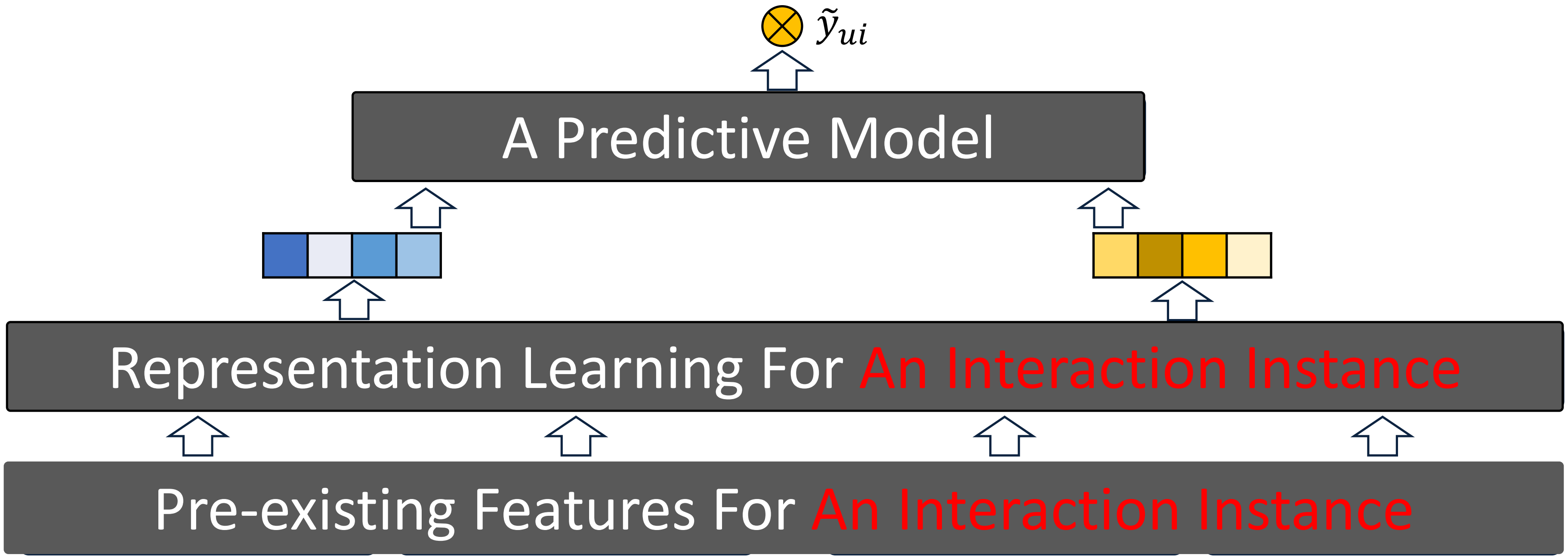
Neural Graph Collaborative Filtering

Xiang Wang, Xiangnan He, Meng Wang,
Fuli Feng, Tat-Seng Chua

22 July 2019

Modern Recommendation Paradigm

SIGIR 2019



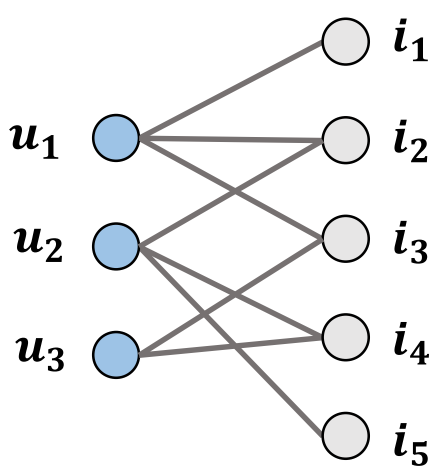
Information Isolated Island Effect:

- Model each instance individually
- While overlooking **relations** among instances
- Might result in **suboptimal performance**:
  - Making an i **Limited Representation Ability** nt only on its own features
  - Making inte **Suboptimal Model Capacity**

