Multiplex Graph Neural Networks for Multi-behavior Recommendation

Weifeng Zhang
Zhejiang University
Hangzhou, China
zhangwf@zju.edu.cn

Jingwen Mao
Zhejiang University
Hangzhou, China
jingwenmao@zju.edu.cn

Yi Cao
Zhejiang University
Hangzhou, China
cao_yi@zju.edu.cn

Congfu Xu*
Zhejiang University
Hangzhou, China
xucongfu@zju.edu.cn

ABSTRACT

This paper focuses on the multi-behavior recommendation problem, i.e., generating personalized recommendation based on multiple types of user behaviors. Methods proposed recently usually leverage the ordinal assumption, which means that users' different types of behaviors should take place in a fixed order. However, this assumption may be too strong in some scenarios. In this paper, a more general model named Multiplex Graph Neural Network (MGNN) is proposed as a remedy. MGNN tackles the multi-behavior recommendation problem from a novel perspective, i.e., the perspective of link prediction in multiplex networks. By taking advantage of both the multiplex network structure and graph representation learning techniques, MGNN learns shared embeddings and behavior-specific embeddings for users and items to model the collective effect of multiple types of behaviors. Experiments conducted on both ordinal-behavior datasets and generic-behavior datasets demonstrate the effectiveness of the proposed MGNN model.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Multi-behavior recommendation; Multiplex networks; Graph neural networks

ACM Reference Format:
Weifeng Zhang, Jingwen Mao, Yi Cao, and Congfu Xu. 2020. Multiplex Graph Neural Networks for Multi-behavior Recommendation. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3340531.3412119

*Corresponding author.

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3412119

1 INTRODUCTION

In online information platforms, a large amount of various user behavior data are generated everyday since users interact with items in various ways. For example, a user may view, click or purchase items in an E-commerce platform. For recommender systems, such multi-behavior data are valuable, since different behaviors are correlated and the predictive accuracy of behaviors with few data may be improved by aggregating information from other behaviors. Therefore, in the paper, we focus on the multi-behavior recommendation problem, which is to generate recommendation w.r.t some target behaviors by utilizing multiple types of user behaviors.

Increasing attention has been paid on novel methods for multi-behavior recommendation. Existing methods usually extend traditional Matrix Factorization (MF) technique to integrate users' multi-behavior data. For instance, Collective Matrix Factorization (CMF) [7] and its extensions factorize multiple interaction matrices simultaneously, while correlating behaviors by adding constraints on the resulting latent factor matrices. Recently, Neural Multi-Task Recommendation (NMTR) [1] and chainRec [8] leverage the ordinal assumption, which means that users' different types of behaviors should take place in a fixed order, and achieve better performance than MF-extended methods on the ordinal-behavior datasets. However, this assumption may be too strong in some scenarios. For example, it is common that a user may add an item to his/her favorite list or share the item to other users, but in this situation, the occurrence order of these two behaviors may be different for different users.

The goal of this paper is to propose a general and effective model for multi-behavior recommendation problem, which does not rely on the ordinal assumption and is capable to model the collective effect of multiple behaviors. To achieve this goal, we first model the multi-behavior recommendation problem from a novel perspective, i.e., the perspective of link prediction in multiplex networks. And then we propose Multiplex Graph Neural Networks (MGNN) to tackle this problem. By making full use of the structure of the multiplex network and the power of graph neural networks, MGNN learns the embedding of each user/item w.r.t each behavior, and generates behavior-specific recommendation. The learned embedding consists of the shared embedding and the behavior-specific embedding, which is to capture the commonalities of the user/item shared by all behaviors and the characteristic of a particular behavior, respectively.

Overall, the main contributions of this paper are as follows:

- We study the multi-behavior recommendation problem from a novel perspective, i.e., the perspective of multiplex networks.
- We propose Multiplex Graph Neural Networks (MGNN), a novel model which takes advantage of both multiplex network structures and graph representation learning techniques, to tackle the multi-behavior recommendation problem.
- Experiments conducted on both ordinal-behavior datasets and generic-behavior datasets demonstrate the effectiveness of the proposed MGNN model.

