Inductive Matrix Completion Based on Graph Neural Networks

Link **ICLR 2020**

Abstract

- A majority of existing matrix completion methods by factorizing the rating matrix are transductive.
- This paper proposes an Inductive Graph-based Matrix Completion(IGMC)
- It is possible to train inductive matrix completion models without using side information.
- It shows that a local graph patterns around a (user, item) pair are effective predictors of the rating this user gives to the item.

Introduction

Matrix completion


Harry Potter

The Triplets of Belleville


Shrek

The Dark Knight Rises


Memento

✔		✔	✔	
	✔			✔
✔	✔	✔		
			✔	✔

≈

1	.1
-1	0
.2	-1
.1	1

.9	-1	1	1	-.9
-.2	-.8	-1	.9	1

.88	-1.08	0.9	1.09	-0.8
-0.9	1.0	-1.0	-1.0	0.9
0.38	0.6	1.2	-0.7	-1.18
-0.11	-0.9	-0.9	1.0	0.91

Introduction

Local graph pattern

- In this paper, we propose a novel inductive matrix completion method that dose not use any content.
- Further, this method does not need to take the whole rating matrix as input, and can infer ratings for individual user-item pairs.
- A major class of link prediction methods are heuristic methods. However these traditional link prediction heuristics only work for simple graphs where nodes and edges both only have a single type
- We can build a bipartite graph from the rating matrix.
- Predicting unknown ratings => Predicting labeled links in the bipartite graph. This transforms matrix completion into a link prediction problem.

Introduction
Local graph pattern

- Can we find some heuristics for labeled link prediction in bipartite graph?

  If u_0 likes an item v_0, we may expect to see very often that v_0 is also liked by some other user u_1 who shares a similar taste to u_0

$$u_0 \rightarrow_{like} v_1 \rightarrow_{liked\ by} u_1 \rightarrow_{like} v_0$$

  If there many such paths between u_0 and v_0, we may infer that u_0 is highly likely to like v_0

  In fact, many neighborhood-based recommender systems rely on similar heuristics.

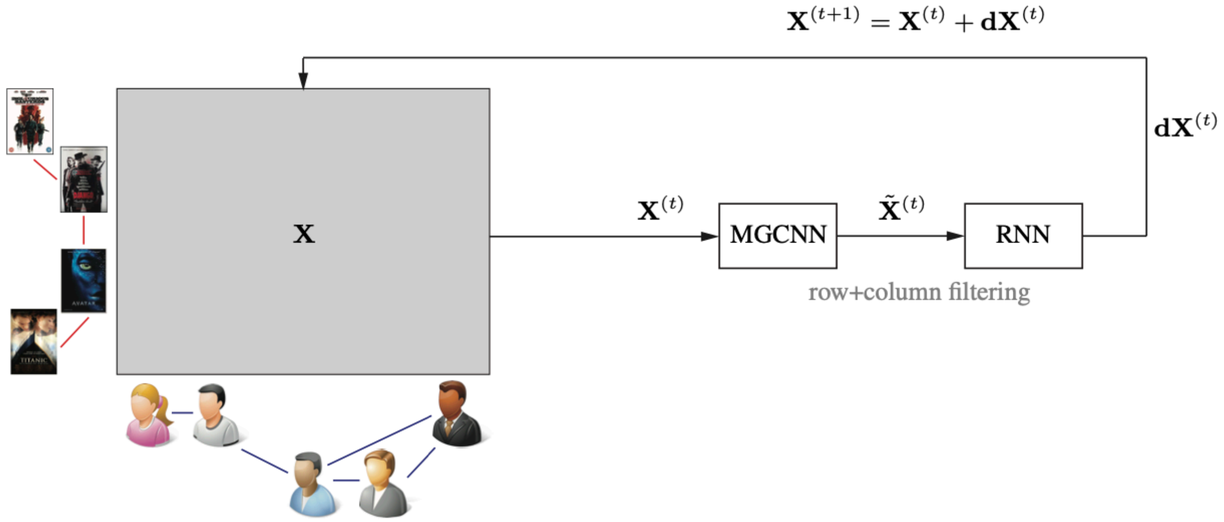
Introduction

Local graph pattern

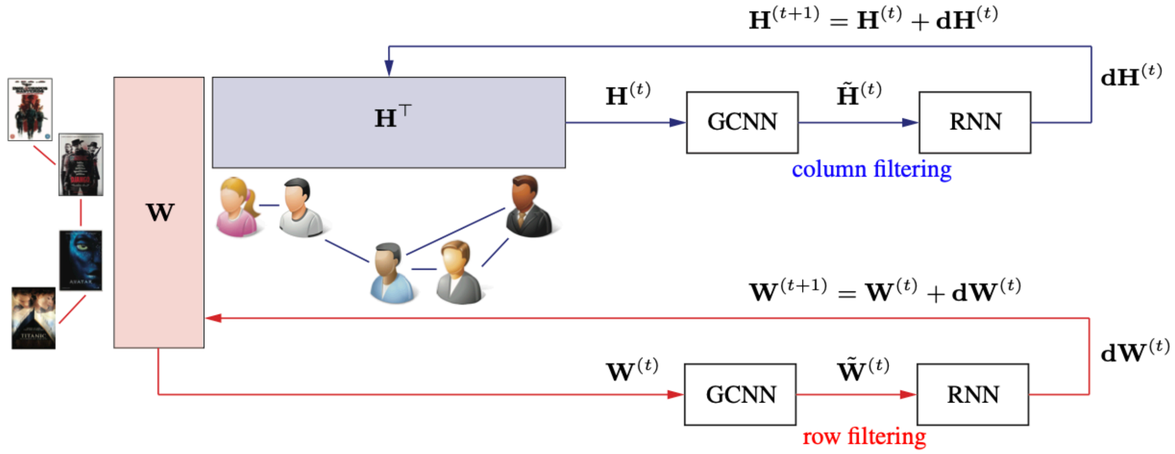
- We take a different approach that automatically learns suitable heuristics instead of trying to manually define many such intuitive heuristics and test their effectiveness.
- To do so, we first extract an h-hop enclosing subgraph for each training user-item pari (u, v).
- Subgraph regression(GNN) => Rating between (u,v)