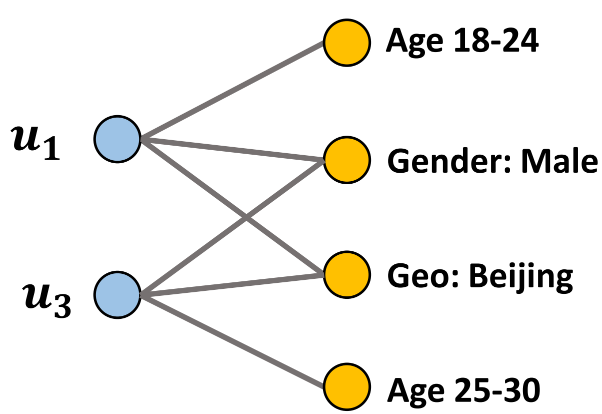
One Solution: Reorganizing Data into Graphs!

The data is more **closely connected** than we might think!

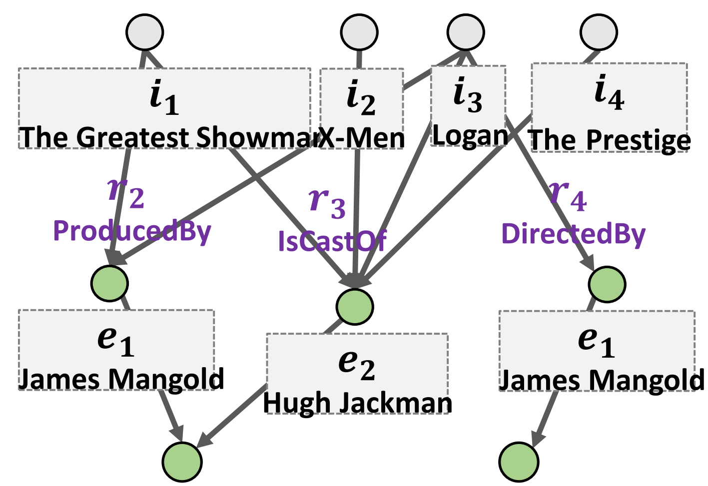
User-Item Interactions



User/Item Profiles



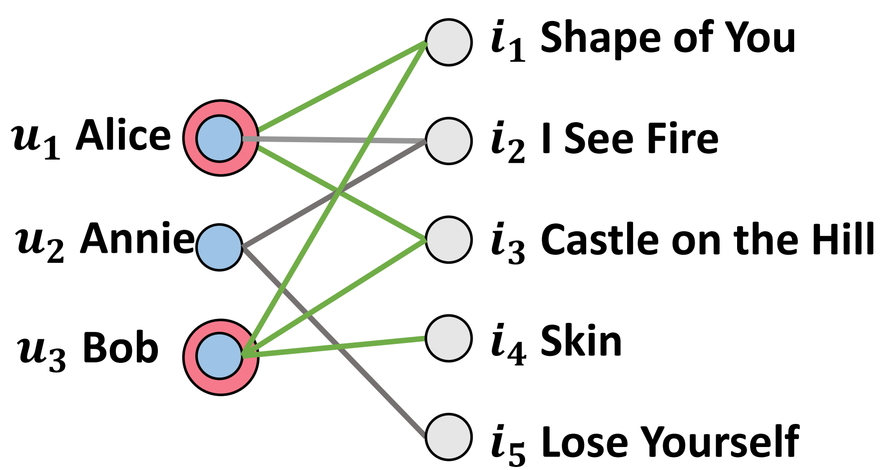
Knowledge Graph



Limited Representation Ability	⇨	Information Propagation along with the connections
Suboptimal Model Capacity	⇨	High-order connectivity complementary to user-item interactions

Collaborative Filtering (CF)

- **Collaborative Filtering** (CF) is the most well-known technique for recommendation.
  - Homophily assumption: a user preference can be predicted from his/her similar users.
- **Collaborative Signals -> Behavioral Similarity of users**
  - if u_1 and u_3 have interacted with the same items $\{i_1, i_3\}$, u_1 is likely to have similar preferences on other items $\{i_4\}$.

