SIGIR 2022
Madrid

Graph Trend Filtering Networks for Recommendation

Wenqi Fan[1†], Xiaorui Liu[2†], Wei Jin[2], Xiangyu Zhao[3*], Jiliang Tang[2], and Qing Li[1]

[1]The Hong Kong Polytechnic University, [2]Michigan State University,

[3]City University of Hong Kong

Code (Pytorch): https://github.com/wenqifan03/GTN-SIGIR2022

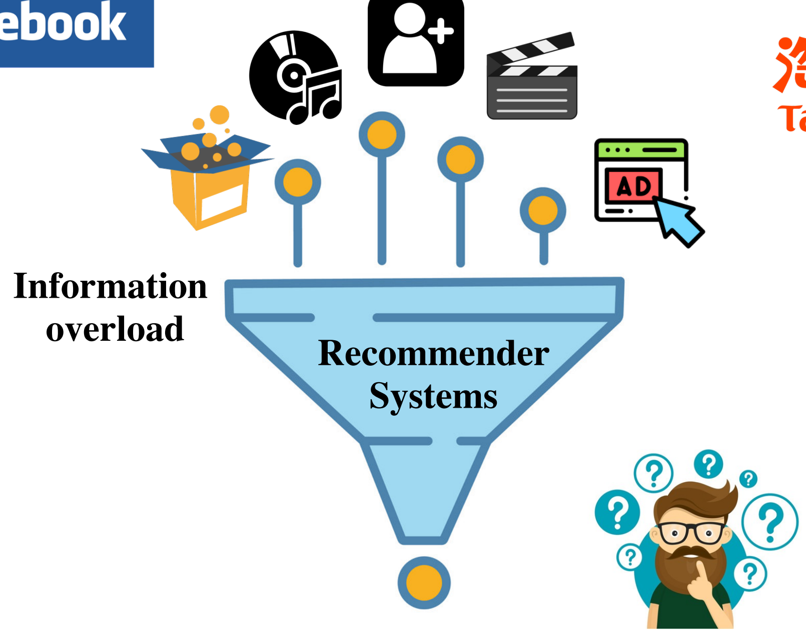
香港城市大學
City University of Hong Kong

MICHIGAN STATE UNIVERSITY

THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

Recommender Systems

Age of Information Explosion



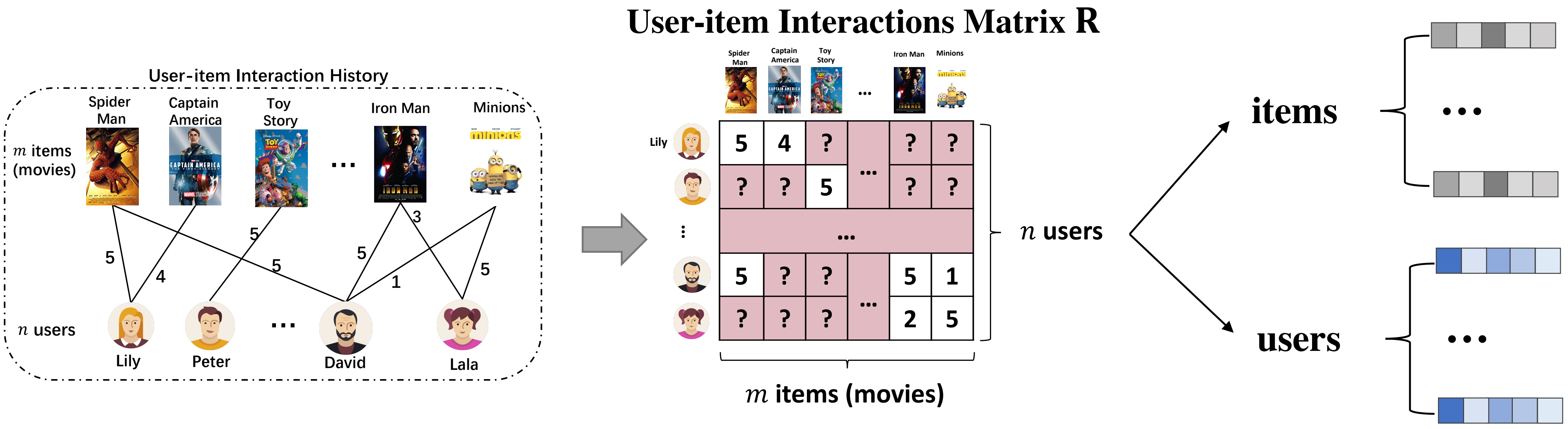
Recommend item X to user

Items can be: Products, News, Movies, Videos, Friends, etc.