Related Work GNNs for matrix completion

MGCNN [link]



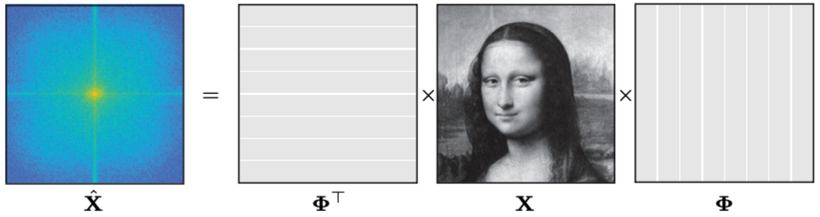
Recurrent MGCNN (RMGCNN)



Separable Recurrent MGCNN (sRMGCNN)

Geometric Matrix Completion with Recurrent Multi-Graph Neural Networks

MGCNN [link]



Column-wise trasform + Row-wise transform = 2D transform

Link

- Two-dimensional Fourier transform $\hat{X} = \Phi_r^T X \Phi_c$
- Multi-graph version of the spectral convolution
  $X \star Y = \Phi_r^T (\hat{X} \circ \hat{Y}) \Phi_c^T$
- Chebyshev polynomial filters of degree p
  $$\tilde{X} = \sum_{j,j'=0}^{p} \theta_{jj'} T_j(\tilde{\Delta}_r) X T_{j'}(\tilde{\Delta}_c)$$
- Loss function
  $$L(\Theta, \sigma) = ||X^{(T)}_{\Theta,\sigma}||^2_{g_r} + ||X^{(T)}_{\Theta,\sigma}||^2_{g_c} + \frac{\mu}{2} ||\Omega \circ (X^{(T)}_{\Theta,\sigma} - Y)||^2_F$$

Geometric Matrix Completion with Recurrent Multi-Graph Neural Networks

GC-MC [link]

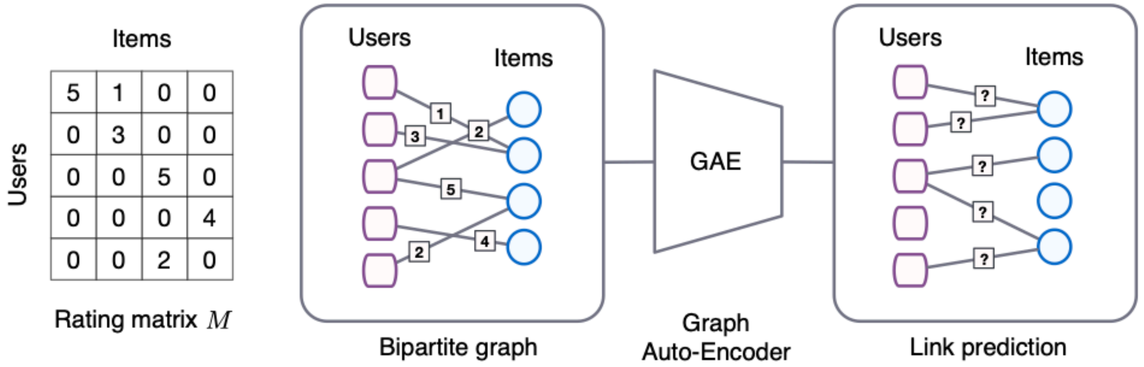
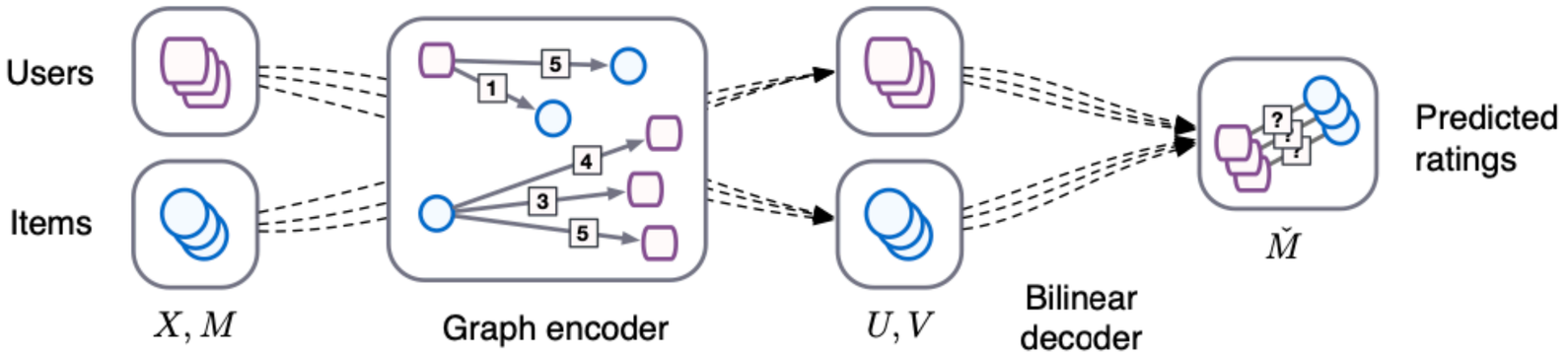


Figure 1: *Left*: Rating matrix M with entries that correspond to user-item interactions (ratings between 1-5) or missing observations (0). *Right*: User-item interaction graph with bipartite structure. Edges correspond to interaction events, numbers on edges denote the rating a user has given to a particular item. The matrix completion task (i.e. predictions for unobserved interactions) can be cast as a link prediction problem and modeled using an end-to-end trainable graph auto-encoder.

