ACM The Web Conference 2023
User Modeling and Personalization Track

# Fine-tuning Partition-aware Item Similarities for Efficient and Scalable Recommendation

*Tianjun Wei, Jianghong Ma, Tommy W.S. Chow*

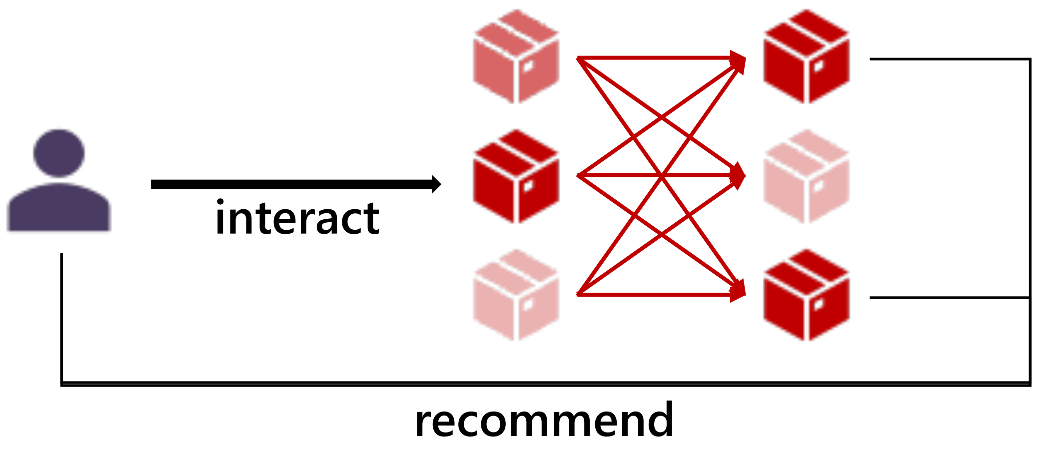
*May 4, 2023*
*Austin, TX, USA*

香港城市大學
City University of Hong Kong

# Background

## Item Similarity Models

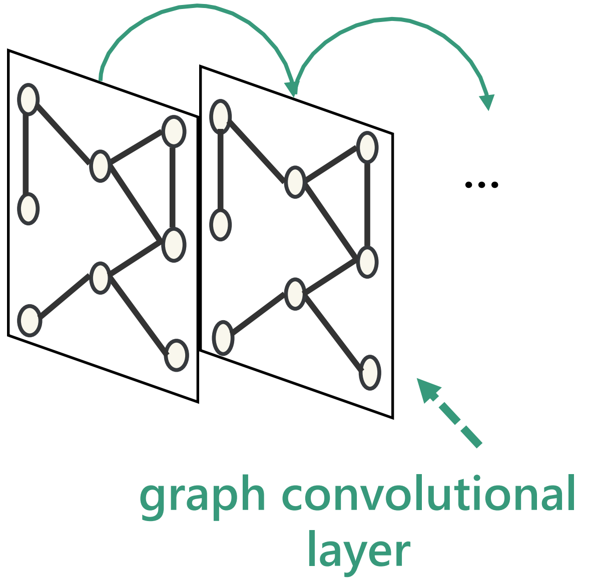
- Classical and straightforward method in CF
- Construct item-item weight (similarity) matrix
- Recommend items based on user historical interactions
- Face scalability problem with the growing number of items

# Graph in CF

## Graph Convolution Network (GCN)

- Stack multiple convolutional layers
- Capture high-order user-item relationship

