210531 Special Lectures on Database (RecSys)

Neural Graph Collaborative Filtering

Heeseung Yun

heeseung.yun@vision.snu.ac.kr

Contents

- Motivation
- Approach
- Experiment
- Conclusion

Motivation

Rethinking Collaborative Filtering

- Two components of CF

	Embedding	Interaction
MF	Factorization	Inner product
Collaborative DL	NN	-
Neural CF	-	NN
Translation CF		Euclidean distance

$$\mathbf{R}_{ij} \sim \mathcal{N}(\mathbf{u}_i^T \mathbf{v}_j, \mathbf{C}_{ij}^{-1})$$

Wang et al. Collaborative Deep Learning for Recommender Systems. In KDD 2015.

Rethinking Collaborative Filtering

- Two components of CF

	Embedding	Interaction
MF	Factorization	Inner product
Collaborative DL	NN	-
Neural CF	-	NN
Translation CF		Euclidean distance

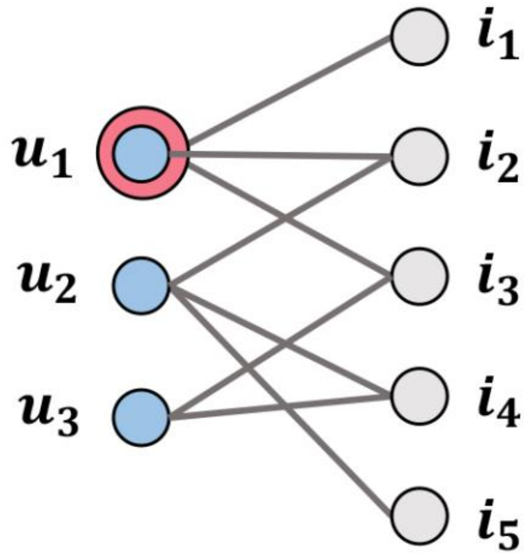
$$\mathbf{R}_{ij} \sim \mathcal{N}(\mathbf{u}_i^T \mathbf{v}_j, \mathbf{C}_{ij}^{-1})$$

- Question: Are they truly exploiting "collaborative signal"?
  - Not really… (only for objective function)
  - **Embedding** is suboptimal

Wang et al. Collaborative Deep Learning for Recommender Systems. In KDD 2015.

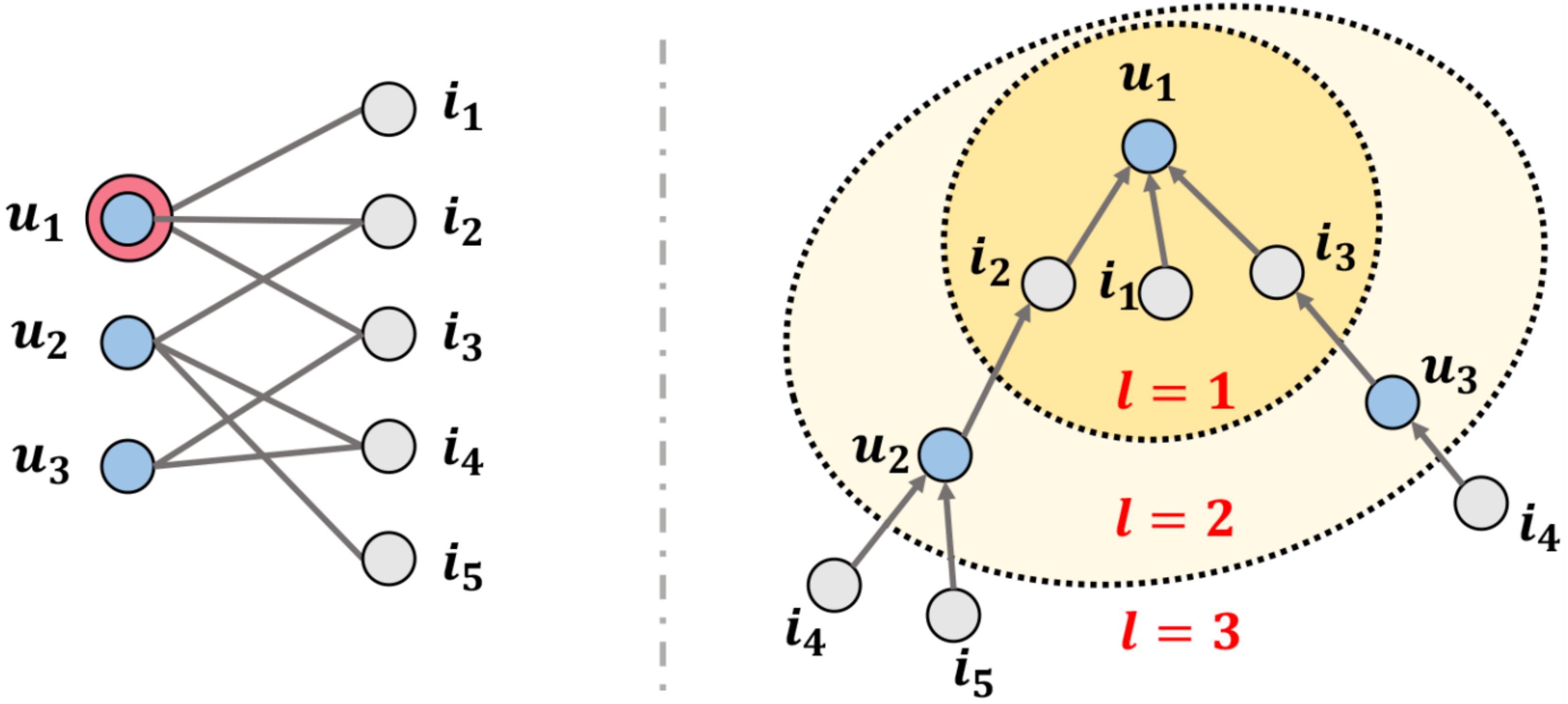
Rethinking Collaborative Filtering

- Gist: <u>High-order connectivity</u> from user-item interactions
  - Bipartite (1st order)



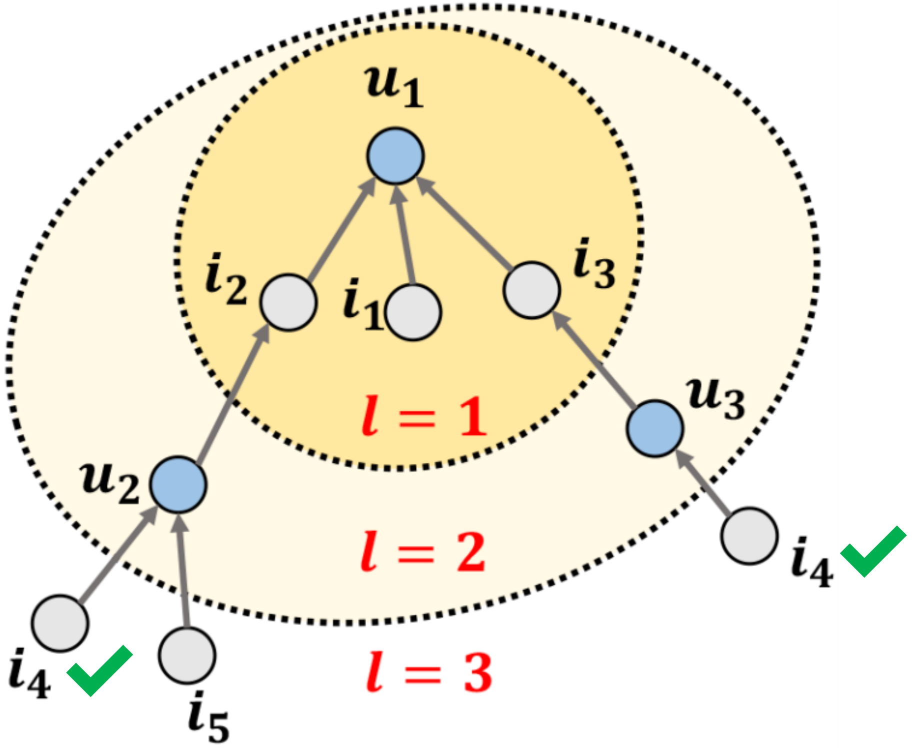
Rethinking Collaborative Filtering

- Gist: <u>High-order connectivity</u> from user-item interactions
  - Bipartite (1st order) → Tree (high order)



