Individual Items Meet User Generated Lists

Da Cao[1,2], Liqiang Nie[3], Xiangnan He[4], Xiaochi Wei[5],
Shunzhi Zhu[2], Shunxiang Wu[1], Tat-Seng Chua[4]

1. Xiamen University; 2. Xiamen University of Technology; 3. Shandong University;
4. National University of Singapore; 5. Beijing Institute of Technology

40th SIGIR 2017 TOKYO • JAPAN

Outline

- Background
- Proposed Method
- Experiments and Results
- Conclusion

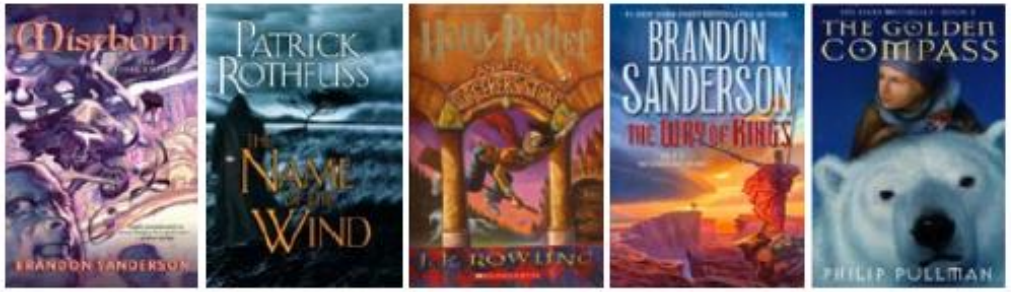
User Generated Booklists

My Top 2013-2014 Romance Reads

100 books — 9 voters

Mental Hospital Novels

158 books — 491 voters



Most Interesting Magic System

1,393 books — 4,603 voters

Best French Literature

467 books — 907 voters

User Generated Playlists

♥ Playlist A		
01	♥	Song A
02	♡	Song B
03	♥	Song C
04	♥	Song D
05	♡	Song E

♡ Playlist B		
01	♡	Song F
02	♡	Song G
03	♥	Song H
04	♡	Song I
05	♡	Song J

♥ Playlist C		
01	♥	Song C
02	♡	Song F
03	♥	Song A
04	♥	Song D
05	♥	Song H

The illustrations of 1) a user’s preference over lists; 2) the user’s preference over items within lists; and 3) relationships among items and lists.

To the best of our knowledge

- User generated list recommendation task

Second

- Factorization approaches & embedding-based algorithms

First

Challenges

- The relationship among items within a list
- New-item cold-start
- User-item and user-list recommendation