Recommender Systems

- **Collaborative Filtering (CF) is the most well-known technique for recommendation.**
  - Similar users (with respect to their historical interactions) have similar preferences.
  - Modelling users’ preference on items based on their past interactions (e.g., ratings and clicks).

- **Learning representations of users and items is the key of CF.**

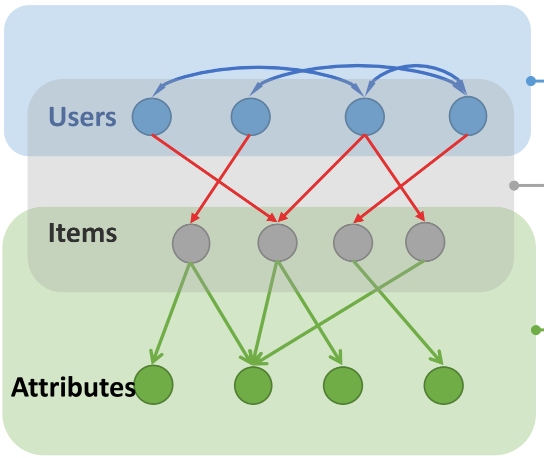


Data as Graphs

Most of the data in RS has essentially a graph structure

- E-commerce, Content Sharing, Social Networking ...

The world is more closely connected than you might think!



user-user social graph

User-user Connections
(social networks)

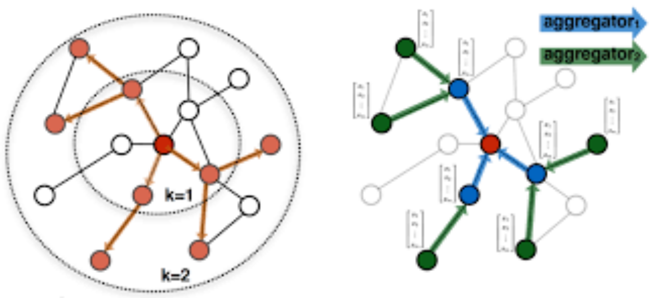
User-Item Interactions
(purchase, click, add-to-cart, etc.)

user-item graph

Item-item Connections
(substitutes/complements; external knowledge)

item graph

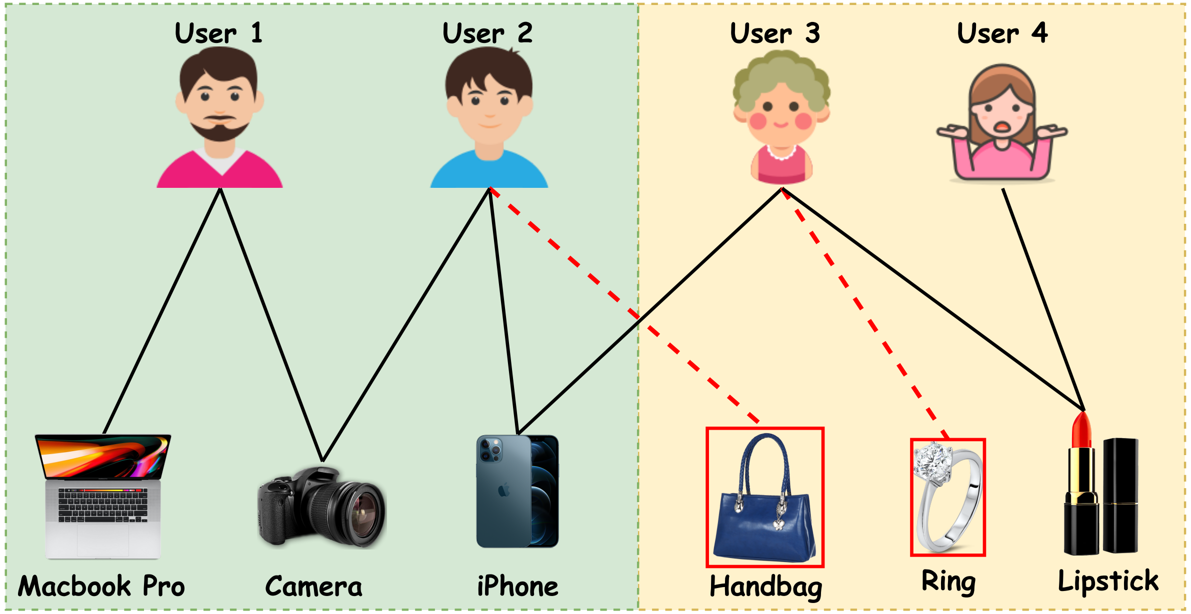
Graph Neural Networks (GNNs)