Rethinking Collaborative Filtering

- Gist: High-order connectivity from user-item interactions
  - Bipartite (1st order) → Tree (high order)
  - Embedding propagation
    - Recursive
    - Strength of information flow

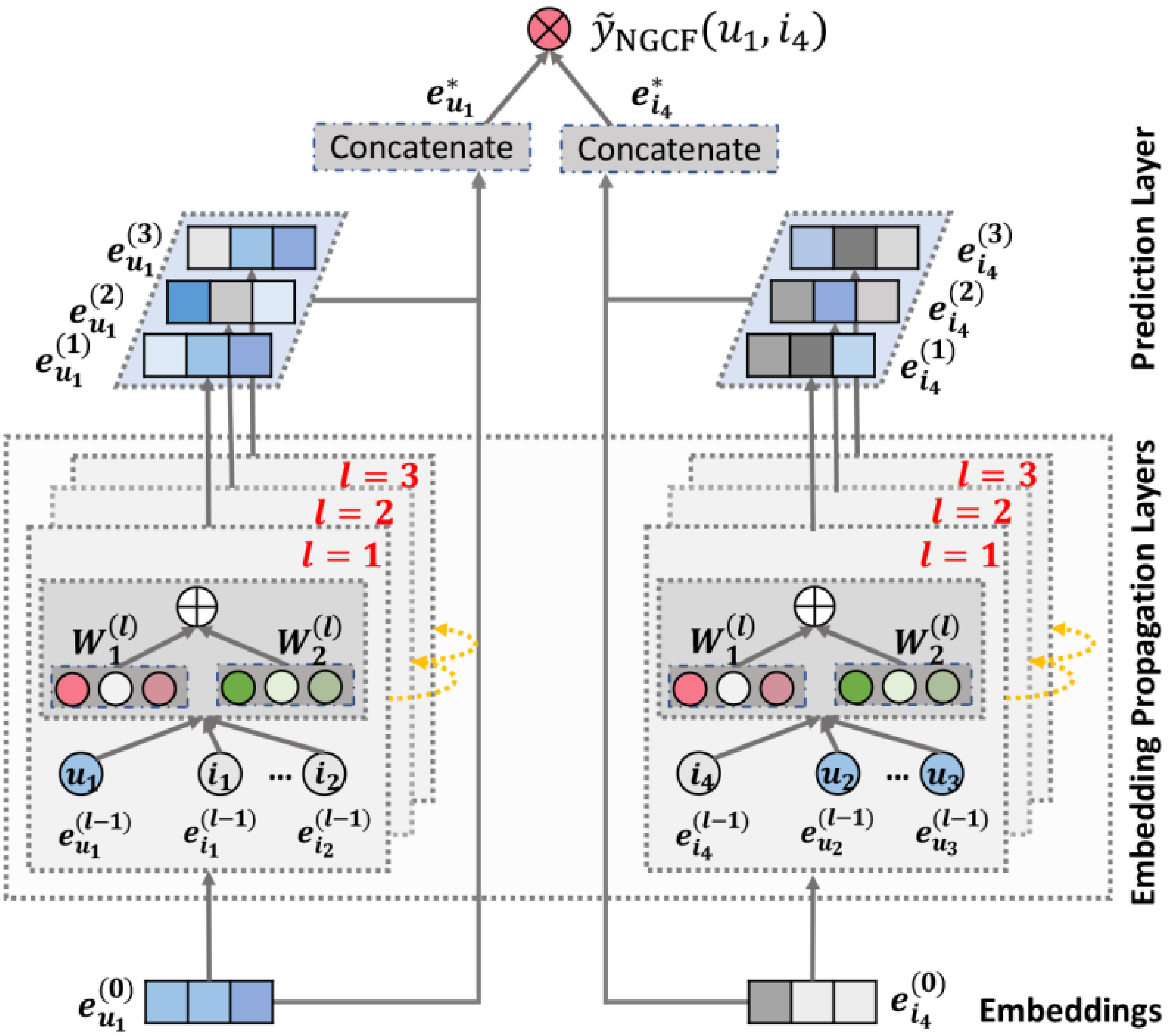


- Solution: **NGCF**

Approach

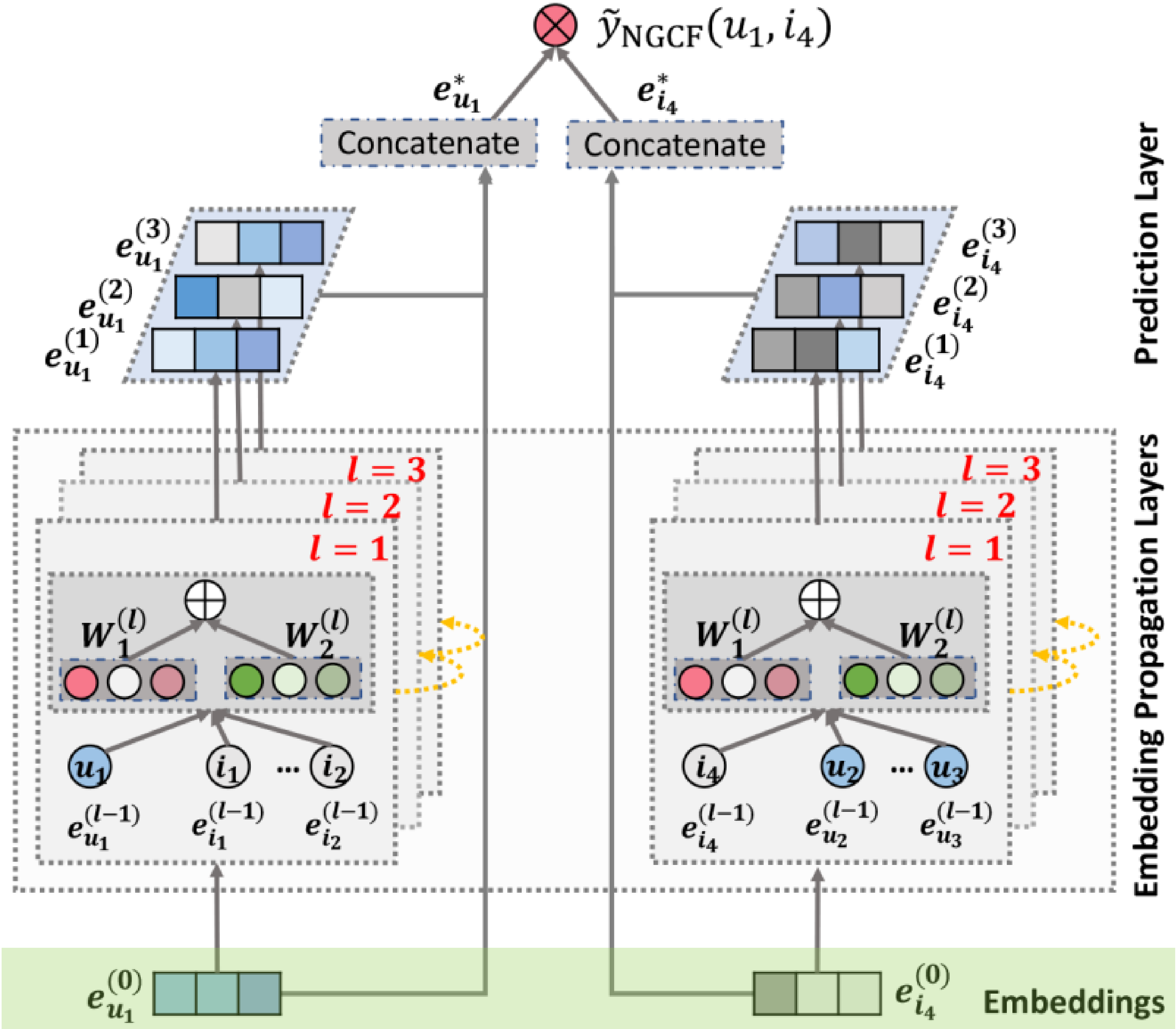
Model

- Embeddings
- Propagation
- Prediction



Model

- Embeddings
- Propagation
- Prediction



Embedding Layer

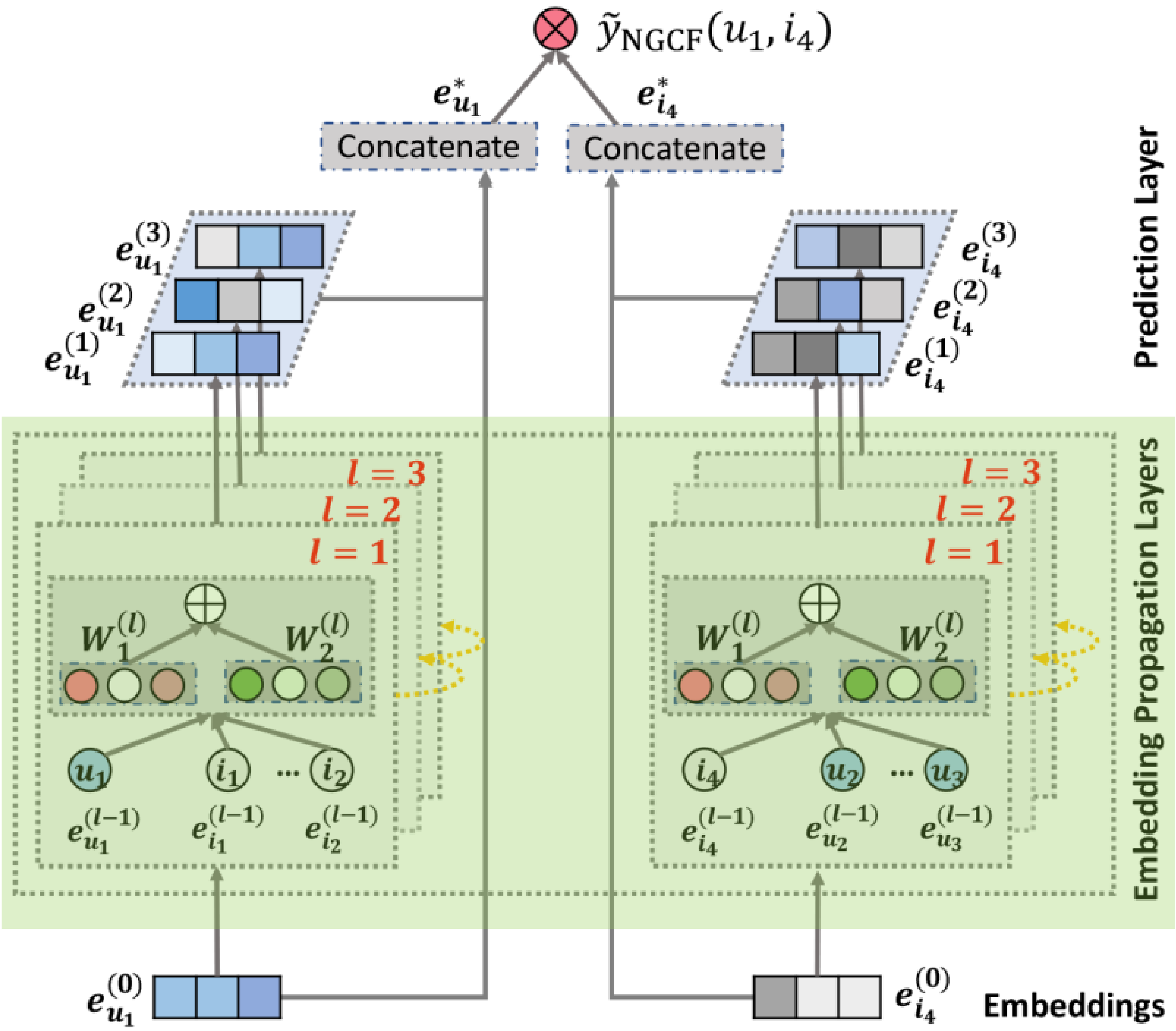
- Embedding matrix $\mathbf{E} \in \mathbb{R}^{(N+M)\times d}$

$$\mathbf{E} = [\ \mathbf{e}_{u_1}, \cdots, \mathbf{e}_{u_N}\ ,\ \mathbf{e}_{i_1}, \cdots, \mathbf{e}_{i_M}\ ]$$