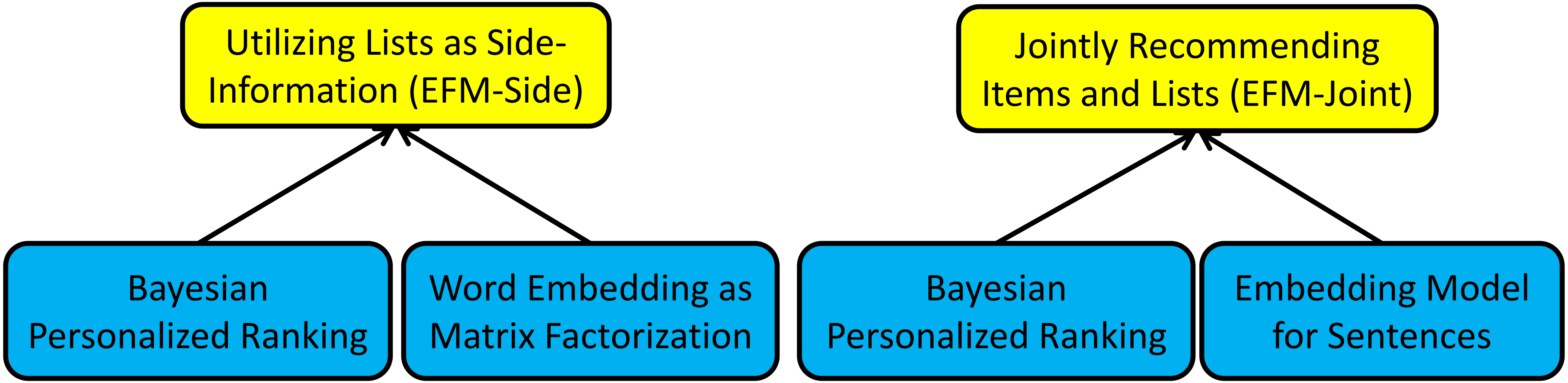
Outline

- Background
- Proposed Method
- Experiments and Results
- Conclusion

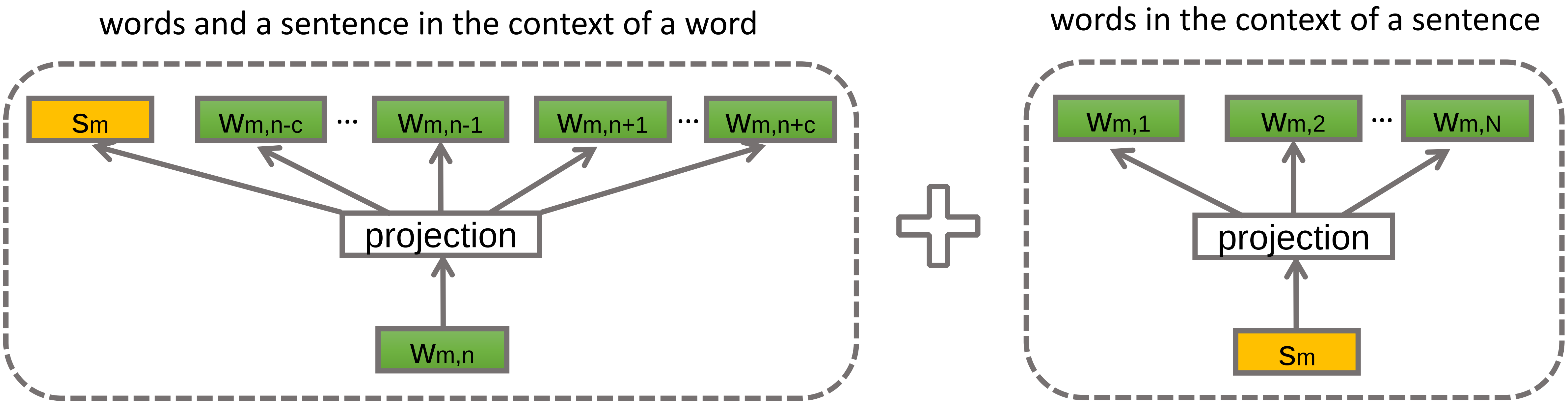
Proposed Method

- Bayesian Personalized Ranking [Rendle et al. 2009]
- Word Embedding as Matrix Factorization [Levy et al. 2014]
- Embedding Model for Sentences
- Utilizing Lists as Side-Information
- Jointly Recommending Items and Lists

Framework


Utilizing Lists as Side-Information (EFM-Side)
Bayesian Personalized Ranking
Word Embedding as Matrix Factorization
Jointly Recommending Items and Lists (EFM-Joint)
Bayesian Personalized Ranking
Embedding Model for Sentences

Sentence2vec



$$\mathrm{PMI}(i, j) = \log \frac{\#(i, j) \cdot \mathcal{D}}{\#(i) \cdot \#(j)} \quad \text{where } \#(i) = \sum_j \#(i, j),\ \#(j) = \sum_i \#(i, j), \text{ and } \mathcal{D} = \sum_{ij} \#(i, j).$$

Utilizing Lists as Side-Information

$$\mathcal{L}_{\mathrm{EFM-Side}} = \sum_{(u,i,j)\in\mathcal{D}_T} \ln\sigma(\hat{x}_{uij}) + \lambda_s \sum_{(i,j,t)\in\mathcal{D}_S} \ln\sigma(\hat{m}_{ijt}) - \lambda_\Theta ||\Theta||$$