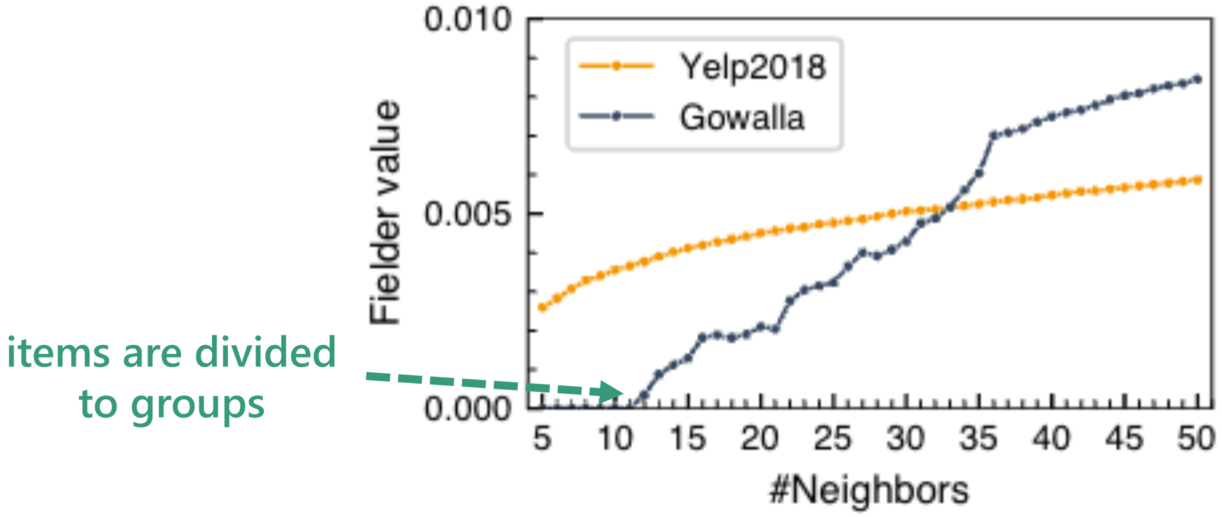



## UltraGCN [1]

- Approximate infinite convolutional layers
- Construct an item adjacency graph
- Adopt graph sampling in item adjacency graph

*[1] Kelong Mao, Jieming Zhu, Xi Xiao, Biao Lu, Zhaowei Wang, Xiuqiang He. UltraGCN: Ultra Simplification of Graph Convolutional Networks for Recommendation, in CIKM 2021.*

# Identify Group Structure




Algebraic connectivity (Fielder value) of the sampled item adjacency graph in UltraGCN

## Our investigation

- Reveal the group structure in sampled item adjacency graph
- Demonstrate that the graph sampling strategy in UltraGCN is an approximation of graph partitioning

# Inspiration & Challenges

## Inspiration

- Implement graph partitioning to item similarity models
- Community structure exists in real-world cases

cuisine in restaurant

location of hotel

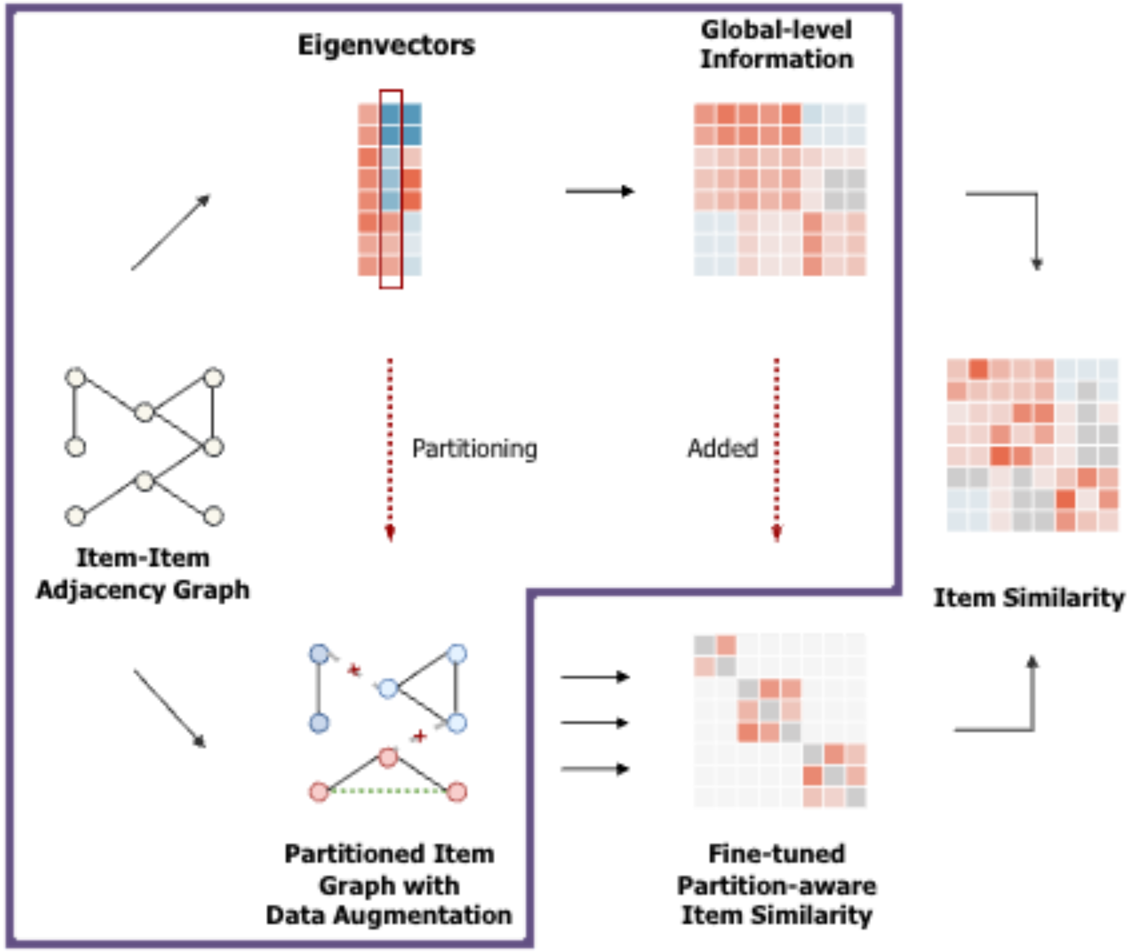
## Challenges

- Real-world data and partitioning result are not ideal
- INFORMATION LOSS

# Methodology

## Spectral Information in Graph

- Be used to perform graph partitioning
- Be revealed to be better in preserving inter-partition item relationships
- Be added to the fine-tuning of item similarity within each partition as global-level information