  - Embeddings are refined for both training and <u>inference</u>
  - *i.e.,* model is used in inference stage

Model

- Embeddings
- Propagation
- Prediction



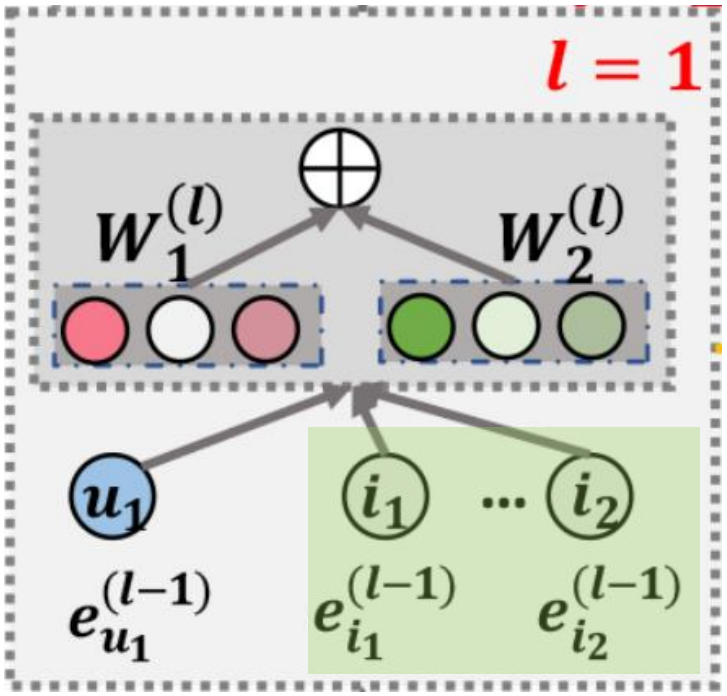
Embedding Propagation Layer

- Message Construction
  - Message = interaction unit

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

Interaction

p_{ui} = Propagation decay w.r.t. first-hop neighbors
(Graph Laplacian Norm)



- cf) GraphConv: $\mu_{j\rightarrow i,r} = \frac{1}{c_{ij}} W_r x_j$

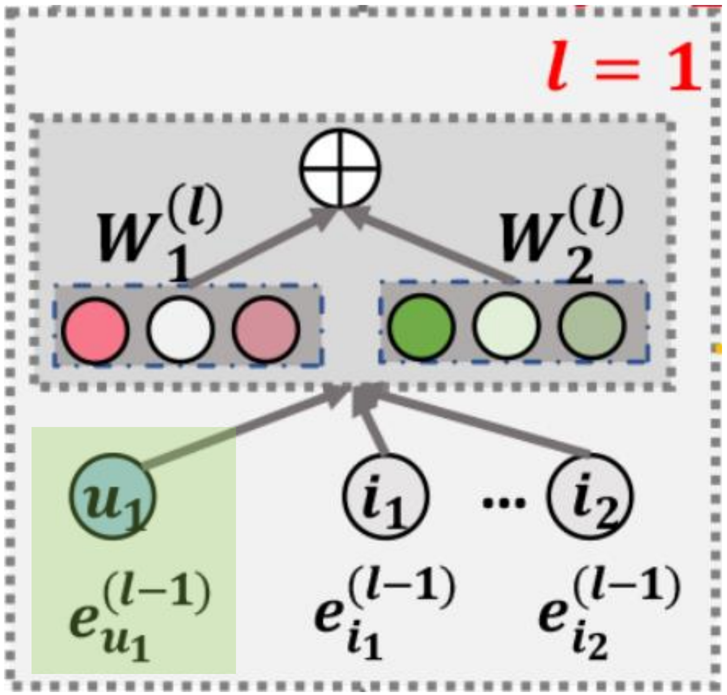
Van den Berg et al. Graph Convolutional Matrix Completion. In KDD 2017.