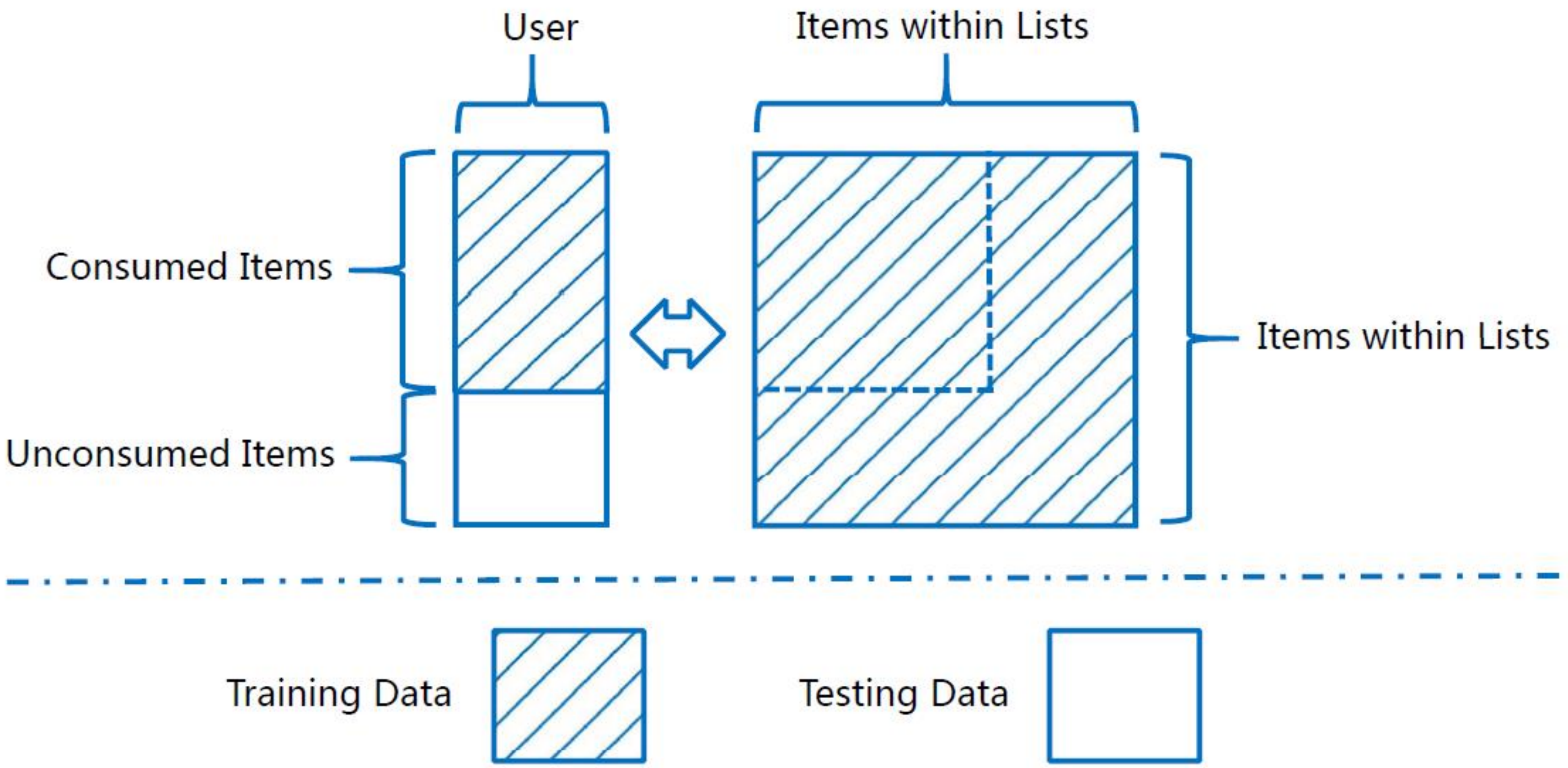
where $\mathcal{D}_T$ is the training corpus for the user-item entry $\hat{x}_{ui}$; $\mathcal{D}_S$ is the training corpus for the item-item entry $\hat{m}_{ij}$; λ_s balances the performance between BPR and word2vec's equivalent matrix factorization.