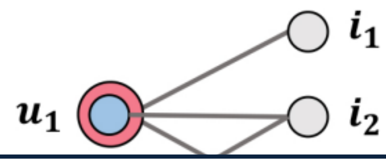


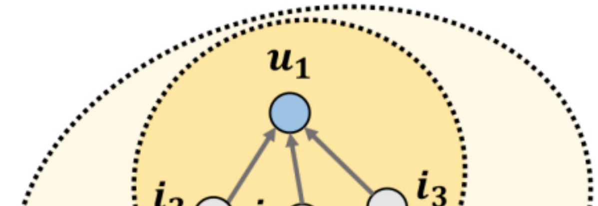
Revisiting CF via High-order Connectivity

SIGIR 2019

High-order Connectivity from User-item Bipartite Graphs

Why u_1 may like i_4?





Existing CF methods (e.g., MF, FISM, AutoRec) don't model high-order connectivity explicitly.

- **Embedding function only considers descriptive features (e.g., ID, attributes)**
- **User-item interactions are not considered**

Our contribution: CF modeling with high-order connectivity via GNN.

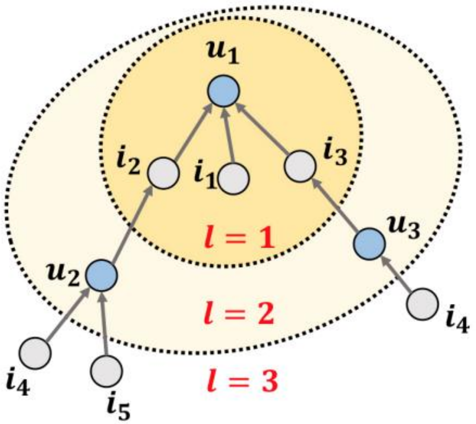
- Definition: the paths that reach u_1 from any node with the path length l larger than 1.
- A natural way to encode collaborative signals

Embedding Propagation, inspired by GNNs

- Propagate embeddings recursively on the graph
- Construct information flows in the embedding space

First-order Propagation

- **Message Construction:** generate message from one neighbor



message passed from i to u

$$\mathbf{m}_{u \leftarrow i} = \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}} \left(\mathbf{W}_1 \mathbf{e}_i + \mathbf{W}_2 (\mathbf{e}_i \odot \mathbf{e}_u) \right)$$

discount factor

- message dependent on the affinity, distinct from GCN, GraphSage, etc.
- Pass more information to similar nodes

- **Message Aggregation:** update ego node's representation by aggregating message from all neighbors

$$\mathbf{e}_u^{(1)} = \text{LeakyReLU}\left(\mathbf{m}_{u \leftarrow u} + \sum_{i \in \mathcal{N}_u} \mathbf{m}_{u \leftarrow i} \right)$$