Embedding Propagation Layer

- Message Aggregation
  - LeakyReLU for encoding small negativity

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

$$\mathbf{m}_{u\leftarrow u} = \mathbf{W}_1\mathbf{e}_u$$



High-Order Propagation

- l-hop neighbors
  - Message Construction ($W^{(l)} \in \mathbb{R}^{d_l \times d_{l-1}}$)

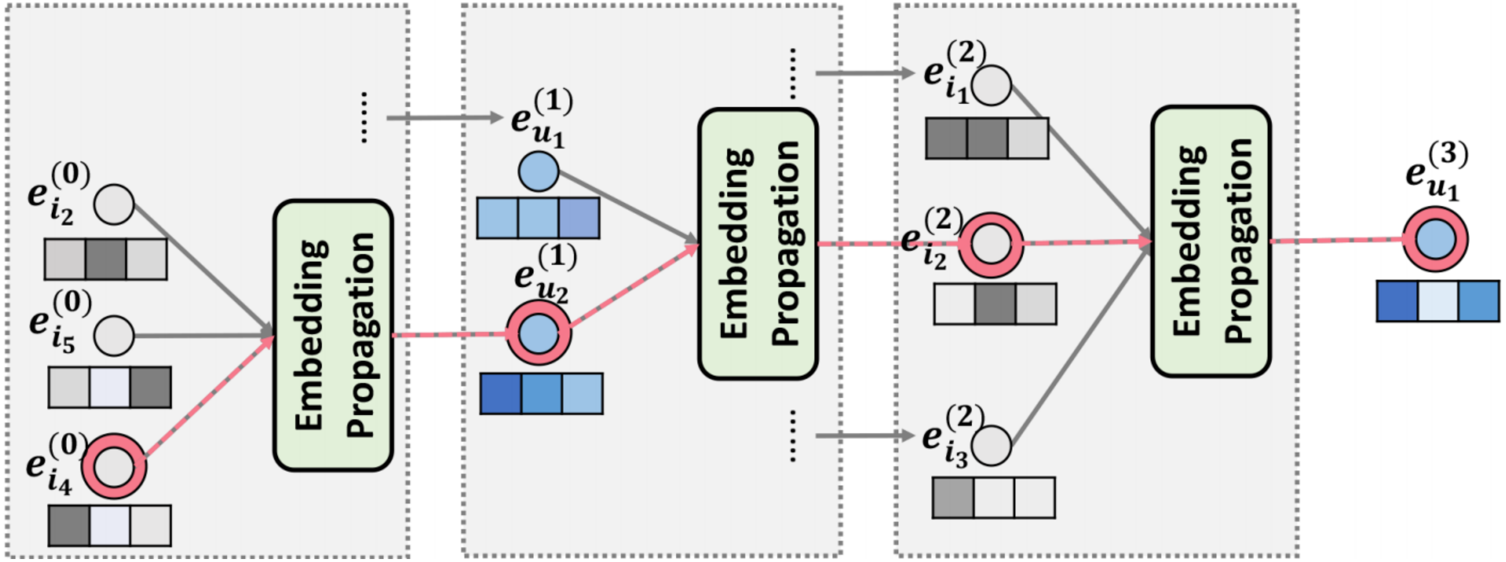
$$\begin{cases} \mathbf{m}_{u \leftarrow i}^{(l)} = p_{ui} \left(\mathbf{W}_1^{(l)} \mathbf{e}_i^{(l-1)} + \mathbf{W}_2^{(l)} (\mathbf{e}_i^{(l-1)} \odot \mathbf{e}_u^{(l-1)}) \right) \\ \mathbf{m}_{u \leftarrow u}^{(l)} = \mathbf{W}_1^{(l)} \mathbf{e}_u^{(l-1)}, \end{cases}$$

  - Message Aggregation

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

High-Order Propagation

- Message Aggregation



  - Problem: Sampling which nodes to visit

High-Order Propagation

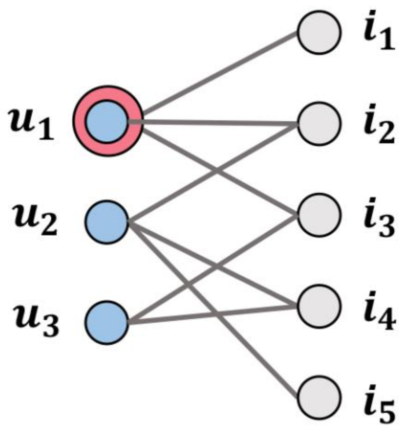
- Propagation in Matrix
  - Intuition: leverage item-user matrix representation

$$\mathbf{E}^{(l)} = \mathrm{LeakyReLU}\Big((\mathcal{L} + \mathbf{I})\mathbf{E}^{(l-1)}\mathbf{W}_1^{(l)} + \mathcal{L}\mathbf{E}^{(l-1)} \odot \mathbf{E}^{(l-1)}\mathbf{W}_2^{(l)}\Big)$$

$$\mathcal{L} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} \text{ and } \mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{R}^\top & \mathbf{0} \end{bmatrix}$$

Interaction matrix

  - Example



$$\mathbf{R} = \begin{matrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{matrix}$$

High-Order Propagation

- Propagation in Matrix
  - Intuition: leverage item-user matrix representation

$$\mathbf{E}^{(l)} = \mathrm{LeakyReLU}\Big((\mathcal{L} + \mathbf{I})\mathbf{E}^{(l-1)}\mathbf{W}_1^{(l)} + \mathcal{L}\mathbf{E}^{(l-1)} \odot \mathbf{E}^{(l-1)}\mathbf{W}_2^{(l)}\Big)$$

$$\mathcal{L} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} \text{ and } \mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{R}^\top & \mathbf{0} \end{bmatrix}$$