# Methodology (cont.)

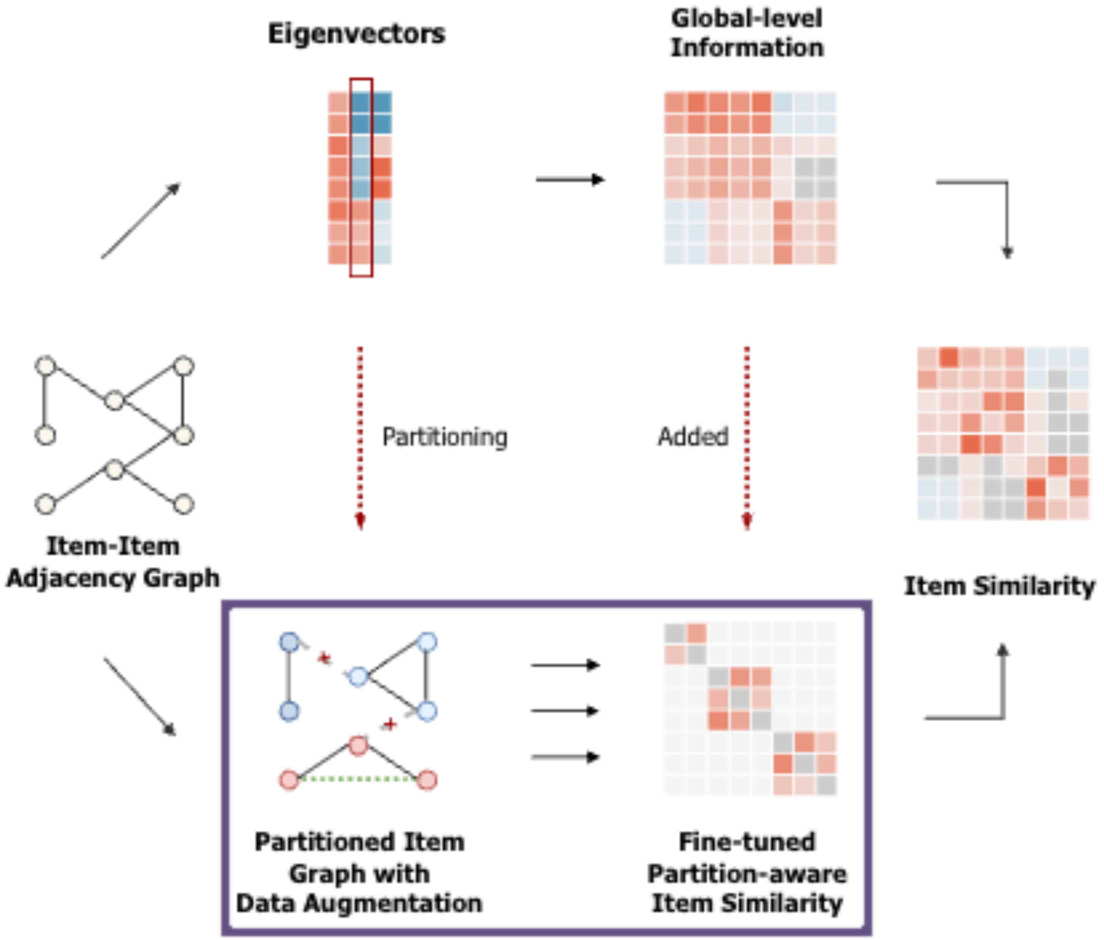
## Local Prior Knowledge of Partitions

- A data augmentation strategy
- Virtual users who have interacted with all items in a partition
- Directly add values to item adjacency matrix
- No extra computational cost