Jointly Recommending Items and Lists

$$\mathcal{L}_{\mathrm{EFM-Joint}} = \sum_{(u,i,j)\in\mathcal{D}_T} \ln\sigma(\hat{x}_{uij}) + \sum_{(u,i,j)\in\mathcal{D}_L} \ln\sigma(\hat{y}_{uij}) + \lambda_r \sum_{(i,j,t)\in\mathcal{D}_R} \ln\sigma(\hat{r}_{ijt}) - \lambda_{\Theta}||\mathbf{\Theta}||$$

where $\mathcal{D}_T$ is the training corpus for the user-item entry $\hat{x}_{ui}$; $\mathcal{D}_L$ is the training corpus for the user-list entry $\hat{y}_{ui}$; $\mathcal{D}_R$ is the training corpus for the item-item-list entry $\hat{r}_{ij}$; and λ_r balances the performance between BPR and the embedding model for sentences' equivalent factorization model.

New-Item Cold-Start



The illustration of the new-item cold-start problem where cold-start items only exist in lists and are never consumed by users.

Outline

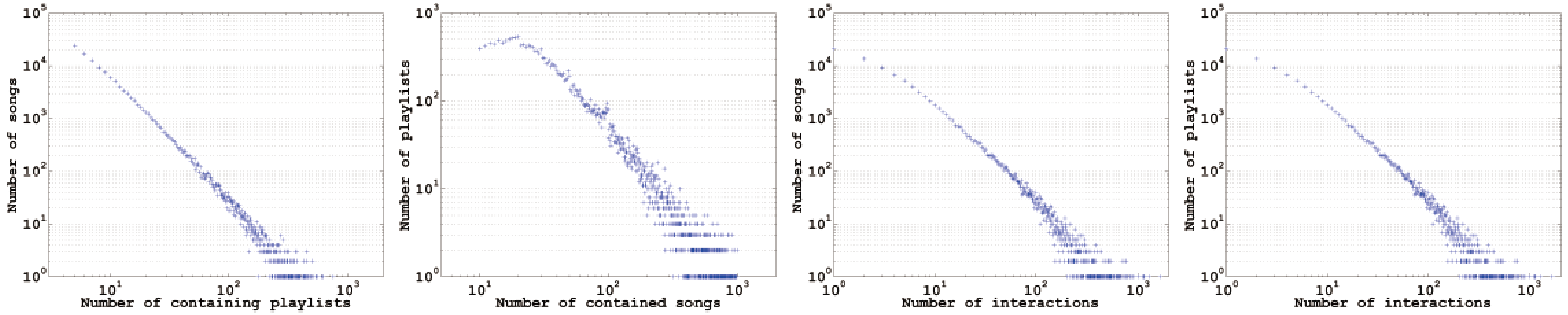
- Background
- Proposed Method
- Experiments and Results
- Conclusion

Data Statistics

Statistics of the evaluation datasets

Dataset	User-song interaction#	User-list interaction#	User#	Song#	Playlist#	User-song density	User-list density
User-Song	1, 128, 065	0	18, 528	123, 628	22, 864	0.05%	0
User-Song-Playlist	1, 128, 065	528, 128	18, 528	123, 628	22, 864	0.05%	0.12%

Dataset statistics w.r.t. song and playlist



(a) Song-playlist. (b) Playlist-song. (c) Song-interaction. (d) Playlist-interaction.

Research Questions

(RQ1) Overall performance comparison w.r.t. individual item recommendation.

(RQ2) New-item cold-start problem.

(RQ3) Performance analysis w.r.t. items.

(RQ4) Overall performance comparison w.r.t. item and list recommendation.

(RQ5) Importance of items within a list.