Interaction matrix

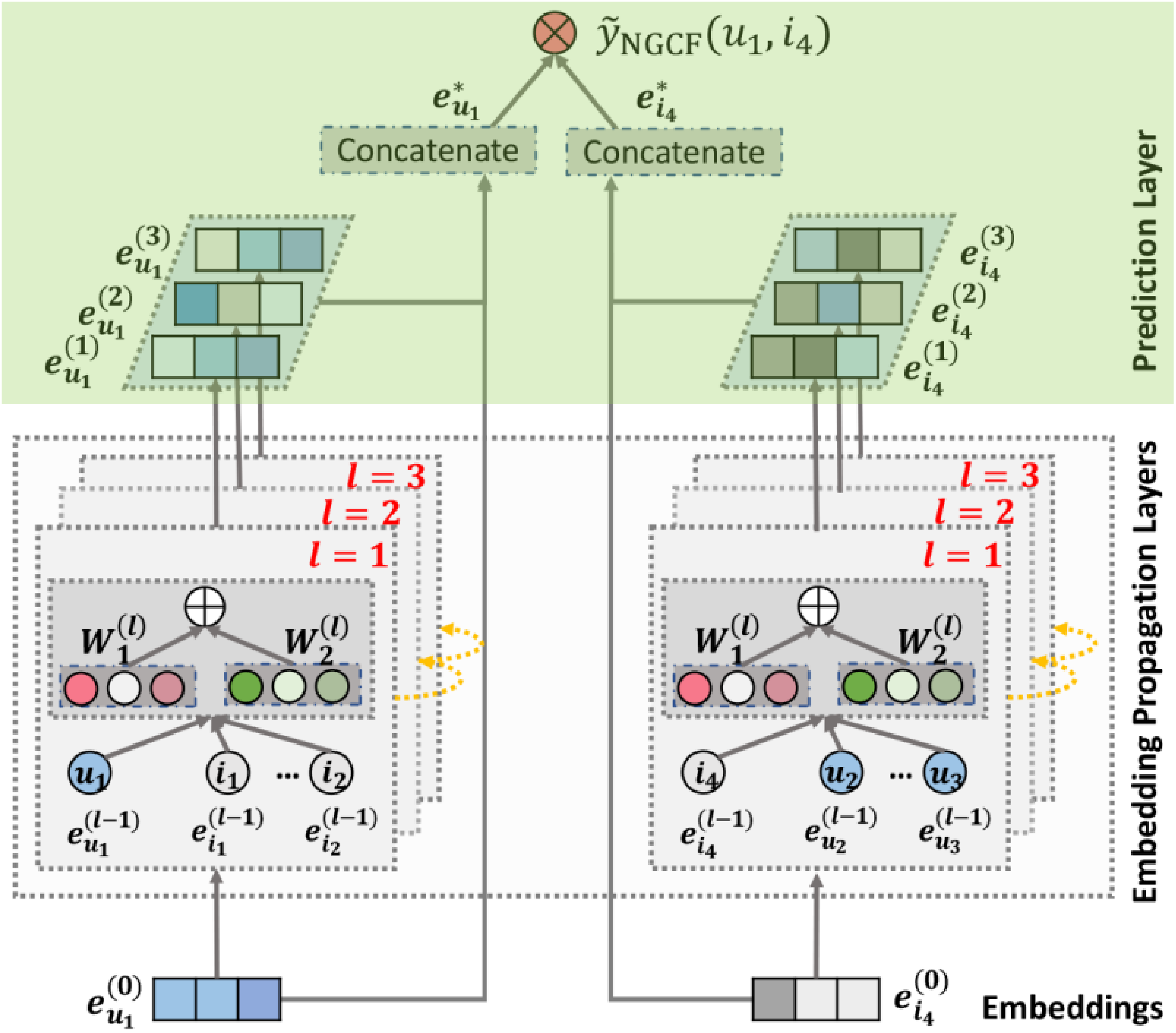
Degree matrix

$$D_{tt} = |\mathcal{N}_t|,$$

$$\mathcal{L}_{ui} = \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}}$$

Model

- Embeddings
- Propagation
- Prediction



Model Prediction

- Final user-item embeddings
  - Concatenation

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

  - cf) Weighted sum, attention, pooling, LSTM, etc.

- Predict preference
  - Inner product

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

Optimization

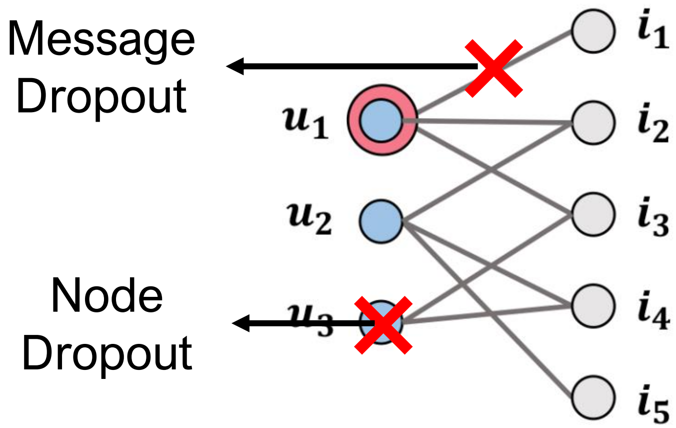
- Loss: Pairwise BPR with regularization
  - Observed > Unobserved

$$Loss = \sum_{(u,i,j)\in\mathcal{O}} -\ln \sigma(\hat{y}_{ui} - \underbrace{\hat{y}_{uj}}_{\text{Unobserved}}) + \lambda \|\Theta\|_2^2$$

  - Optimizer: Minibatch Adam
  - Triplet sampling: random

Optimization

- Dropout
  - Message Dropout & Node Dropout (per layer)
  - Training stage only



Complexity Analysis *Gowalla Dataset

- Space Complexity
  - $|\Theta_{NGCF}| = \underbrace{|\Theta_{MF}|}_{\text{Embeddings}} + \sum 2Ld_l d_{l-1} \ (L = 3)$
  - Example: MF (4.5M) vs. NGCF (4.524M)
- Time (Computational) Complexity
  - $O(\sum_{l=1}^{L} \underbrace{|\mathcal{R}^+|}_{\text{Nonzero entries in } \mathcal{L}} d_l d_{l-1} + \underbrace{\sum_{l=1}^{L} |\mathcal{R}^+| d_l}_{\text{Prediction Layer}})$
  - Training: MF (20s/epoch) vs. NGCF (80s/epoch)
  - Inference: MF (80s) vs. NGCF (260s)

Generalizability Analysis

- Comparison with SVD++
  - NGCF-1 without transformation matrix & nonlinear activation

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

$$\hat{y}_{\text{NGCF-SVD}} = \big(\mathbf{e}_u + \sum_{i' \in \mathcal{N}_u} p_{ui'} \mathbf{e}_{i'}\big)^{\top} \big(\mathbf{e}_i + \sum_{u' \in \mathcal{N}_i} p_{iu'} \mathbf{e}_i\big)$$

  - cf) SVD++

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T \left(p_u + |\mathrm{R}(u)|^{-\frac{1}{2}} \sum_{j \in \mathrm{R}(u)} y_j \right)$$

Generalizability Analysis

- Comparison with SVD++
  - NGCF-1 without transformation matrix & nonlinear activation

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

$$\hat{y}_{\text{NGCF-SVD}} = \Big(\mathbf{e}_u + \sum_{i'\in\mathcal{N}_u} p_{ui'}\mathbf{e}_{i'}\Big)^{\top}\Big(\mathbf{e}_i + \sum_{u'\in\mathcal{N}_i} \cancel{p_{iu'}}\mathbf{e}_i\Big)$$

0

- cf) SVD++

$$\hat{r}_{ui} = \mu + b_i + b_u + \boxed{q_i^T\left(p_u + |\mathrm{R}(u)|^{-\frac{1}{2}} \sum_{j\in\mathrm{R}(u)} y_j\right)}$$

Experiments

Dataset

- Gowalla
  - Check-in dataset
- Yelp2018
  - Local business (restaurants & bars) dataset for Yelp challenge
- Amazon-book
  - Subset of Amazon-review

- Settings
  - 10-core assumption
  - 70% : 10% : 20% split
  - recall@20, ndcg@20

Dataset	#Users	#Items	#Interactions	Density
Gowalla	29, 858	40, 981	1, 027, 370	0.00084
Yelp2018*	31, 668	38, 048	1, 561, 406	0.00130
Amazon-Book	52, 643	91, 599	2, 984, 108	0.00062

Baseline

- Factorization-based
  - Matrix Factorization (MF)
    - Trained with BPR loss
  - Neural CF (**NeuMF**)
    - NN-based prediction (interaction)
- Memory-based
  - Collaborative Memory Network (**CMN**)
    - Memory slots for similar neighboring users

Baseline

- Graph-based
  - High-order Proximity Recommender (**HOP-Rec**)
    - Random walk to enrich training data with multi-hop connected items
- Graph Convolution-based
  - PinSage
    - Industrial solution for item-item interaction, two-layered
  - Graph Convolution Matrix Completion (GC-MC)
    - Single-layered (first-order neighbor)
  - SpectralCF
    - Spectral convolution (eigen-decomposition is too heavy)

Experiment - Performance

- State-of-the-Art performance
  - with statistical significance

	Gowalla		Yelp2018*		Amazon-Book	
	recall	ndcg	recall	ndcg	recall	ndcg
MF	0.1291	0.1109	0.0433	0.0354	0.0250	0.0196
NeuMF	0.1399	0.1212	0.0451	0.0363	0.0258	0.0200
CMN	0.1405	0.1221	0.0457	0.0369	0.0267	0.0218
HOP-Rec	0.1399	0.1214	0.0517	0.0428	0.0309	0.0232
GC-MC	0.1395	0.1204	0.0462	0.0379	0.0288	0.0224
PinSage	0.1380	0.1196	0.0471	0.0393	0.0282	0.0219
NGCF-3	**0.1569***	**0.1327***	**0.0579***	**0.0477***	**0.0337***	**0.0261***
%Improv.	11.68%	8.64%	11.97%	11.29%	9.61%	12.50%
p-value	2.01e-7	3.03e-3	5.34e-3	4.62e-4	3.48e-5	1.26e-4

Experiment - Performance

- Naïve factorization performed worst
  - Only shallow interactions are taken into account

	Gowalla		Yelp2018*		Amazon-Book	
	recall	ndcg	recall	ndcg	recall	ndcg
MF	0.1291	0.1109	0.0433	0.0354	0.0250	0.0196
NeuMF	0.1399	0.1212	0.0451	0.0363	0.0258	0.0200
CMN	0.1405	0.1221	0.0457	0.0369	0.0267	0.0218
HOP-Rec	0.1399	0.1214	0.0517	0.0428	0.0309	0.0232
GC-MC	0.1395	0.1204	0.0462	0.0379	0.0288	0.0224
PinSage	0.1380	0.1196	0.0471	0.0393	0.0282	0.0219
NGCF-3	**0.1569***	**0.1327***	**0.0579***	**0.0477***	**0.0337***	**0.0261***
%Improv.	11.68%	8.64%	11.97%	11.29%	9.61%	12.50%
p-value	2.01e-7	3.03e-3	5.34e-3	4.62e-4	3.48e-5	1.26e-4