self-connections

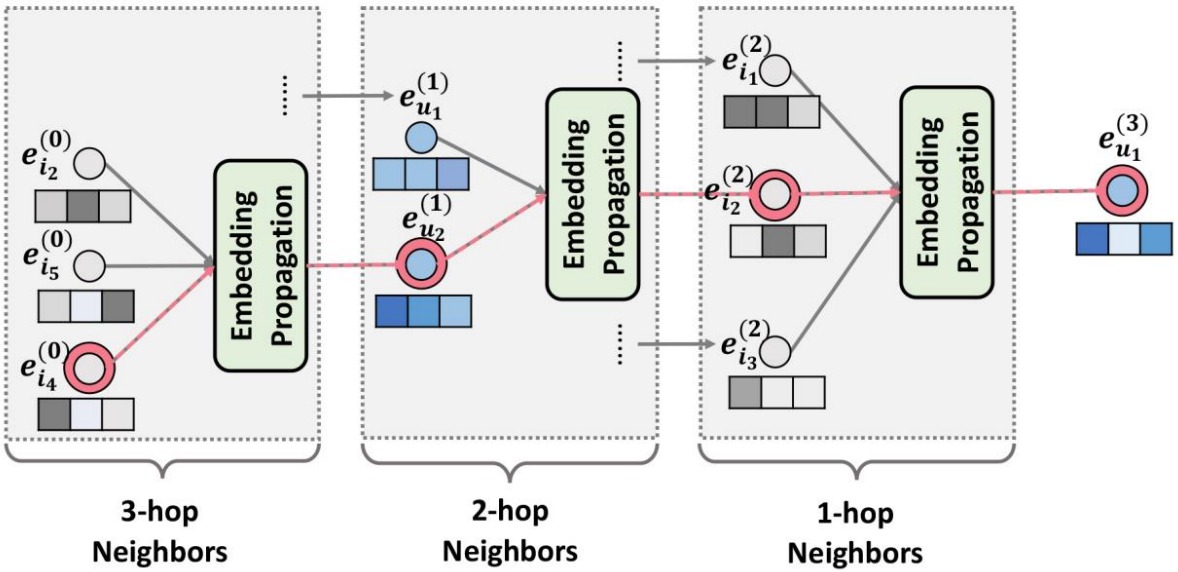
all neighbors of u

High-order Propagation

- We stack more embedding propagation layers to explore the high-order connectivity information.

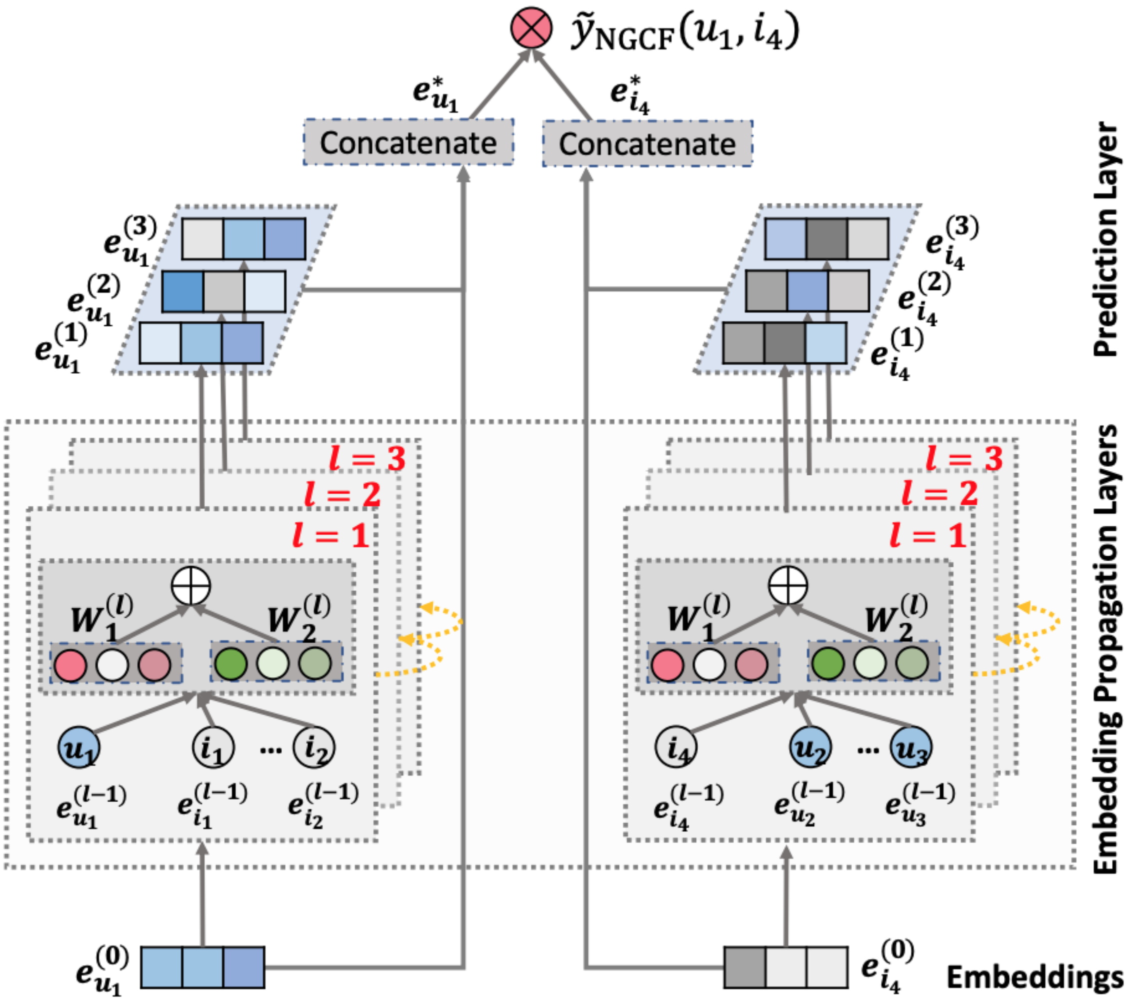
$$\mathbf{e}_u^{(l)} = \text{LeakyReLU}\Big(\mathbf{m}_{u\leftarrow u}^{(l)} + \sum_{i\in\mathcal{N}_u} \mathbf{m}_{u\leftarrow i}^{(l)}\Big),$$

representation of u at the l-th layer



- The collaborative signal like u1 ← i2 ← u2 ← i4 can be captured in the embedding propagation process.
- **Collaborative signal can be injected into the representation learning process.**