E.g., NGCF, LightGCN, GraphRec

Drawbacks of existing GNN-based recommendations

- Unreliable user-item interactions (e.g., random/bait clicks)



E.g.,
(1) User 3 was affected by the click-bait issue.
(2) User 2 bought a one-time item for his mother's birthday present;

Graph Trend Filtering Networks for Recommendations, SIGIR'2022. arXiv:2108.05552.

Drawbacks of existing GNN-based recommendations

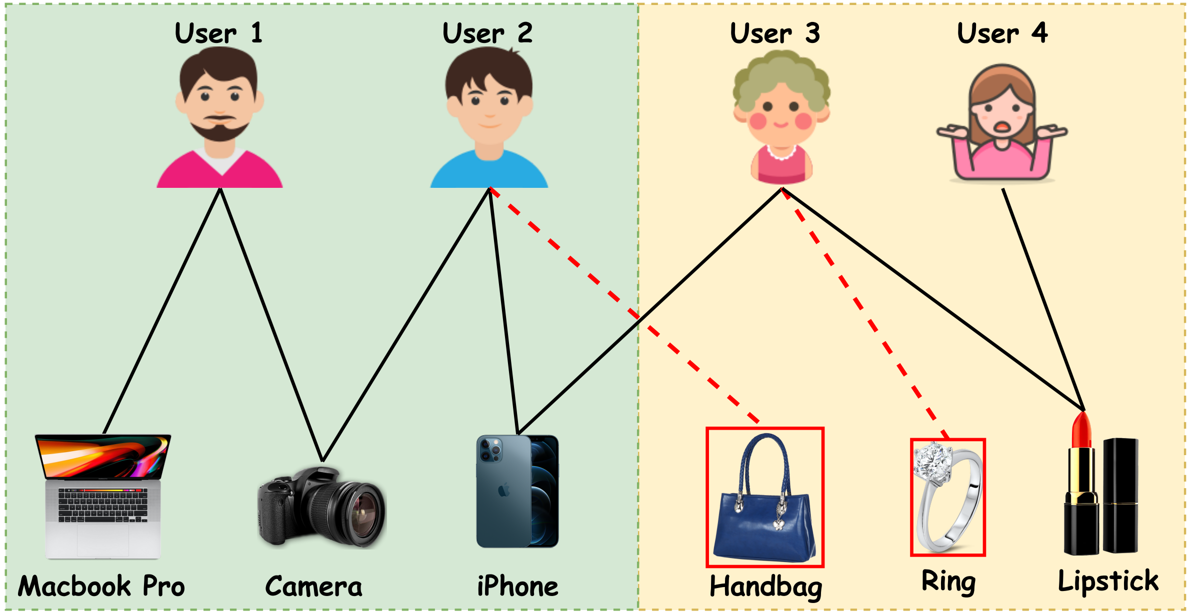
- Unreliable user-item interactions (e.g., random/bait clicks)
- *Embedding Propagation Rule* in GNN-based methods

$$\mathbf{e}_u^{k+1} = \frac{1}{\sqrt{|\mathcal{N}(u)|}} \sum_{i \in \mathcal{N}(u)} \frac{1}{\sqrt{|\mathcal{N}(i)|}} \mathbf{e}_i^k$$

$$\mathbf{e}_i^{k+1} = \frac{1}{\sqrt{|\mathcal{N}(i)|}} \sum_{u \in \mathcal{N}(i)} \frac{1}{\sqrt{|\mathcal{N}(u)|}} \mathbf{e}_u^k$$