Experiment - Performance

- Memory or GCN-based were generally better
  - Memory-based (attending w.r.t. similarity), GCN (heuristic weighting)

	Gowalla		Yelp2018*		Amazon-Book	
	recall	ndcg	recall	ndcg	recall	ndcg
MF	0.1291	0.1109	0.0433	0.0354	0.0250	0.0196
NeuMF	0.1399	0.1212	0.0451	0.0363	0.0258	0.0200
CMN	<u>0.1405</u>	<u>0.1221</u>	0.0457	0.0369	0.0267	0.0218
HOP-Rec	0.1399	0.1214	<u>0.0517</u>	<u>0.0428</u>	<u>0.0309</u>	<u>0.0232</u>
GC-MC	0.1395	0.1204	0.0462	0.0379	0.0288	0.0224
PinSage	0.1380	0.1196	0.0471	0.0393	0.0282	0.0219
NGCF-3	**0.1569***	**0.1327***	**0.0579***	**0.0477***	**0.0337***	**0.0261***
%Improv.	11.68%	8.64%	11.97%	11.29%	9.61%	12.50%
p-value	2.01e-7	3.03e-3	5.34e-3	4.62e-4	3.48e-5	1.26e-4

Experiment - Performance

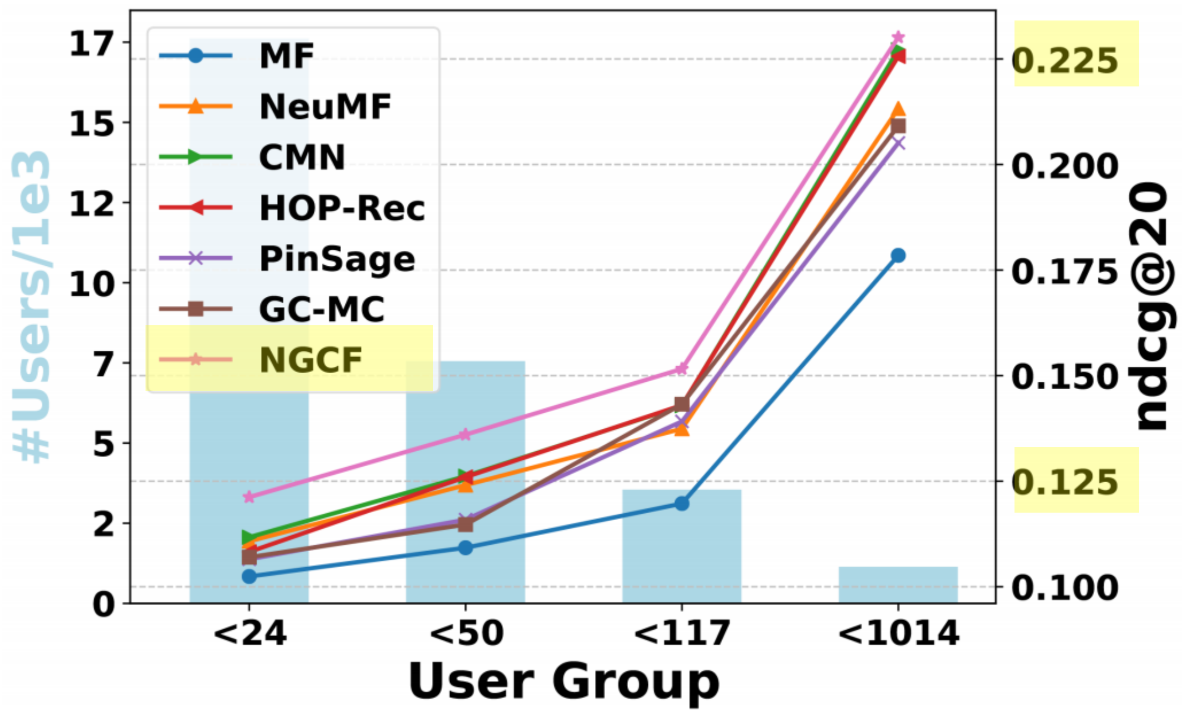
- Models with high-order propagation were generally better
  - But HOP-Rec heavily depends on random walk

	Gowalla		Yelp2018*		Amazon-Book	
	recall	ndcg	recall	ndcg	recall	ndcg
MF	0.1291	0.1109	0.0433	0.0354	0.0250	0.0196
NeuMF	0.1399	0.1212	0.0451	0.0363	0.0258	0.0200
CMN	0.1405	0.1221	0.0457	0.0369	0.0267	0.0218
HOP-Rec	0.1399	0.1214	0.0517	0.0428	0.0309	0.0232
GC-MC	0.1395	0.1204	0.0462	0.0379	0.0288	0.0224
PinSage	0.1380	0.1196	0.0471	0.0393	0.0282	0.0219
NGCF-3	**0.1569***	**0.1327***	**0.0579***	**0.0477***	**0.0337***	**0.0261***
%Improv.	11.68%	8.64%	11.97%	11.29%	9.61%	12.50%
p-value	2.01e-7	3.03e-3	5.34e-3	4.62e-4	3.48e-5	1.26e-4

Experiment - Sparsity

- Four user groups with same total interactions
  - ex. ~3% of users account for 25% of interaction
  - Huge discrepancy between interactive and non-interactive group (~2x)

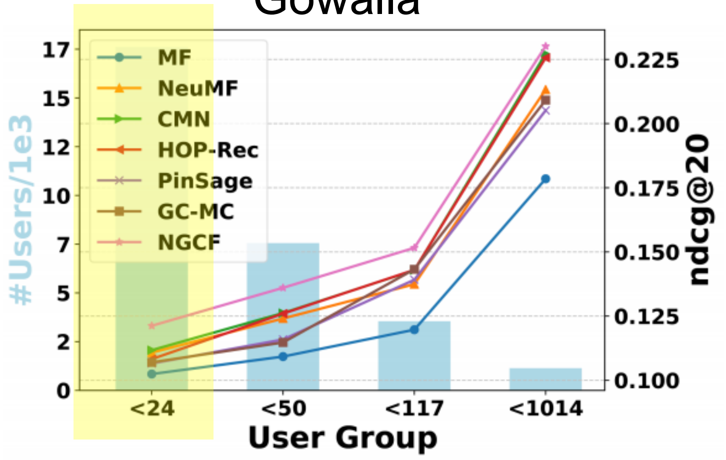
Gowalla Dataset



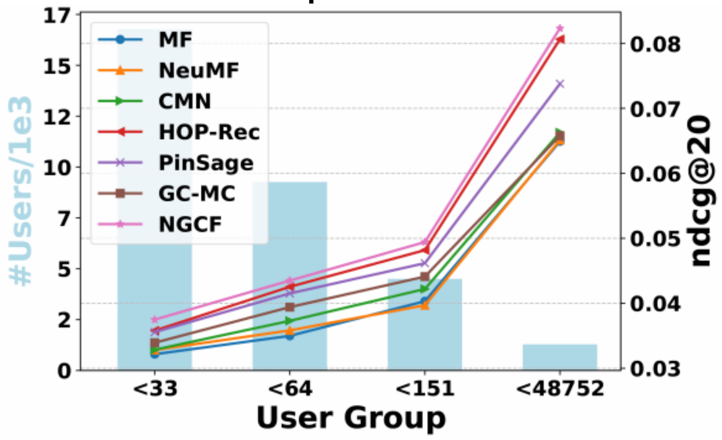
Experiment - Sparsity

- Four user groups with same total interactions
  - ex. ~3% of users account for 25% of interaction
  - Huge discrepancy between interactive and non-interactive group (~2x)
  - Performance gap widens for non-interactive groups