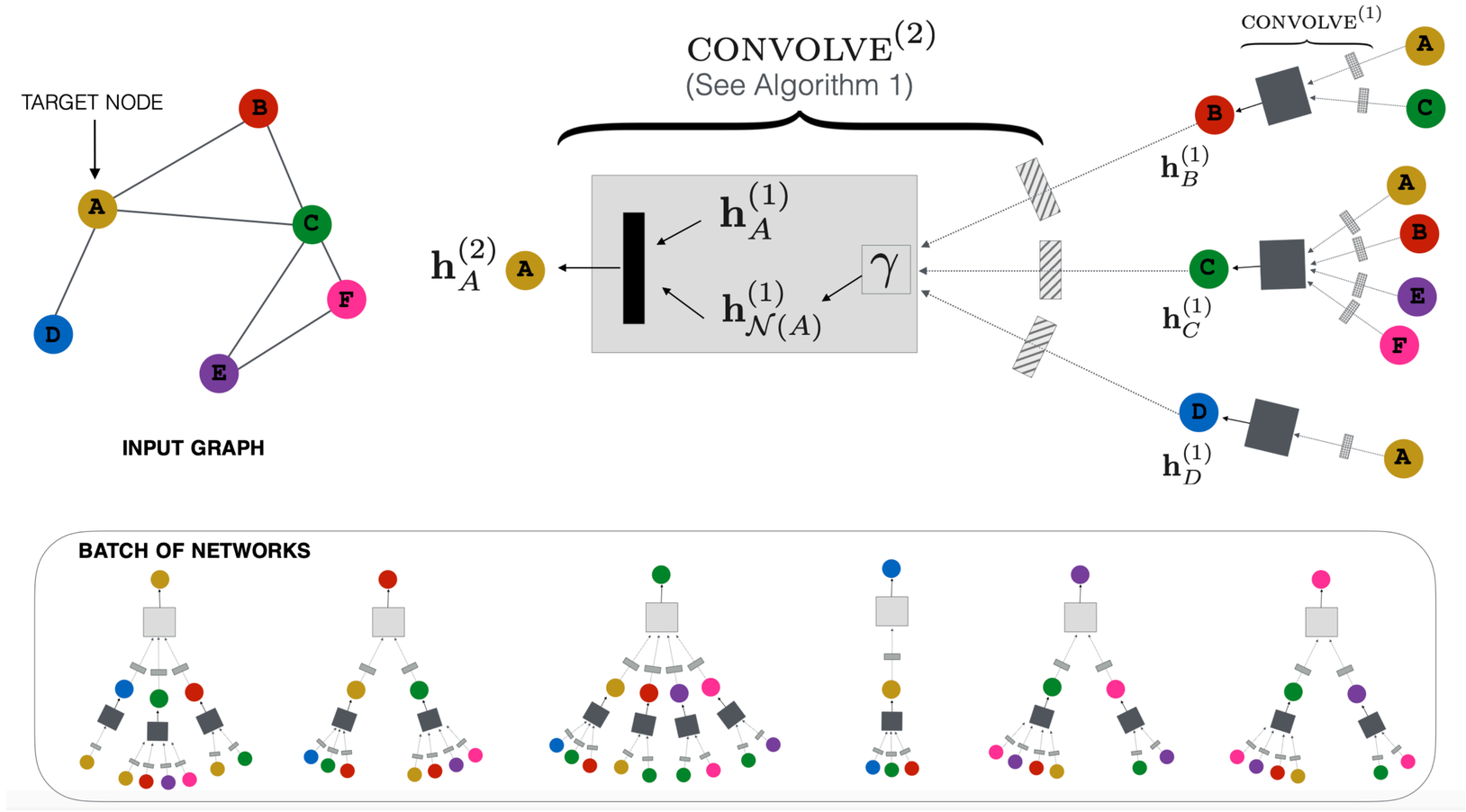


Graph Convolutional Matrix Completion

GC-MC [link]

- Matrix completion or recommendation can be cast as a link prediction problem on a bipartite user-item interaction graph.
- They propose a graph auto-encoder framework based on differentiable message passing on the bipartite interaction graph.
- Graph auto-encoders are comprised of a graph encoder model and a pairwise decoder model.

PinSage [link]



Graph Convolutional Neural Networks for Web-Scale Recommender Systems

PinSage [link]

- They develop a data efficient GCN algorithm PinSage, which combines efficient random walks and graph convolutions to generate embeddings of nodes.