➢ **Overlook unreliable interactions and uniformly treat all the interactions**

➢ **non-adaptive propagation/non-robustness**

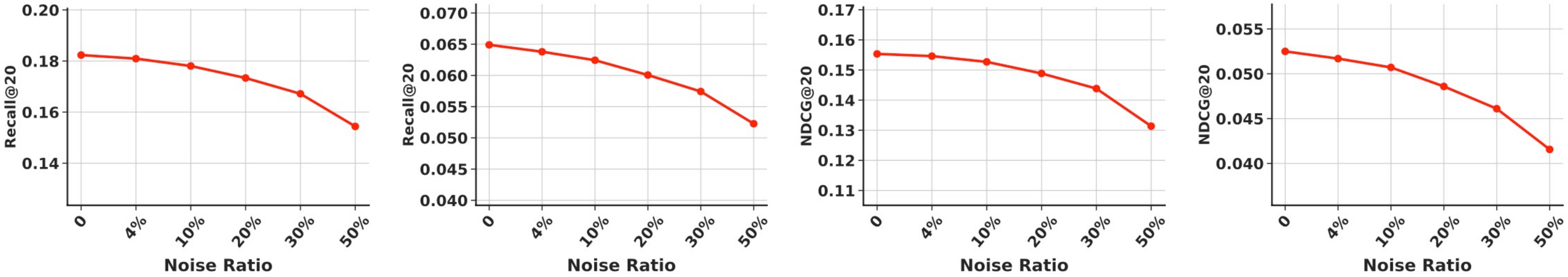


E.g.,
(1) User 3 was affected by the click-bait issue.
(2) User 2 bought a one-time item for his mother's birthday present;

Graph Trend Filtering Networks for Recommendations, SIGIR'2022. arXiv:2108.05552.
LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. SIGIR 2020.

Preliminary study

- Performance of LightGCN under different perturbation ratios.



(a) Gowalla-Recall@20 (b) Yelp2018-Recall@20 (c) Gowalla-NDCG@20 (d) Yelp2018-NDCG@20

Figure 2: Performance of LightGCN under different perturbation rates.

Graph Trend Filtering Networks for Recommendations, SIGIR'2022. arXiv:2108.05552.
LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. SIGIR 2020.

Preliminary study

▪ Performance of LightGCN under different perturbation ratios.

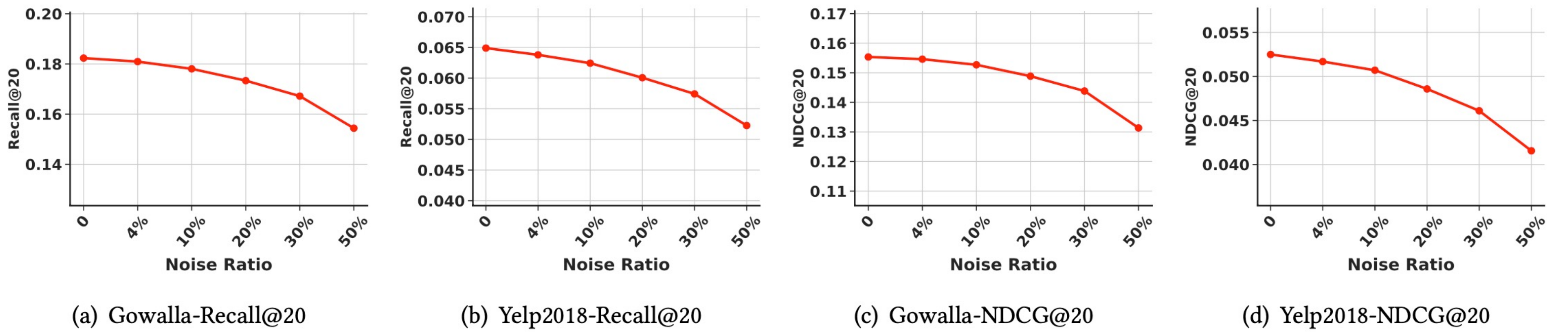


Figure 2: Performance of LightGCN under different perturbation rates.

Our Solutions:

- ➢ **To adaptively propagate the embedding in recommender systems**
- ➢ **To design a new collaborative filtering method that models the adaptive smoothness over the user-item interactions**
- ➢ **Motivated by the concepts of trend filtering and graph trend filtering**

Graph Trend Filtering Networks for Recommendations, SIGIR'2022. arXiv:2108.05552.
LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. SIGIR 2020.