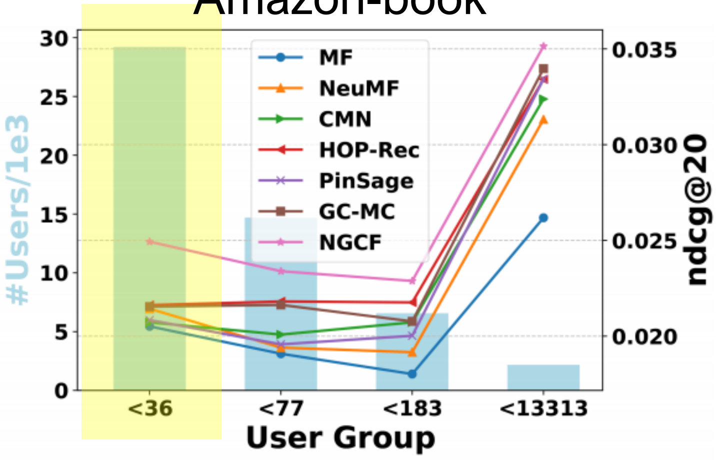
Gowalla



Yelp2018



Amazon-book



Experiment - Ablation

- Number of layers
  - NGCF-1 outperforms previous SotA
  - NGCF-4 displays signal of overfitting

	Gowalla		Yelp2018*		Amazon-Book	
	recall	ndcg	recall	ndcg	recall	ndcg
NGCF-1	0.1556	0.1315	0.0543	0.0442	0.0313	0.0241
NGCF-2	0.1547	0.1307	0.0566	0.0465	0.0330	0.0254
NGCF-3	0.1569	0.1327	**0.0579**	**0.0477**	0.0337	0.0261
NGCF-4	**0.1570**	**0.1327**	0.0566	0.0461	**0.0344**	**0.0263**
CMN	0.1405	0.1221	0.0457	0.0369	0.0267	0.0218
HOP-Rec	0.1399	0.1214	0.0517	0.0428	0.0309	0.0232

Experiment - Ablation

- Propagation Mechanism
  - Variants of NGCF-1
    - NGCF without transformation matrix & activation (SVD++)

$$\hat{y}_{\text{NGCF-SVD}} = \Big(\mathbf{e}_u + \sum_{i' \in \mathcal{N}_u} p_{ui'}\mathbf{e}_{i'}\Big)^\top \Big(\mathbf{e}_i + \sum_{u' \in \mathcal{N}_i} \cancelto{0}{p_{iu'}\mathbf{e}_i}\Big)$$

$$\hat{r}_{ui} = \mu + b_i + b_u + \boxed{q_i^T \left(p_u + |\mathrm{R}(u)|^{-\frac{1}{2}} \sum_{j \in \mathrm{R}(u)} y_j \right)}$$

Experiment - Ablation

- Propagation Mechanism
  - Variants of NGCF-1
    - NGCF without transformation matrix & activation (SVD++)
    - NGCF with GraphConv style message construction (GC-MC, PinSage)
  - NGCF-1 outperforms the rest

$$\mu_{j\to i,r} = \frac{1}{c_{ij}} W_r x_j$$

	Gowalla		Yelp2018*		Amazon-Book	
	recall	ndcg	recall	ndcg	recall	ndcg
NGCF-1	**0.1556**	**0.1315**	**0.0543**	**0.0442**	**0.0313**	**0.0241**
NGCF-$1_{\text{SVD++}}$	0.1517	0.1265	0.0504	0.0414	0.0297	0.0232
NGCF-$1_{\text{GC-MC}}$	0.1523	0.1307	0.0518	0.0420	0.0305	0.0234
NGCF-1_{PinSage}	0.1534	0.1308	0.0516	0.0420	0.0293	0.0231

Experiment - Ablation

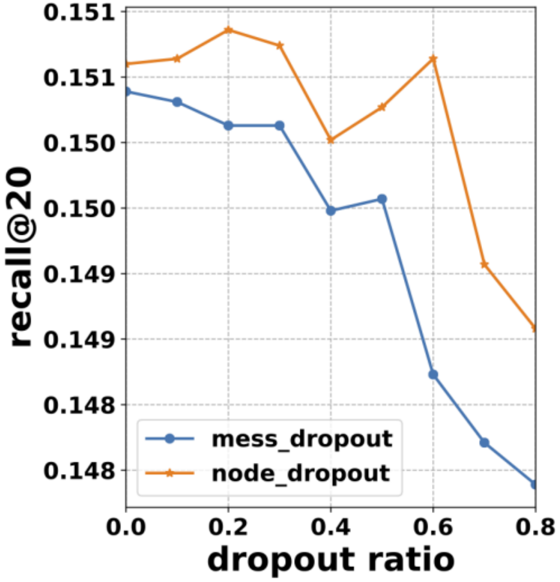
- Propagation Mechanism
  - Variants of NGCF-1
    - NGCF without transformation matrix & activation & item normalization (SVD++)
    - NGCF with GraphConv style message construction (GC-MC, PinSage)
  - NGCF-1 outperforms the rest
  - GraphConv variants outperforms original counterparts

	Gowalla		Yelp2018*		Amazon-Book	
	recall	ndcg	recall	ndcg	recall	ndcg
NGCF-1	**0.1556**	**0.1315**	**0.0543**	**0.0442**	**0.0313**	**0.0241**
NGCF-$1_{\text{SVD++}}$	0.1517	0.1265	0.0504	0.0414	0.0297	0.0232
NGCF-$1_{\text{GC-MC}}$	0.1523	0.1307	0.0518	0.0420	0.0305	0.0234
NGCF-1_{PinSage}	0.1534	0.1308	0.0516	0.0420	0.0293	0.0231
GC-MC	0.1395	0.1204	0.0462	0.0379	0.0288	0.0224
PinSage	0.1380	0.1196	0.0471	0.0393	0.0282	0.0219

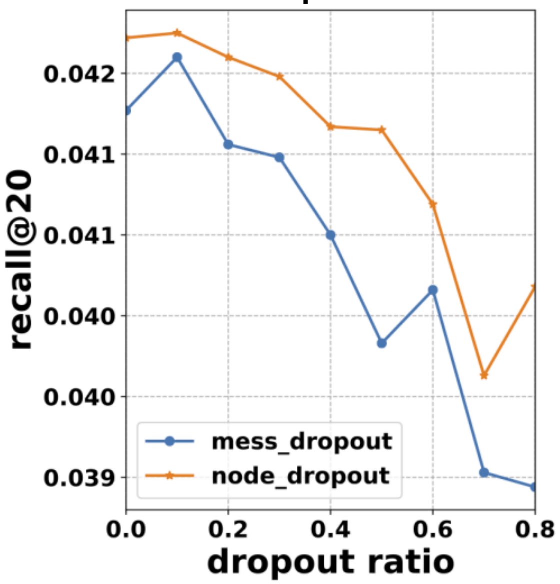
Experiment - Ablation

- Node & Message Dropout
  - <u>Node dropout</u> is generally better than message dropout
  - Optimal value varies w.r.t. dataset
  - (Note: optimal setup was 0.1 MD w/o ND)