Algorithm 1: CONVOLVE

Input : Current embedding $\mathbf{z}_u$ for node u; set of neighbor embeddings $\{\mathbf{z}_v | v \in \mathcal{N}(u)\}$, set of neighbor weights $\boldsymbol{\alpha}$; symmetric vector function $\gamma(\cdot)$

Output : New embedding $\mathbf{z}_u^{\text{NEW}}$ for node u

$\mathbf{n}_u \leftarrow \gamma\left(\{\text{ReLU}\left(\mathbf{Q}\mathbf{h}_v + \mathbf{q}\right) \mid v \in \mathcal{N}(u)\}, \boldsymbol{\alpha}\right)$;

$\mathbf{z}_u^{\text{NEW}} \leftarrow \text{ReLU}\left(\mathbf{W} \cdot \text{CONCAT}(\mathbf{z}_u, \mathbf{n}_u) + \mathbf{w}\right)$;

$\mathbf{z}_u^{\text{NEW}} \leftarrow \mathbf{z}_u^{\text{NEW}} / \|\mathbf{z}_u^{\text{NEW}}\|_2$

- If a query item q is given, we can recommend items whose embedding are the K-nearest neighbors of the query item's embedding.

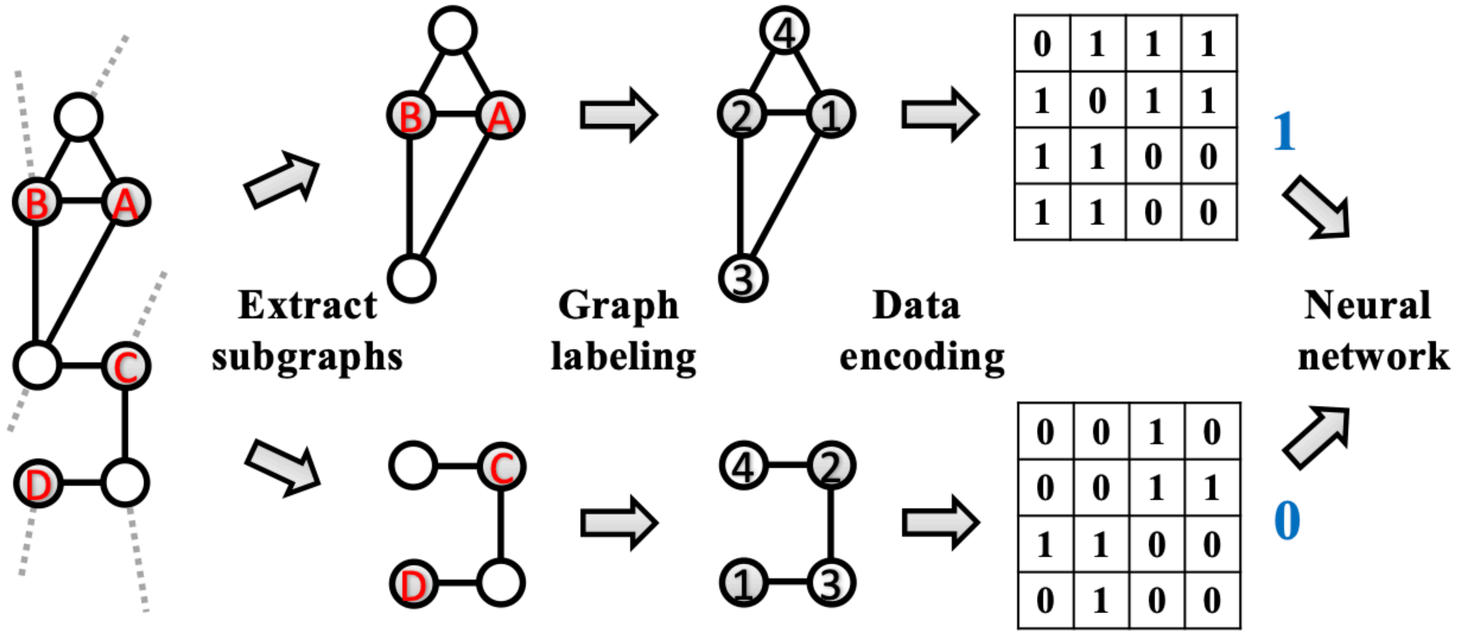
Graph Convolutional Neural Networks for Web-Scale Recommender Systems

GNNs for matrix completion

- Transductive
  - MGCNN and SpectralCF require graph Laplacians which do not generalize to new graphs.
  - GC-MC uses one-hot encoding of node IDs as initial node features thus cannot generalize to unseen users/items.
- Inductive
  - PinSage uses node content as initial node features which relies heavily on the rich visual and text content associated with the pins which is not often accessible in other recommendation task.

Related Work Link prediction based on graph patterns

WLNM [link]



Weisfeiler-Lehman Neural Machine for Link Prediction

WLNM [link]

- WLNM extracts an enclosing subgraph of each target link and encodes the subgraph as an adjacency matrix.
- After that, a neural network is trained on these adj. matrices to learn a predictive model.