Overall Framework



$$\mathbf{e}^*_u = \boxed{\mathbf{e}^{(0)}_u \| \cdots \| \mathbf{e}^{(L)}_u}$$

$$\mathbf{e}^*_i = \mathbf{e}^{(0)}_i \| \cdots \| \mathbf{e}^{(L)}_i$$

$$\hat{y}_{NGCF}(u, i) = \mathbf{e}^{*\top}_u \mathbf{e}^*_i$$

The representations at different layers

- emphasize the messages passed over different connections
- have different contributions in reflecting user preference

Experiments

Datasets

- Gowalla, Amazon-Book, Yelp2018

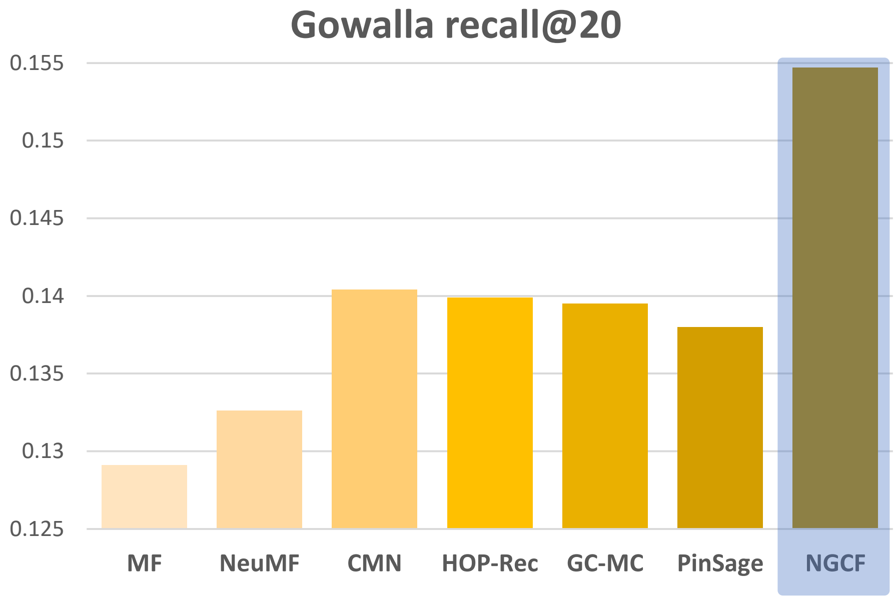
Evaluation Metrics

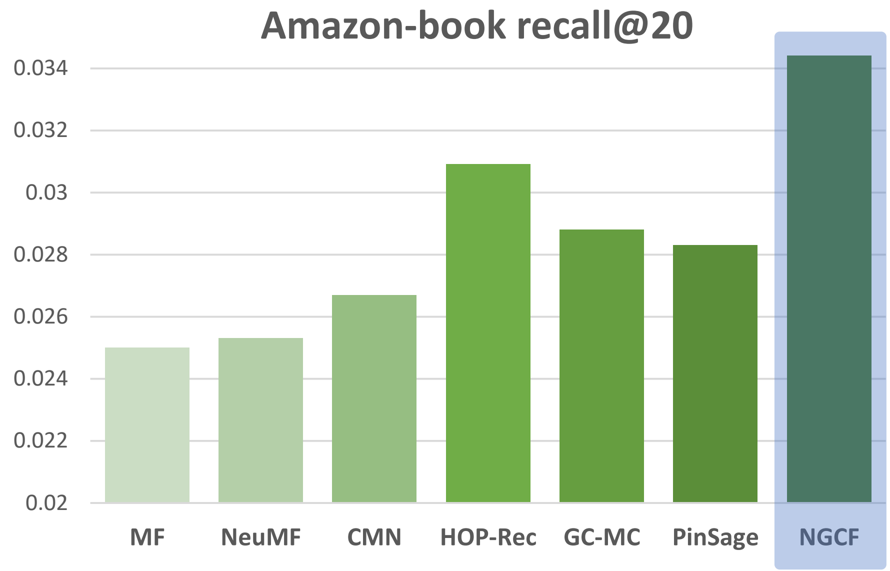
- recall@K, ndcg@K

Baselines