Trend Filtering

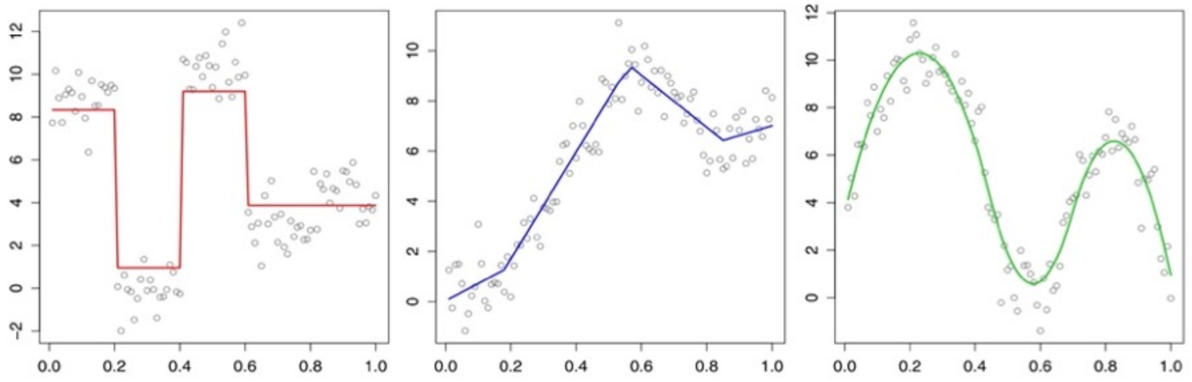
Nonparametric regression (univariate)

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \frac{1}{2}\|y-\beta\|_2^2 + \frac{n^k}{k!} \cdot \lambda \| D^{(k+1)} \beta \|_1$$

$$D^{(1)} = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & & & & & \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{bmatrix} \in \mathbb{R}^{(n-1)\times n}$$

$$D^{(k+1)} = D^{(1)} \cdot D^{(k)}$$

Adapt to the local level of smoothness



K = 0 K = 1 K = 2

*L*1 Trend filtering, S.-J. Kim et al, SIAM Review, 2009
Adaptive piecewise polynomial estimation via trend filtering, Ryan Tibshirani, Annals of Statistics, 2014

Graph Trend Filtering (GTF)

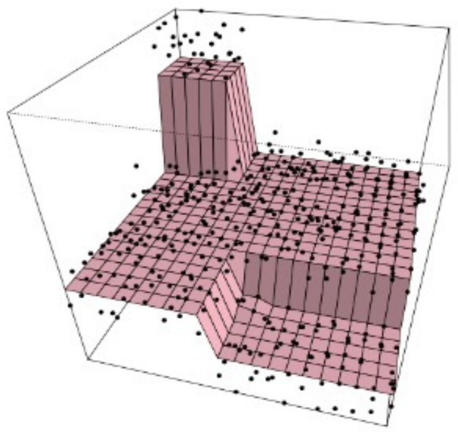
$$\underset{\mathbf{f}\in\mathbb{R}^n}{\arg\min} = \frac{1}{2}\|\mathbf{f}-\mathbf{x}\|_2^2 + \lambda\|\Delta^{(k+1)}\mathbf{f}\|_1$$

Incident matrix

$$\Delta_\ell = (0, \ldots, \underbrace{-1}_{i}, \ldots, \underbrace{1}_{j}, \ldots, 0)$$

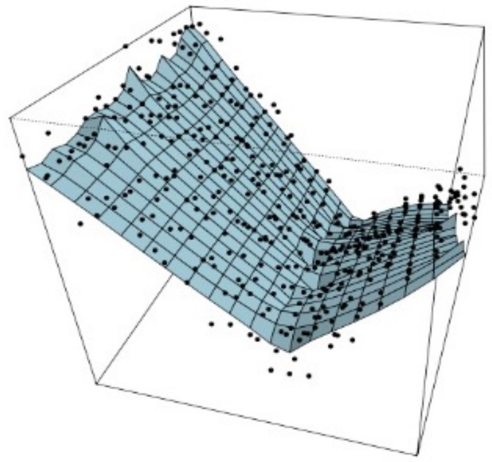
Local Smoothness adaptivity: piecewise behavior

GTF with $k = 0$



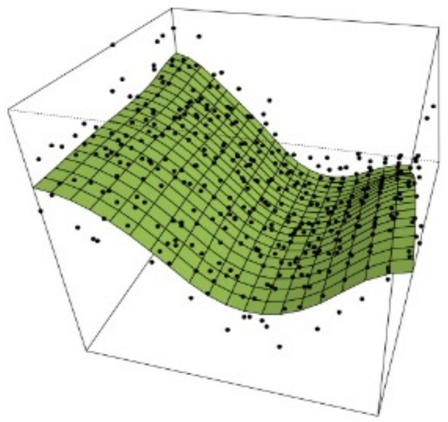
Penalty: $\sum_{(i,j)\in E} |\beta_i - \beta_j|$

GTF with $k = 1$



$$\sum_{i=1}^{n} n_i \left|\beta_i - \frac{1}{n_i}\sum_{j:(i,j)\in E}\beta_j\right|$$

GTF with $k = 2$



$$\sum_{(i,j)\in E}\left| n_i\left(\beta_i - \frac{1}{n_i}\sum_{\ell:(i,\ell)\in E}\beta_\ell\right) - n_j\left(\beta_j - \frac{1}{n_j}\sum_{\ell:(j,\ell)\in E}\beta_\ell\right)\right|$$

Trend filtering on graphs, Yu-Xiang Wang et al, JMLR 2016
Elastic Graph Neural Networks, ICML, 2021.