Baseline Methods

- BPR [Rendle et al. 2009] (benchmark method)
- BPR-map [Gantner et al. 2010] (two-step model)
- LIRE [Liu et al. 2014] (list recommendation)
- CoFactor [Liang et al. 2016] (relationship among items)

Individual Items Recommendation (RQ1)

Overall performance comparison under the EFM-Side framework

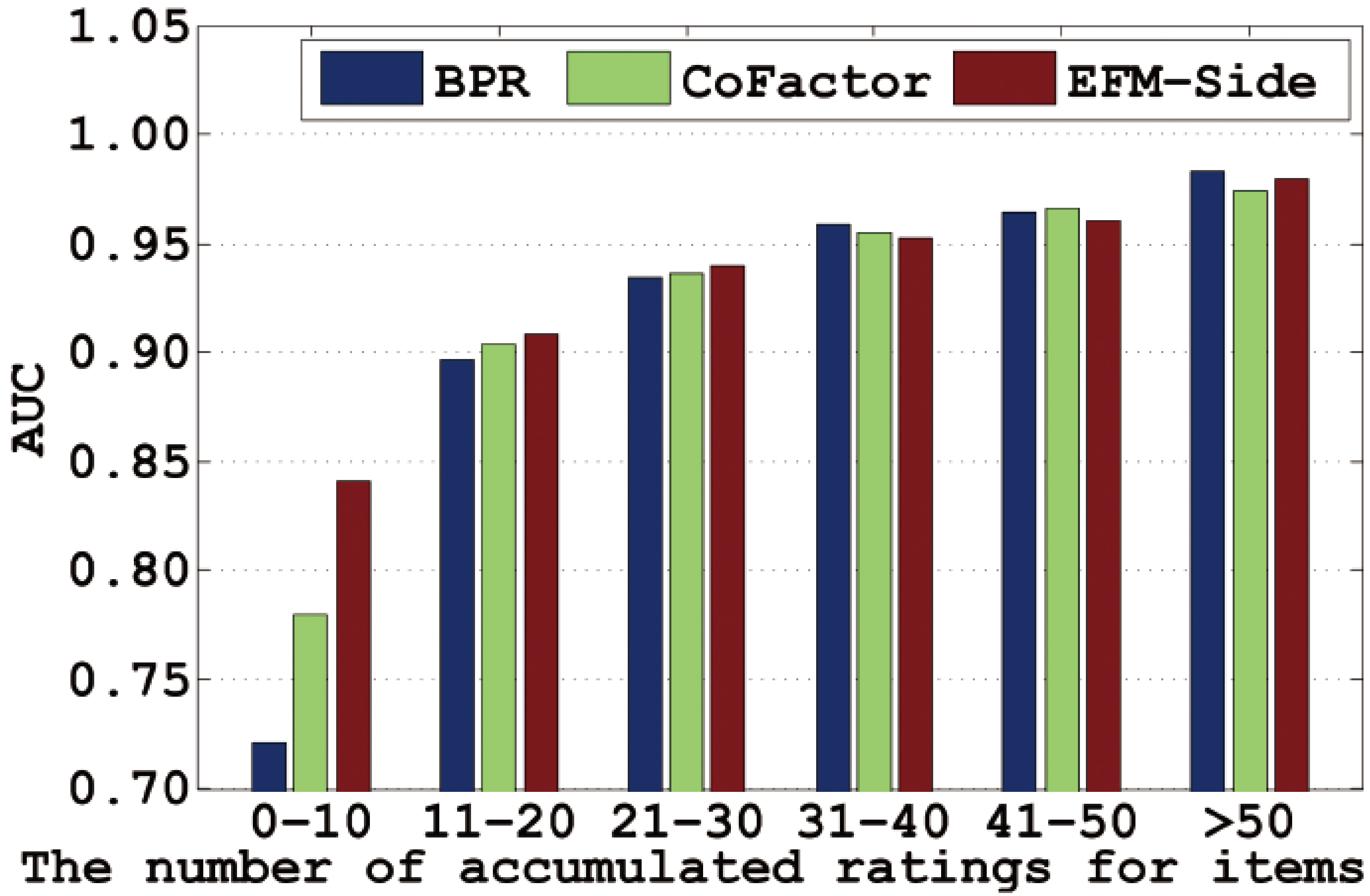
Methods	k=10		k=20	
	AUC	p-value	AUC	p-value
BPR	0.9101 ± 0.002	$3.49e$-10	0.9149 ± 0.003	$2.86e$-10
CoFactor	0.9226 ± 0.003	$5.09e$-09	0.9221 ± 0.004	$9.95e$-10
EFM-Side	$\mathbf{0.9357 \pm 0.004}$	–	$\mathbf{0.9418 \pm 0.003}$	–