Weisfeiler-Lehman Neural Machine for Link Prediction

SEAL [link]

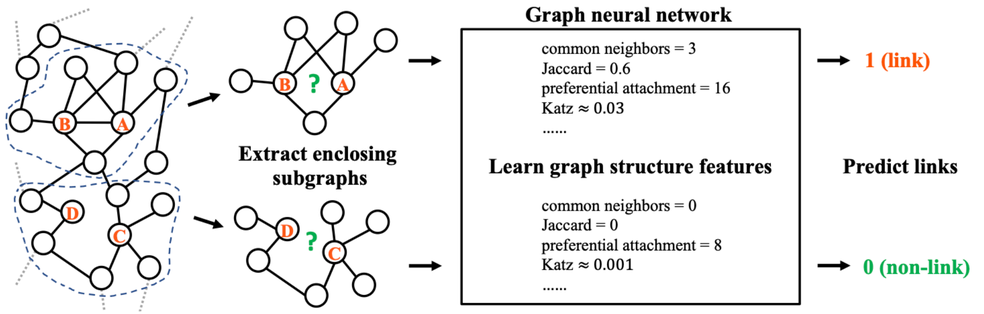


Figure 1: The SEAL framework. For each target link, SEAL extracts a local enclosing subgraph around it, and uses a GNN to learn general graph structure features for link prediction. Note that the heuristics listed inside the box are just for illustration – the learned features may be completely different from existing heuristics.

Link Prediction Based on Graph Neural Networks

Chen et al., 2005[link]; Zhou et al., 2007

- Traditional link prediction heuristics are adapted to bipartite graphs which show promising performance for recommender systems.

Link prediction approach to collaborative filtering.

Bipartite network projection and personal recommendation.

Li & Chen, 2013

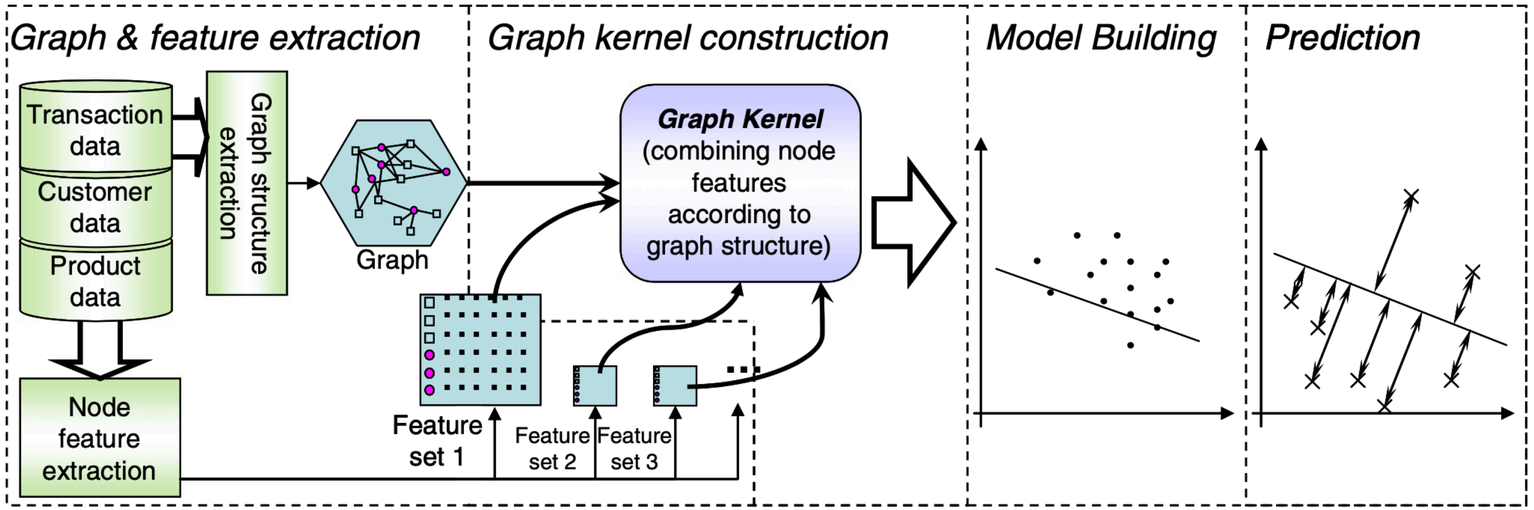


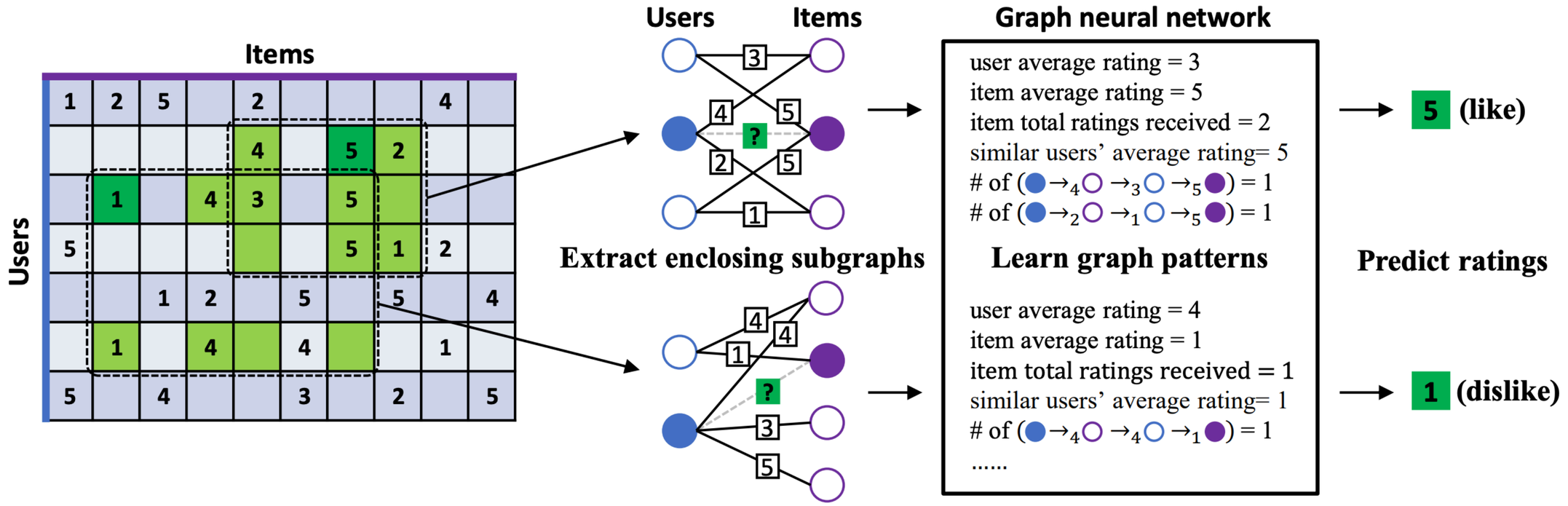
Fig. 1. A graph kernel-based recommendation framework.

Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach

Inductive Graph-Based Matrix Completion

- G : undirected bipartite graph constructed from the given rating matrix R.
- Nodes : either a user (u, corresponding to a row in R)
  or an item(v, corresponding to a column in R.
- Edges : (u, v) has a value r = $R_{u,v}$.
- $N_r(u)$: the set of u's neighbors that connect to u with edge type r.

IGMC


Items
Users
Users
Items
Extract enclosing subgraphs
Graph neural network
user average rating = 3
item average rating = 5
item total ratings received = 2
similar users' average rating= 5
of (●→4○→3○→5●) = 1
of (●→2○→1○→5●) = 1
Learn graph patterns
user average rating = 4
item average rating = 1
item total ratings received = 1
similar users' average rating= 1
of (●→4○→4○→1●) = 1
.....
Predict ratings
5 (like)
1 (dislike)

IGMC

1. Enclosing Subgraph Extraction

   - For each observed rating $R_{u,v}$, we extract an h-hop enclosing subgraphs.

2. Node Labeling

   - Before we feed an enclosing subgraph to the GNN, we first apply a node labeling to it, which gives an integer label to every node in the subgraph.

IGMC

3. GNN Architecture

- Message passing layers : Relational Graph Convolutional operator(R-GCN)

$$\mathrm{x}_i^{l+1} = \mathrm{W}_0^l \mathrm{x}_i^l + \sum_{r \in R} \sum_{j \in N_r(i)} \frac{1}{|N_r(i)|} \mathrm{W}_r^l \mathrm{x}_j^l$$

- Node i's feature vectors from different layers are concatenated as its final representation h_i

$$h_i = \mathrm{concat}(\mathrm{x}_i^1, \mathrm{x}_i^2, \ldots, \mathrm{x}_i^L) .$$

- Pooling layer :

$$g = \mathrm{concat}(h_u, h_v)$$

- Graph representation :

$$\hat{r} = \mathrm{w}^T \sigma(\mathrm{W} g)$$

IGMC

4. Model Training

- Loss function :

$$L = \frac{1}{|\{(u,v)\}|\Omega_{u,v}=1|} \sum_{(u,v):\Omega_{u,v}=1} (R_{u,v} - \hat{R}_{u,v})^2$$

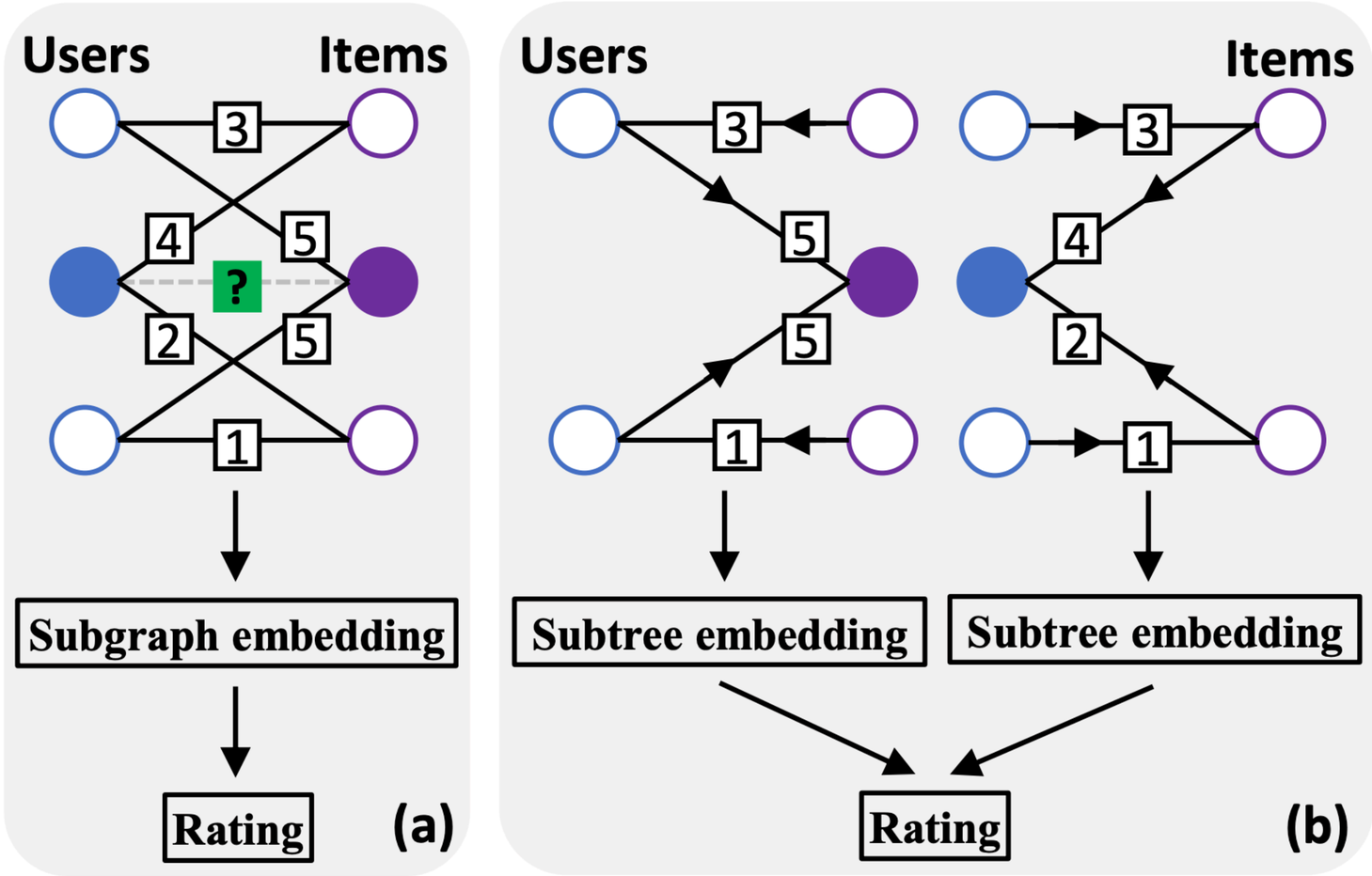
- Adjacent rating regularization :