	Data for Embedding Function	Connectivity	Aggregation Type in GNNs	Jump Knowledge
MF	ID	-	-	-
NeuFM	ID	-	-	-
CMN	Personal History	First-order	-	-
HOP-Rec	Multi-hop Neighbors	High-order	-	-
PinSage	Collaborative Signals	Second-order	Concatenation	-
GC-MC	Collaborative Signals	First-order	Sum	-
NGCF	Collaborative Signals	**High-order**	Sum + **Element-wise Product**	**Jump Knowledge**

Overall Performance Comparison (1)

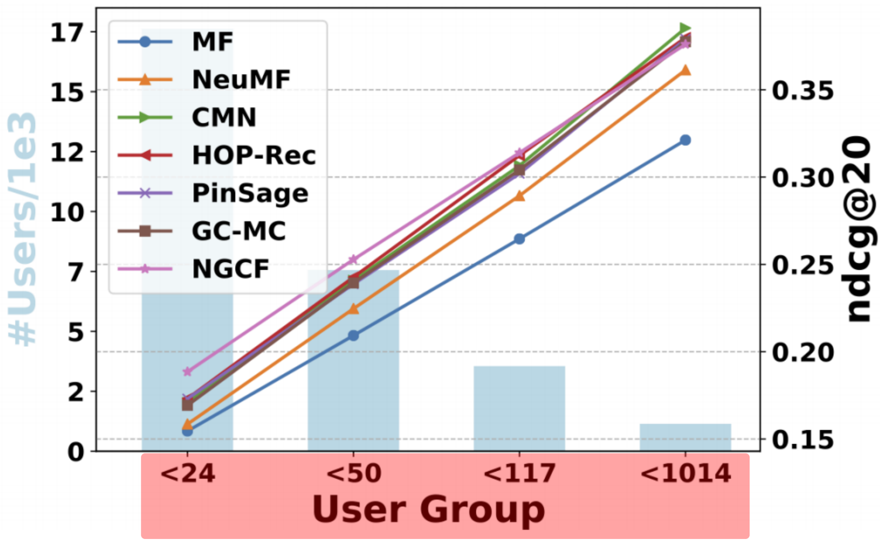


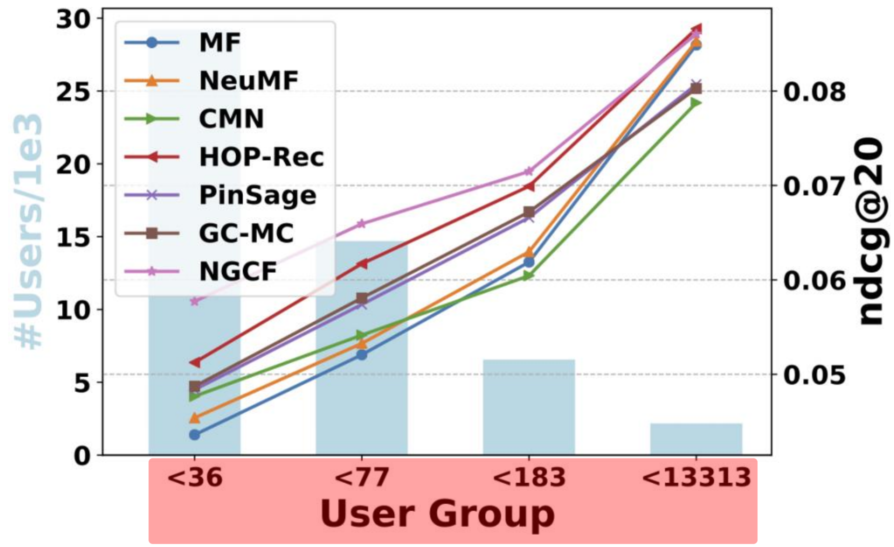


- NGCF consistently yields the best performance on all the datasets.
- **This verifies the importance of capturing collaborative signal in embedding function.**