## Similarity Modeling in Each Partition

$$\underset{S}{argmin} \quad tr(\, S^T QS - 2(\, I - \lambda W^T)\, QS) \;\; \ldots$$

$$Q = R^T R + \eta$$



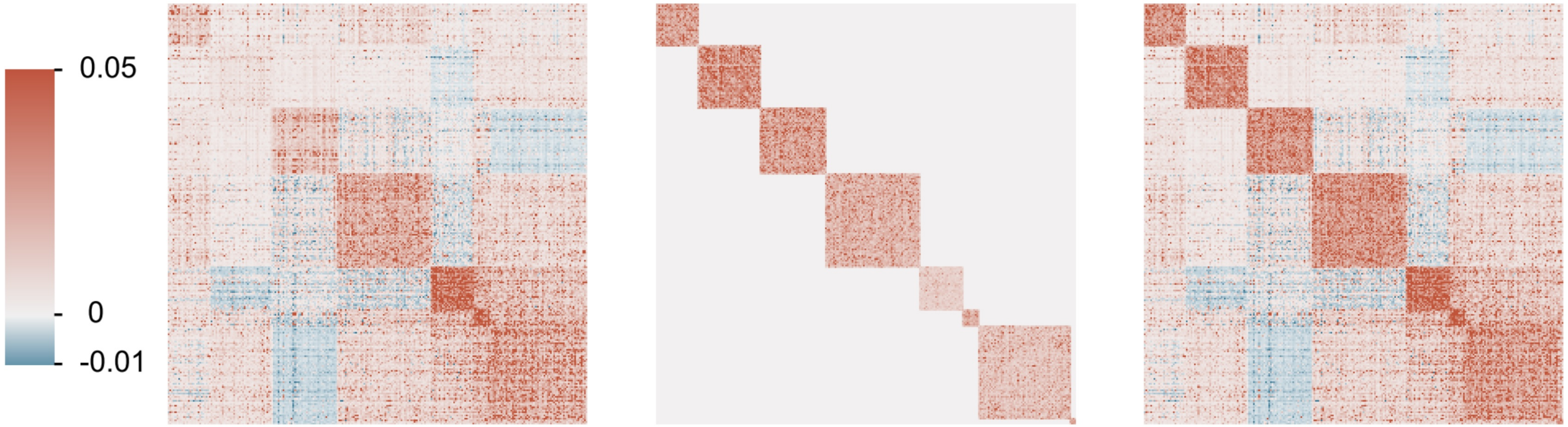
# Experiment

- Achieve great recommendation performance
- High training speed
- Low parameter storage
- Ensure both efficiency and scalability

| Dataset | Metric | GF-CF | LightGCN | SimGCL | UltraGCN | EASE | BISM | FPSR |
|---|---|---|---|---|---|---|---|---|
| *Yelp2018* | ***Recall@20*** | *0.0697* | *0.0653* | *0.0681* | *0.0683* | *0.0657* | *0.0662* | ***0.0703*** |
| | ***NDCG@20*** | *0.0571* | *0.0532* | *0.0556* | *0.0561* | *0.0552* | *0.0559* | ***0.0584*** |
| | ***#Params*** | - | *4.46M* | *4.46M* | *4.46M* | *1448M* | *191M* | *3.27M* |
| | ***Time (s)*** | *23* | *~74000* | *~2900* | *617* | *22* | *783* | *35* |
| *Douban* | ***Recall@20*** | *0.1719* | *0.1571* | *0.1699* | *0.1925* | *0.2038* | ***0.2158*** | *0.2095* |
| | ***NDCG@20*** | *0.1365* | *0.1206* | *0.1346* | *0.1556* | *0.1786* | *0.1889* | ***0.1950*** |
| | ***#Params*** | - | *5.04M* | *5.04M* | *5.04M* | *499M* | *158M* | *2.14M* |
| | ***Time (s)*** | *19* | *~22000* | *~2700* | *~5900* | *16* | *410* | *33* |

Performance comparison (partial)

# Detailed Analysis




*Global-level information* *Fine-tuned item similarity* *Weighted sum*

Visualization of item similarity matrix learned in *Yelp2018* dataset

| Dataset | *Amazon-cds* | | *Douban* | | *Gowalla* | | *Yelp2018* | |
|---|---|---|---|---|---|---|---|---|
| Metrics | *R@20* | *N@20* | *R@20* | *N@20* | *R@20* | *N@20* | *R@20* | *N@20* |
| w/o global-level information | *0.1540* | *0.0873* | *0.2046* | *0.1909* | *0.1883* | ***0.1566*** | *0.0702* | *0.0582* |
| w/o local prior knowledge | *0.1542* | *0.0887* | *0.2085* | *0.1945* | *0.1785* | *0.1486* | *0.0662* | *0.0556* |
| w/o $l_2$ regularization | *0.1539* | *0.0870* | *0.2084* | *0.1943* | *0.1832* | *0.1501* | *0.0692* | *0.0574* |
| FPSR | ***0.1576*** | ***0.0896*** | ***0.2095*** | ***0.1950*** | ***0.1884*** | ***0.1566*** | ***0.0703*** | ***0.0584*** |

Ablation analysis

# Thank you!

**The code is available at GitHub: *Joinn99/FPSR***

*Presenter:*

*Tianjun Wei*

*City University of Hong Kong*

*tjwei2-c@my.cityu.edu.hk*