$$L_{\text{ARR}} = \sum_{i=1,2,\ldots,|R|-1} \left\| \mathrm{W}_{r_{i+1}} - \mathrm{W}_{r_i} \right\|_F^2$$

- The final loss function :

$$L_{final} = L + \lambda L_{ARR}$$

Graph-Level GNN vs. Node-Level GNN



Experiments

Datasets

Dataset	Users	Items	Ratings	Density	Rating types
Flixster	3,000	3,000	26,173	0.0029	0.5, 1, 1.5, ..., 5
Douban	3,000	3,000	136,891	0.0152	1, 2, 3, 4, 5
YahooMusic	3,000	3,000	5,335	0.0006	1, 2, 3, ..., 100
ML-100K	943	1,682	100,000	0.0630	1, 2, 3, 4, 5
ML-1M	6,040	3,706	1,000,209	0.0447	1, 2, 3, 4, 5

5.1 FLIXSTER, DOUBAN AND YAHOOMUSIC

Table 2: RMSE test results on Flixster, Douban and YahooMusic.

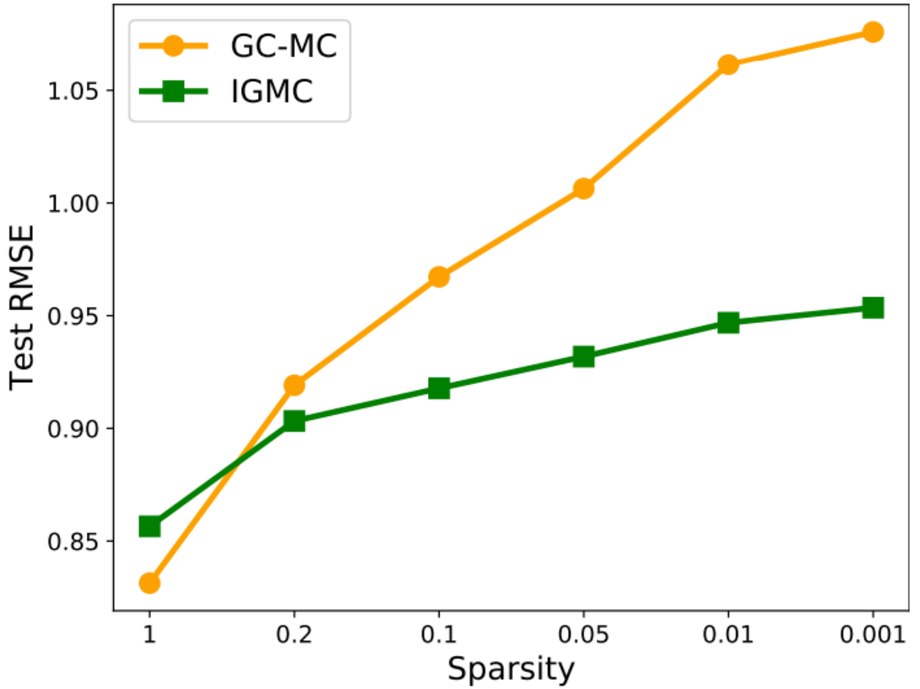
Model	Inductive	Content	Flixster	Douban	YahooMusic
GRALS	no	yes	1.245	0.833	38.0
sRGCNN	no	yes	0.926	0.801	22.4
GC-MC	no	yes	0.917	0.734	20.5
IGC-MC	yes	yes	0.999±0.062	0.990±0.082	21.3±0.989
F-EAE	yes	no	0.908	0.738	20.0
PinSage	yes	yes	0.954±0.005	0.739±0.002	22.9±0.629
IGMC (ours)	yes	no	**0.872**±0.001	**0.721**±0.001	**19.1**±0.138

5.2 ML-100K AND ML-1M

Table 3: RMSE test results on MovieLens-100K (left) and MovieLens-1M (right).

