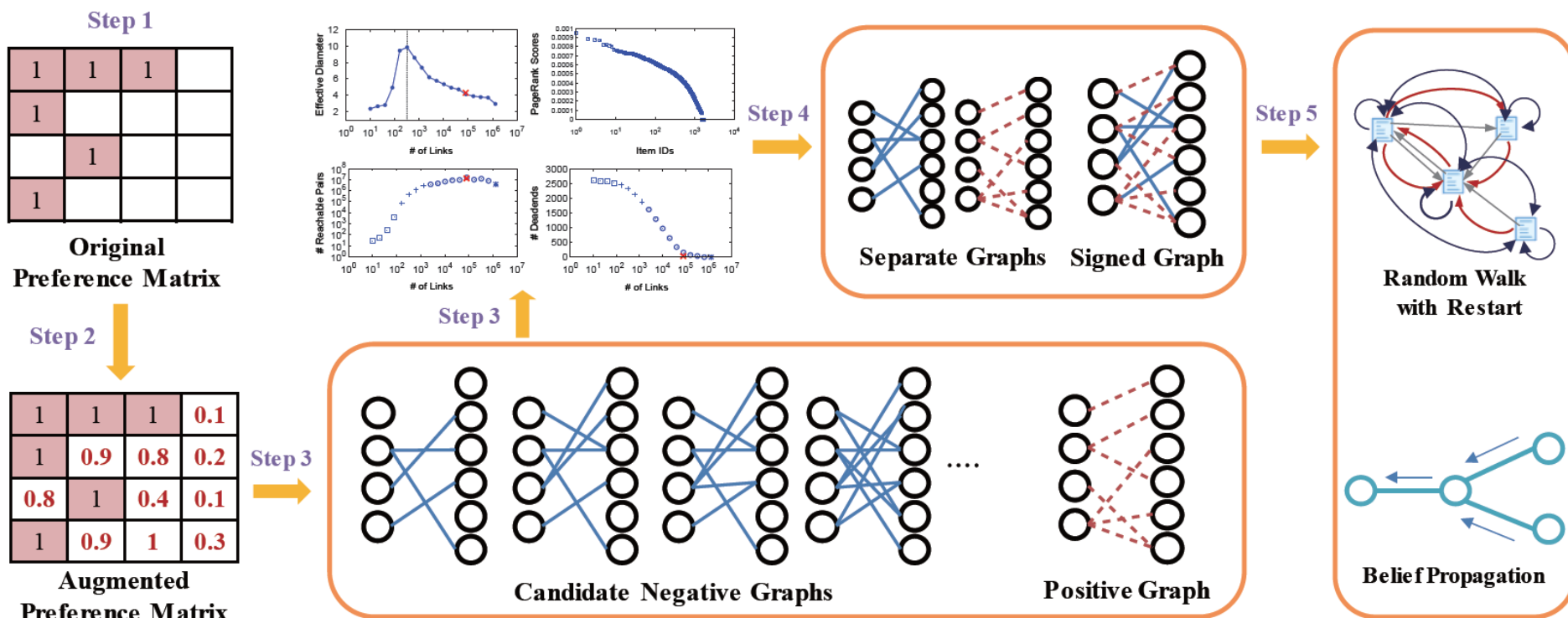


(a) Real Positive Graph

(b) Negative Graph

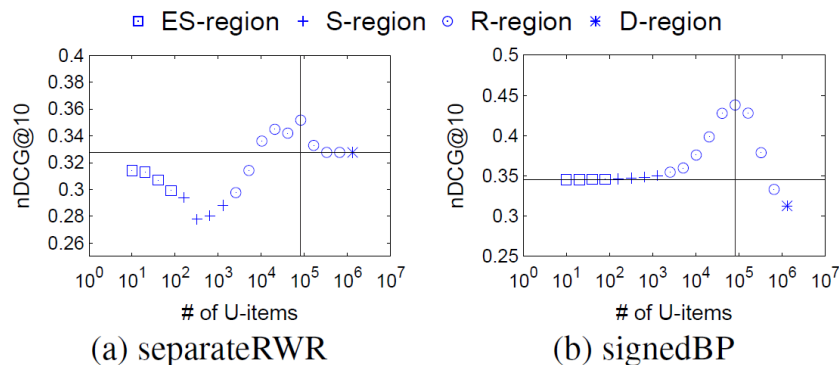
(2) PageRank Scores

Overview of Our Approach: gOCCF (AAAI'18)



Effectiveness of gOCCF (AAAI'18)

- Accuracy according to the number of uninteresting items
- Accuracy before and after exploiting uninteresting items



- Best accuracy when having the same number of negative links as that of positive links