GNN-based Collaborative Filtering

- **Embedding Propagation Rule**

$$\mathbf{e}_u^{k+1} = \frac{1}{\sqrt{|\mathcal{N}(u)|}} \sum_{i \in \mathcal{N}(u)} \frac{1}{\sqrt{|\mathcal{N}(i)|}} \mathbf{e}_i^k$$

$$\mathbf{e}_i^{k+1} = \frac{1}{\sqrt{|\mathcal{N}(i)|}} \sum_{u \in \mathcal{N}(i)} \frac{1}{\sqrt{|\mathcal{N}(u)|}} \mathbf{e}_u^k$$

Matrix form:

$$\mathbf{E}^{K+1} = \tilde{\mathbf{A}}\mathbf{E}^k$$

$$\mathbf{E} = [\underbrace{\mathbf{e}_1, \cdots, \mathbf{e}_n}_{\text{users embeddings}}, \underbrace{\mathbf{e}_{n+1}, \cdots, \mathbf{e}_{n+m}}_{\text{item embeddings}}]^\top \in \mathbb{R}^{(n+m)\times d}$$

- **Graph Laplacian Smoothing Problem:**

$$\arg\min_{\mathbf{E} \in \mathbb{R}^{(n+m)\times d}} \operatorname{tr}(\mathbf{E}^\top (\mathbf{I} - \tilde{\mathbf{A}})\mathbf{E}) = \sum_{(i,j)\in\mathcal{E}} \left\| \frac{\mathbf{e}_i}{\sqrt{d_i+1}} - \frac{\mathbf{e}_j}{\sqrt{d_j+1}} \right\|_2^2$$ **(Edge-wise form)**

Idea: the embedding propagation essentially enforces the user embedding e*i* and item embedding e*j* to be close (i.e., embedding smoothness), if there exist interactions between them.

Graph Trend Filtering Networks for Recommendations, SIGIR'2022. arXiv:2108.05552.
Elastic Graph Neural Networks, ICML, 2021.
A unified view on graph neural networks as graph signal denoising, arXiv:2010.01777, 2020.

Graph Trend Filtering Networks for Recommendations

- Design Motivation from Graph Trend Filtering (GTF)
- Embedding smoothness objective:

$$\arg\min_{\mathbf{E}\in\mathbb{R}^{(n+m)\times d}} \underbrace{\frac{1}{2}\|\mathbf{E}-\mathbf{E}_{\text{in}}\|_F^2}_{\text{Preserve the proximity}} + \lambda\underbrace{\|\tilde{\Delta}\mathbf{E}\|_1}_{\text{Impose embedding smoothness}}$$

$$\|\tilde{\Delta}\mathbf{E}\|_1 = \sum_{(i,j)\in\mathcal{E}} \left\| \frac{\mathbf{e}_i}{\sqrt{d_i+1}} - \frac{\mathbf{e}_j}{\sqrt{d_j+1}} \right\|_1.$$

Elastic Graph Neural Networks, ICML, 2021.
Graph Trend Filtering Networks for Recommendations, SIGIR'2022. arXiv:2108.05552.

Graph Trend Filtering Networks for Recommendations

- Design Motivation from Graph Trend Filtering (GTF)
- Embedding smoothness objective:

GNN-based Collaborative Filtering

$$\sum_{(i,j)\in\mathcal{E}} \left\| \frac{\mathbf{e}_i}{\sqrt{d_i+1}} - \frac{\mathbf{e}_j}{\sqrt{d_j+1}} \right\|_2^2$$