Model	Inductive	Content	ML-100K	Model	Inductive	Content	ML-1M
MC	no	no	0.973	PMF	no	no	0.883
IMC	no	yes	1.653	I-RBM	no	no	0.854
GMC	no	yes	0.996	NNMF	no	no	0.843
GRALS	no	yes	0.945	I-AutoRec	no	no	0.831
sRGCNN	no	yes	0.929	CF-NADE	no	no	**0.829**
GC-MC	no	yes	**0.905**	GC-MC	no	no	0.832
IGC-MC	yes	yes	1.142	IGC-MC	yes	yes	1.259
F-EAE	yes	no	0.920	F-EAE	yes	no	0.860
PinSage	yes	yes	0.951	PinSage	yes	yes	0.906
IGMC	yes	no	**0.905**	IGMC	yes	no	0.857

5.3 SPARSE RATING MATRIX ANALYSIS

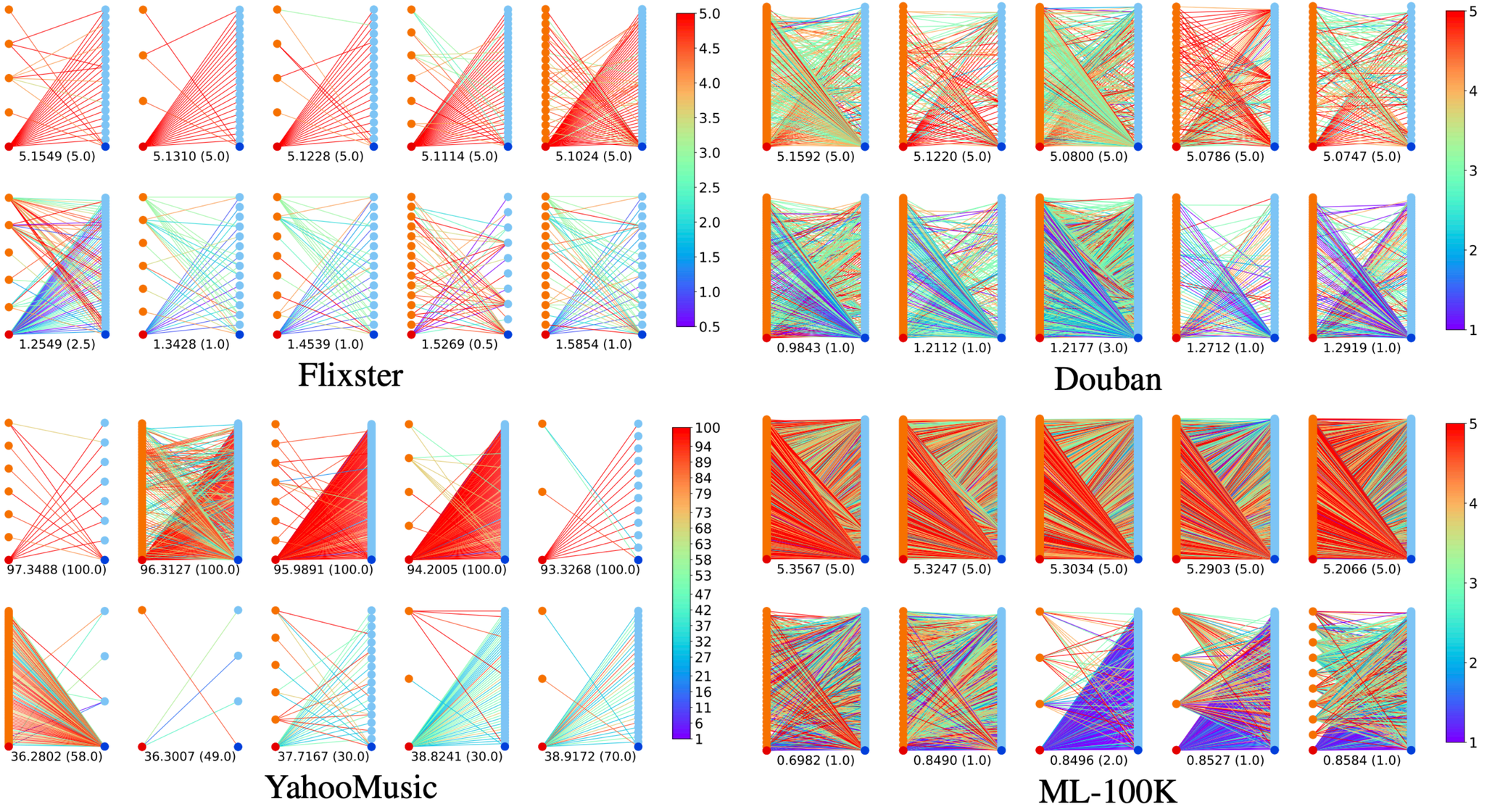


5.4 TRANSFER LEARNING

Table 4: RMSE of transferring the model trained on ML-100K to Flixster, Douban and YahooMusic.

Model	Inductive	Content	Flixster	Douban	YahooMusic
IGC-MC	yes	no	1.290	1.144	25.7
F-EAE	yes	no	0.987	0.766	23.3
IGMC (ours)	yes	no	**0.906**	**0.759**	**20.1**

5.6 VISUALIZATION



Conclusion

- IGMC learns local graph patterns related to ratings inductively based on GNN.
- IGMC does not rely on content (side info.) of users/items.
- IGMC is transferrable to new tasks without any retraining.