Overall Performance Comparison (2)

SIGIR 2019



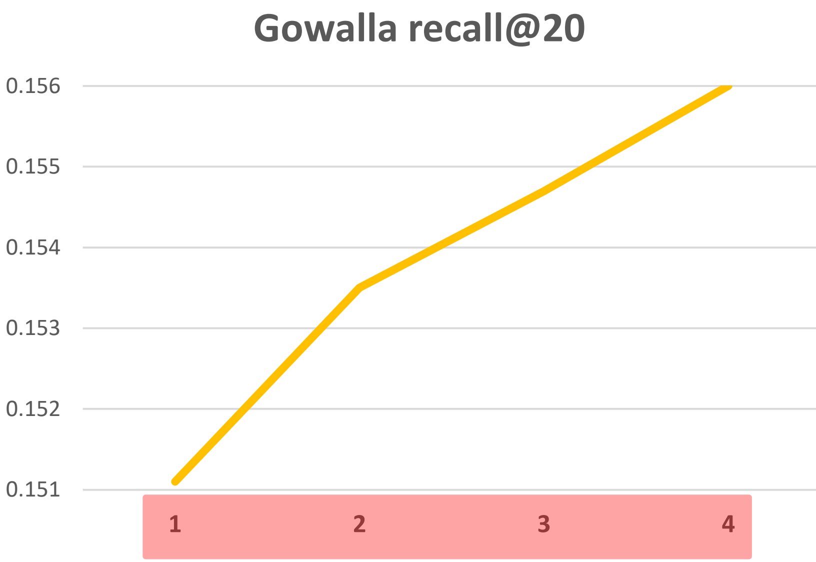


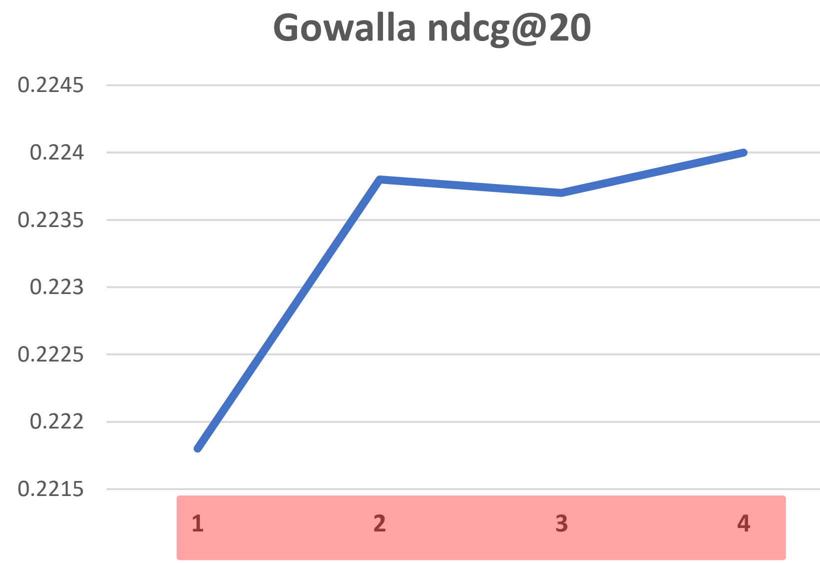
user groups with different group sparsity levels

- NGCF and HOP-Rec consistently outperform all other baselines on most user groups.
- Exploiting high-order connectivity facilitates the representation learning for inactive users.