New-Item Cold-Start Problem (RQ2)

Models comparison in handling the new-item cold-start problem

Methods	k=10		k=20	
	AUC	p-value	AUC	p-value
Random	0.4902 ± 0.002	$1.14e\text{-}14$	0.5019 ± 0.002	$8.35e\text{-}15$
BPR-map	0.7729 ± 0.003	$8.30e\text{-}12$	0.7777 ± 0.003	$2.25e\text{-}12$
EFM-Side	$\mathbf{0.8381 \pm 0.004}$	–	$\mathbf{0.8680 \pm 0.003}$	–

Performance Analysis w.r.t. Items (RQ3)



Micro-analysis w.r.t. items with different scale of accumulated ratings.

Jointly Recommend Items and Lists (RQ4)

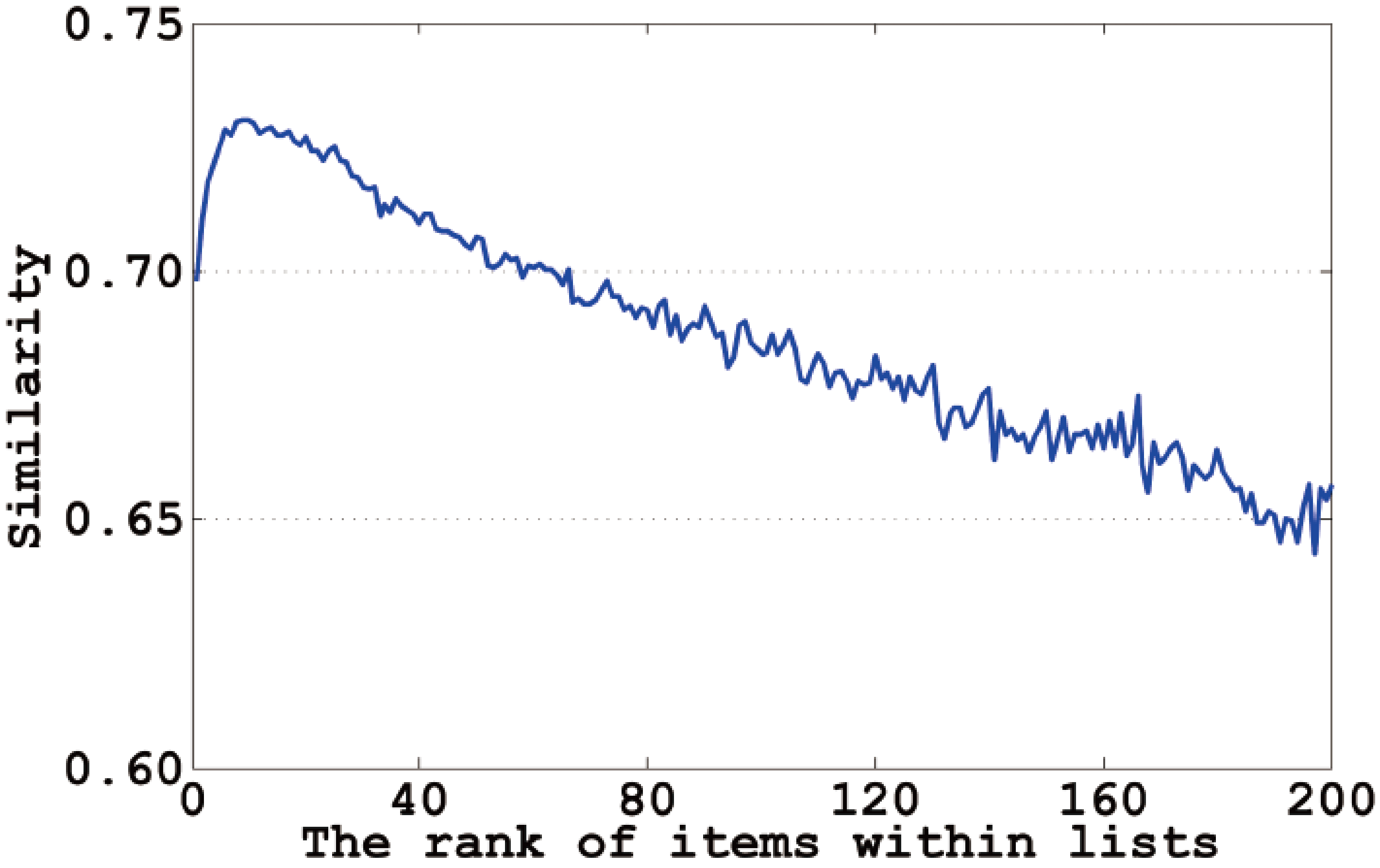
Overall performance comparison under the EFM-Joint framework w.r.t. item recommendation.