Our proposed GTN

$$\|\tilde{\Delta}\mathbf{E}\|_1 = \sum_{(i,j)\in\mathcal{E}} \left\| \frac{\mathbf{e}_i}{\sqrt{d_i+1}} - \frac{\mathbf{e}_j}{\sqrt{d_j+1}} \right\|_1.$$

$$\|\tilde{\Delta}\mathbf{E}\|_1 = \sum_{(i,j)\in\mathcal{E}} \boxed{\mathbf{W}_{ij}} \cdot \left\| \frac{\mathbf{e}_i}{\sqrt{d_i+1}} - \frac{\mathbf{e}_j}{\sqrt{d_j+1}} \right\|_2^2$$

$$\mathbf{W}_{ij} = \frac{\left\| \frac{\mathbf{e}_i}{\sqrt{d_i+1}} - \frac{\mathbf{e}_j}{\sqrt{d_j+1}} \right\|_1}{\left\| \frac{\mathbf{e}_i}{\sqrt{d_i+1}} - \frac{\mathbf{e}_j}{\sqrt{d_j+1}} \right\|_2^2},$$

Elastic Graph Neural Networks, ICML, 2021.
Graph Trend Filtering Networks for Recommendations, SIGIR'2022. arXiv:2108.05552.

GTN

- Embedding initialization

$$\mathbf{E}_{\text{in}} = [\underbrace{\mathbf{e}_1^0, \cdots, \mathbf{e}_n^0}_{\text{users embeddings}}, \underbrace{\mathbf{e}_{n+1}^0, \cdots, \mathbf{e}_{n+m}^0}_{\text{item embeddings}}]^\top \in \mathbb{R}^{(n+m)\times d}.$$

- Embedding Filtering (graph trend collaborative filtering with K iteration)

$$\mathbf{E}^K = [\underbrace{\mathbf{e}_1^K, \cdots, \mathbf{e}_n^K}_{\text{users embeddings}}, \underbrace{\mathbf{e}_{n+1}^K, \cdots, \mathbf{e}_{n+m}^K}_{\text{item embeddings}}]^\top \in \mathbb{R}^{(n+m)\times d}$$

LightGCN: Layer Combination

$$\mathbf{e}_u = \sum_{k=0}^{K} \alpha_k \mathbf{e}_u^{(k)}; \quad \mathbf{e}_i = \sum_{k=0}^{K} \alpha_k \mathbf{e}_i^{(k)}$$

- Model Prediction

$$\hat{y}_{ui} = {\mathbf{e}_u^K}^T \mathbf{e}_i^K$$

- Model Training

$$\mathcal{L}_{\text{BPR}} = \sum_{(u,i,j)\in O} -\ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \alpha \|\mathbf{E}_{\text{in}}\|_F^2$$

Experiment Settings

- Datasets:

  Gowalla, Yelp2018, Amazon-Book, LastFM
- Evaluation Metrics:

  Recall@20, NDCG@20
- Dataset partition:

  randomly select 80% data for training set, and 20% data for testing set.

Table 1: Basic statistics of benchmark datasets.

Datasets	User-Item Interaction		
	#Users	#Items	#Interactions
Gowalla	29,858	40,981	1,027,370
Yelp2018	31,668	38,048	1,561,406
Amazon-Book	52,643	91,599	2,984,108
LastFM	23,566	48,123	3,034,796

Compared Methods

- **GTN**: Graph Trend Filtering Networks for Recommendations
- LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation
- NGCF: Neural Graph Collaborative Filtering
- Mult-VAE: Variational Autoencoders for Collaborative Filtering
- DGCF: Disentangled Graph Collaborative Filtering
- GC-MC: Graph Convolutional Matrix Completion
- NeuCF: Neural Collaborative Filtering
- MF: Bayesian Personalized Ranking from Implicit Feedback

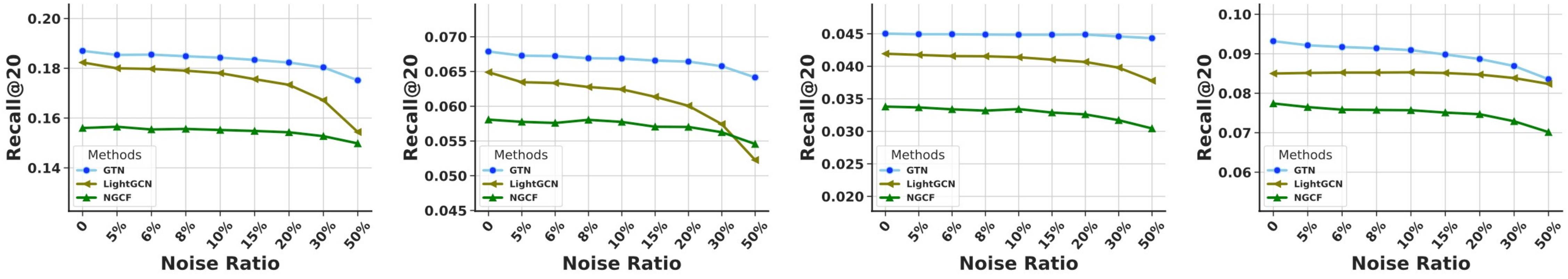
Our GTN Code (Pytorch): https://github.com/wenqifan03/GTN- SIGIR2022

Experiment Results

Table 2: The comparison of overall performance.