- **It might be promising to solve the sparsity issue in recommender systems**

Effect of High-order Connectivity (1)

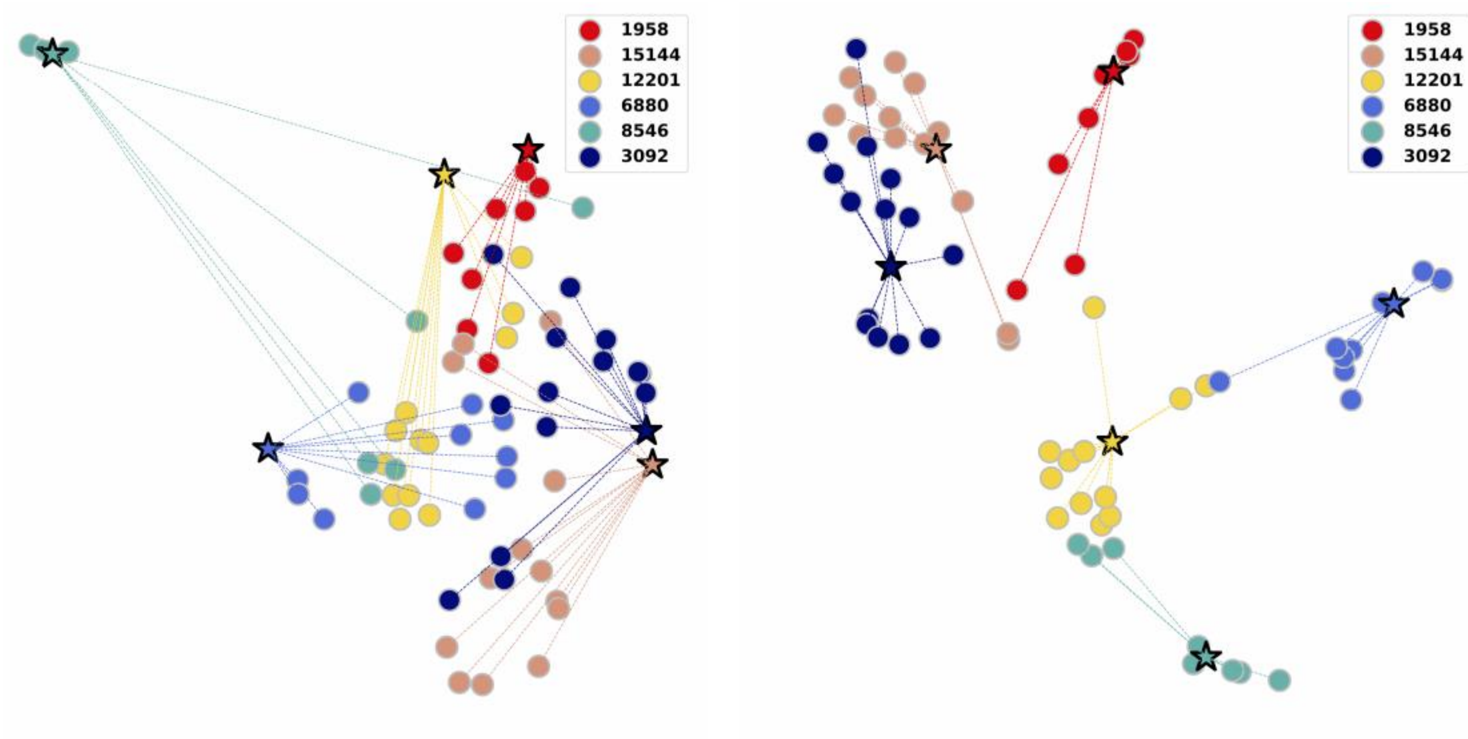




the number of embedding propagation layers {1,2,3,4}

- Increasing the depth of NGCF substantially enhances the recommendation cases.
- User similarity and collaborative signal are carried by the second- and third-order connectivity, respectively.

Effect of High-order Connectivity (2)



(a) MF (NGCF-0)   (b) NGCF-3

- The points with the same colors (i.e., the items consumed by the same users) tend to form the clusters.

- The **connectivities of users and items are well reflected in the embedding space**, that is, they are embedded into the near part of the space.

Take-home messages

- Modeling high-order connectivity from user-item interactions is important for CF.
- We proposed a GNN model to do this in an end-to-end way.

Future Work

- Incorporating knowledge graph into NGCF [Wang et al. KDD’19]
- Automating NGCF, e.g., negative sampling, hyper-parameters
- Keep Human-in-the-loop: conversational recommendation

THANK YOU!

Neural Graph Collaborative Filtering, SIGIR2019;
http://staff.ustc.edu.cn/~hexn/papers/sigir19-NGCF.pdf