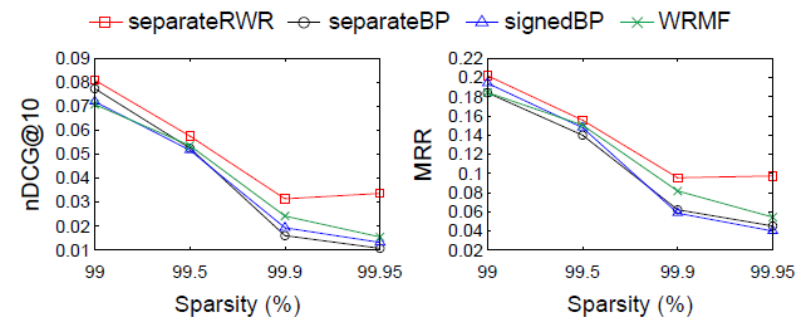
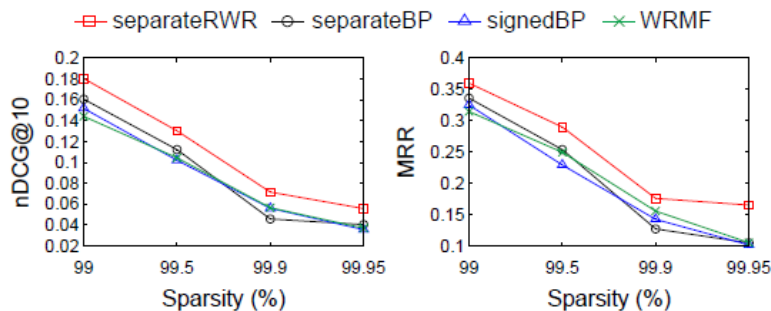
| Metrics | MovieLens | | |
|---------|----------------|----------------|----------------|
| | separateRWR | separateBP | signedBP |
| P@10 | 0.302 (9.2%) | 0.309 (7.2%) | 0.370 (28.2%) |
| R@10 | 0.171 (8.2%) | 0.164 (10.9%) | 0.210 (42.1%) |
| nDCG@10 | 0.352 (7.3%) | 0.365 (5.7%) | 0.438 (27.0%) |
| MRR | 0.584 (2.5%) | 0.604 (1.9%) | 0.679 (14.6%) |
| HLU | 43.894 (2.8%) | 47.195 (1.4%) | 55.980 (20.3%) |
| | Watcha | | |
| | separateRWR | separateBP | signedBP |
| P@10 | 0.113 (10.8%) | 0.124 (13.2%) | 0.151 (38.3%) |
| R@10 | 0.107 (9.6%) | 0.113 (13.4%) | 0.142 (41.9%) |
| nDCG@10 | 0.136 (9.9%) | 0.152 (13.2%) | 0.190 (41.8%) |
| MRR | 0.295 (7.1%) | 0.329 (10.8%) | 0.391 (32.0%) |
| HLU | 14.288 (7.7%) | 17.448 (20.4%) | 23.012 (58.8%) |
| | CiteULike | | |
| | separateRWR | separateBP | signedBP |
| P@10 | 0.122 (21.0%) | 0.112 (26.9%) | 0.091 (2.9%) |
| R@10 | 0.199 (13.5%) | 0.162 (26.4%) | 0.131 (2.4%) |
| nDCG@10 | 0.202 (18.2%) | 0.175 (29.9%) | 0.138 (2.3%) |
| MRR | 0.326 (14.0%) | 0.295 (21.0%) | 0.247 (1.3%) |
| HLU | 21.257 (23.0%) | 19.323 (33.3%) | 14.680 (1.2%) |

Effectiveness of gOCCF (AAAI'18)

- Accuracy of competing methods and our approach
 - CiteULike dataset

| | 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 |

- Accuracy of WRMF and our approach per sparsity
 - MovieLens 100K dataset
 - Watcha dataset



- We propose a *new concept of uninteresting items* to make more accurate CF for both *ratings and one-class settings*
- Our approach
 - To identify uninteresting items
 - To apply them to existing CF methods
 - To exclude them from the final recommendation list
- Strengths of our approach
 - *Orthogonal to* CF methods
 - *Parameter-free* for both *ratings and one-class settings*
 - *Consistently and universally improves* existing CF and OCCF methods

How to Impute Missing Ratings? Claims, Solution, and Its Application to Collaborative Filtering (WWW'18)



CFGAN: A Generic Collaborative Filtering Framework based on GAN (ACM CIKM'18)



RAGAN: Rating Augmentation with GAN towards Accurate Collaborative Filtering (WWW'19)





No, That's Not My Feedback: TV Show Recommendation Using Watchable Interval (IEEE ICDE'19)



RealGraph: A Graph Engine Leveraging the Power-Law Distribution of Real-World Graphs (WWW'19)



- Background
 - Recommendation systems
 - Collaborative filtering (CF)
- Exploiting Uninteresting Items for Effective CF
 - Motivation
 - Proposed method
 - Evaluation
- Extensions
 - To data imputation
 - To one-class settings
- Conclusions

Thank You !