Methods	k=10		k=20	
	AUC	p-value	AUC	p-value
BPR	0.9104 ± 0.003	$4.30e\text{-}10$	0.9175 ± 0.002	$3.49e\text{-}10$
LIRE	0.9194 ± 0.003	$2.73e\text{-}09$	0.9218 ± 0.004	$7.28e\text{-}10$
CoFactor	0.9231 ± 0.004	$8.28e\text{-}09$	0.9245 ± 0.003	$1.25e\text{-}09$
EFM-Joint	$\mathbf{0.9347 \pm 0.003}$	–	$\mathbf{0.9431 \pm 0.003}$	–

Overall performance comparison under the EFM-Joint framework w.r.t. list recommendation.

Methods	k=10		k=20	
	AUC	p-value	AUC	p-value
BPR	0.8593 ± 0.002	$9.56e\text{-}10$	0.8729 ± 0.003	$2.07e\text{-}09$
LIRE	0.8675 ± 0.004	$8.00e\text{-}09$	0.8818 ± 0.003	$4.73e\text{-}08$
CoFactor	0.8605 ± 0.003	$1.22e\text{-}09$	0.8738 ± 0.004	$2.59e\text{-}09$
EFM-Joint	$\mathbf{0.8792 \pm 0.003}$	–	$\mathbf{0.8893 \pm 0.004}$	–

Importance of Items within a List (RQ5)



The similarity between the list and its contained items.

Outline

- Background
- Proposed Method
- Experiments and Results
- <u>Conclusion</u>

Challenges Solved

- The relationship among items within a list
- New-item cold-start
- User-item and user-list recommendation

Website https://listrec.wixsite.com/efms

Embedding Factorization Models for Jointly Recommending User Generated Lists and Their Contained Items

Abstract

Existing recommender systems mainly focused on recommending individual items by utilizing user-item interactions. However, little attention has been paid to recommend user generated lists (e.g., playlists and booklists). On one hand, user generated lists contain rich signal about item co-occurrence, as items within a list are usually gathered based on a specific theme. On the other hand, user's preferences over a list also indicate his/her preferences over items within the list, and vice versa. We believe that 1) if the rich relevance signal within user generated lists can be properly leveraged, a better recommendation for individual items can be provided, and 2) if user-item and user-list interactions are properly utilized, and the relationship between a list and its contained items is discovered, the performance of user-item and user-list recommendations can be mutually reinforced.

Thanksgiving

Da Cao

Assistant Professor in Hunan University

caoda@hnu.edu.cn; caoda0721@gmail.com