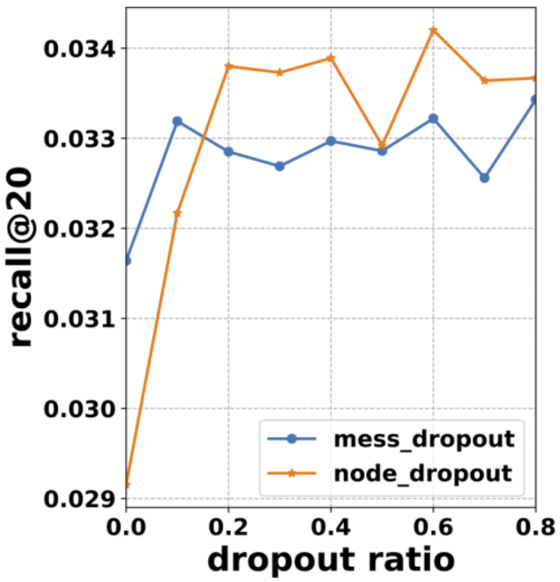
Gowalla



Yelp2018

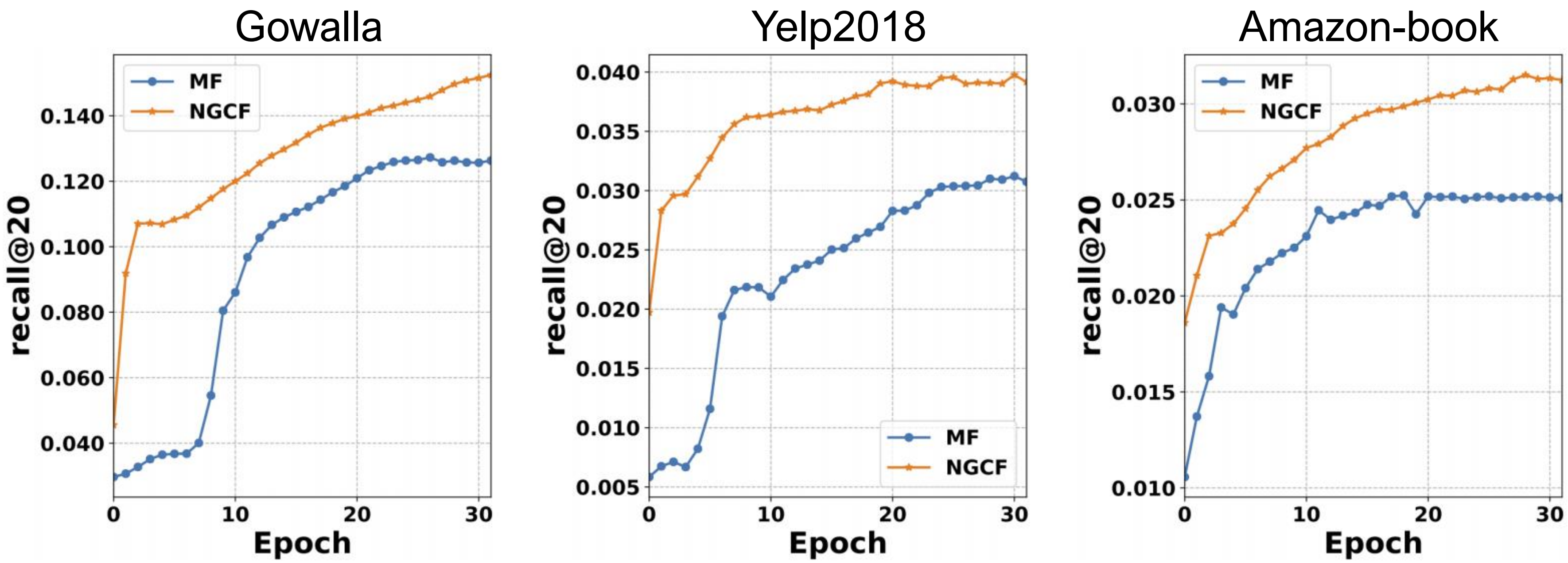


Amazon-book



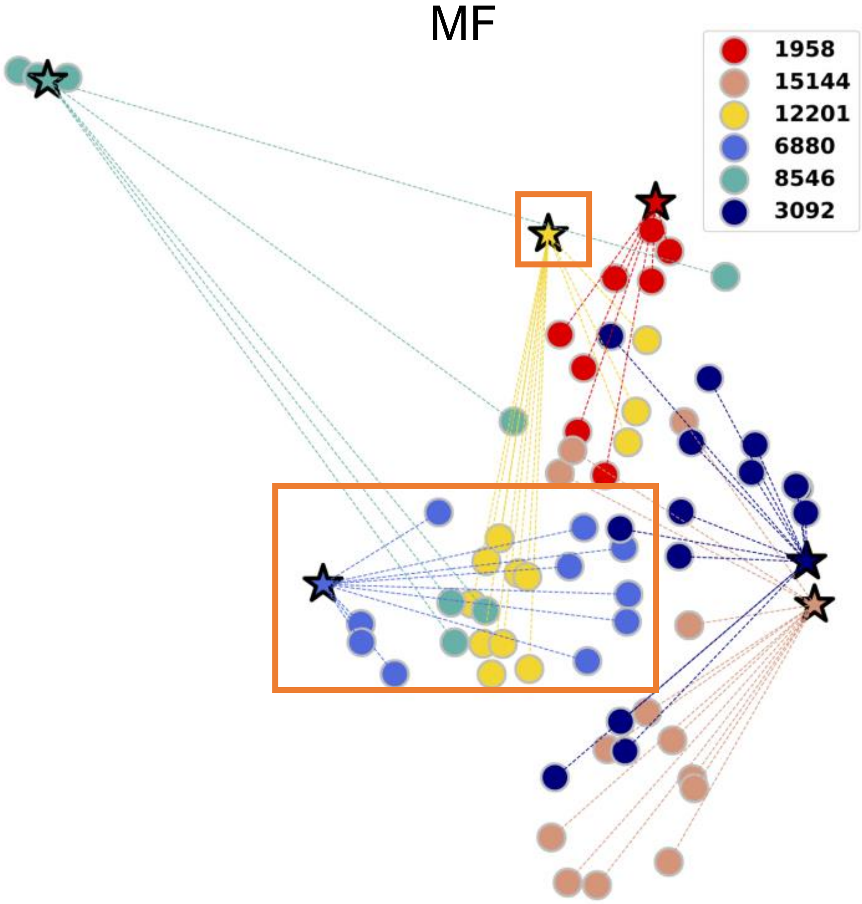
Experiment - Ablation

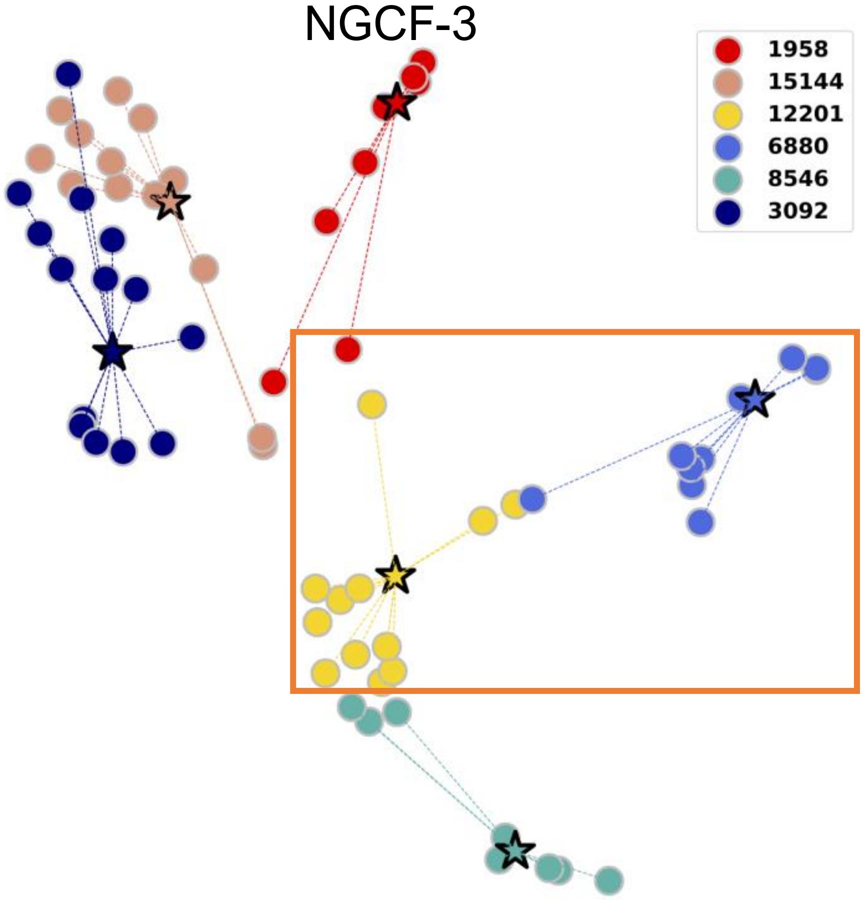
- Convergence
  - NGCF converges faster than MF
  - More interactions are involved in a single batch



Experiment - Qualitative

- t-SNE visualization





Conclusion

Conclusion

- Contribution
  - Importance of explicit exploitation of collaborative signal in embeddings
  - NGCF framework via embedding propagation
  - State-of-the-Art performance on 3 datasets
- Extension
  - NGCF with attention, adversarial learning, etc.
  - Application to other domains with structural knowledge

Thank You

Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, Tat-Seng Chua. Neural Graph Collaborative Filtering. In SIGIR 2019.

Appendix

Experiment Configuration

- Setup
  - Three propagation layers
  - Embedding size=64
  - Batch size=1024
  - Xavier initialization
  - Node dropout=0.0 / Message dropout=0.1
  - Hyperparameter optimization via grid search
  - Early stopping w.r.t. recall@20