Datasets		Gowalla		Yelp2018		Amazon-Book		LastFM	
Metrics		Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
Method	MF	0.1299	0.111	0.0436	0.0353	0.0252	0.0198	0.0725	0.0614
	NeuCF	0.1406	0.1211	0.045	0.0364	0.0259	0.0202	0.0723	0.0637
	GC-MC	0.1395	0.1204	0.0462	0.0379	0.0288	0.0224	0.0804	0.0736
	NGCF	0.156	0.1324	0.0581	0.0475	0.0338	0.0266	0.0774	0.0693
	Mult-VAE	0.1641	0.1335	0.0584	0.045	0.0407	0.0315	0.078	0.07
	DGCF	0.1794	0.1521	0.064	0.0522	0.0399	0.0308	0.0794	0.0748
	LightGCN	0.1823	0.1553	0.0649	0.0525	0.042	0.0327	0.085	0.076
	GTN (Ours)	0.187	0.1588	0.0679	0.0554	0.045	0.0346	0.0932	0.0857
Relative Improvement (%)		2.59	2.26	4.62	5.59	7.15	5.95	9.61	12.77

- Performance Comparison under different perturbation ratios.

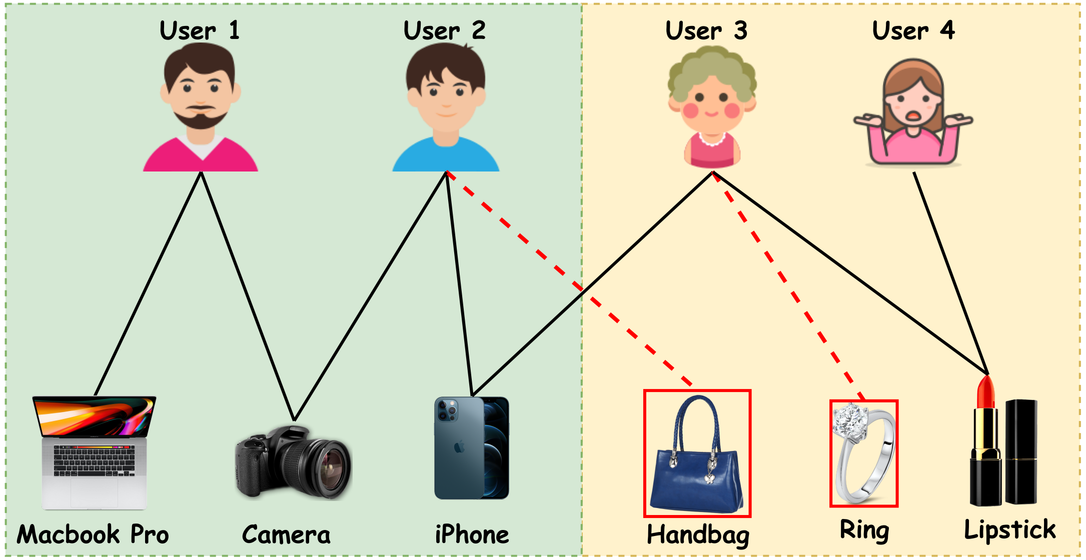


(a) Gowalla - Recall@20 (b) Yelp2018 - Recall@20 (c) Amazon-book - Recall@20 (d) LastFM - Recall@20

Conclusion

- Drawbacks of most existing GNN-based recommendation methods: non-adaptive propagation and non-robustness.
- Propose a graph trend collaborative filtering algorithm (GTN) to capture the adaptive reliability of the interactions.

Thank You!



Wenqi Fan, Xiaorui Liu, Wei Jin, Xiangyu Zhao, Jiliang Tang, and Qing Li

Graph Trend Filtering Networks for Recommendation, SIGIR 2022, Arxiv: 2108.05552

Code (Pytorch): https://github.com/wenqifan03/GTN-SIGIR2022