2 RELATED WORK

Multi-behavior Recommendation. Existing methods usually extend traditional Matrix Factorization (MF) technique to integrate multi-behavior data. For example, Collective Matrix Factorization (CMF) [7] factorizes multiple interaction matrices simultaneously with shared user latent factors. Besides, Multi-Feedback BPR (MF-BPR) [3] extends the standard BPR [6] sampling method by assigning difference preference levels to different behaviors. Recently, methods leveraging the ordinal assumption are proposed. For instance, chainRec [8] explores the monotonic behavior chains, and Neural Multi-Task Recommendation (NMTR) [1] enforces that the prediction on a behavior relies on the predictions of the previous behaviors.

Multiplex Network Embedding. A multiplex network is a multi-layer network, in which each layer represents a particular type of relationship between nodes. Existing methods for multiplex network embedding mainly focus on dealing with the commonality and specificity of nodes in different layers. For example, Multi-task Network Embedding (MTNE) [9] extends the proximity-preserving embedding methods to multiplex networks and jointly learns node embeddings in different layers via enforcing an information-sharing embedding. Multiplex network Embedding via Learning Layer vectors (MELL) [4] enforces embeddings for the same node in different layers to be close and learns layer vectors to characterize the layers' connectivity. Recently, Deep Multiplex Graph Infomax (DMGI) [5] extends Deep Graph Infomax (DGI) to multiplex networks and introduces a consensus embedding on which all layer-specific embeddings should agree.

3 PROPOSED MODEL

3.1 Multi-behavior Data as Multiplex Networks

Let $\mathcal{U}$ denote the user set, $\mathcal{I}$ denote the item set, $\vec{Y} = \{Y^1, \cdots, Y^\alpha, \cdots, Y^m\}$ denote the user-item interaction matrices of m types of behaviors, where $Y^\alpha \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{I}|}$, and $\mathcal{T}$ denote the index set of target behaviors, where $\mathcal{T} \subset \{1, 2, \cdots, m\}$. The goal of multi-behavior recommendation is to recommender items that a user is most likely to interact with w.r.t each target behavior in $\mathcal{T}$.

The above multi-behavior recommendation problem can be represented as a link prediction problem on a corresponding multiplex network. Given $\mathcal{U}$, $\mathcal{I}$ and $\vec{Y}$, the corresponding multiplex network is an m-layer network $\mathcal{M}$. Figure 1 shows a toy example of multi-behavior data and the corresponding multiplex network. Each layer

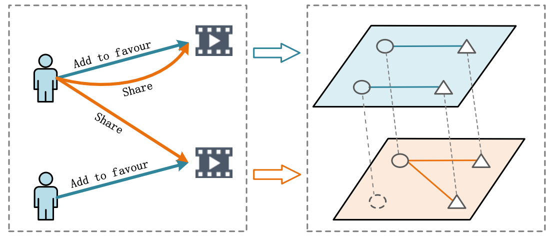

Figure 1: A toy example of multi-behavior data and the corresponding multiplex network.

of $\mathcal{M}$ is a bipartite graph $G_\alpha = (V, E_\alpha)$, which represents the user-item interactions under a certain behavior, and the adjacency matrix of G_α can be represented as $\boldsymbol{A}^\alpha = \begin{pmatrix} \boldsymbol{0} & \boldsymbol{Y}^\alpha \\ \boldsymbol{Y}^{\alpha T} & \boldsymbol{0} \end{pmatrix}$. Each user or item is represented by a node in each layer, even if the user/item has no interactions in that layer, so $V = \mathcal{U} \cup \mathcal{I}$. A node is called a *virtual node* in layer α if no other nodes in layer α are connected to it, and otherwise called an *active node*. Besides, in $\mathcal{M}$, in order to represent the connections between nodes corresponding to the same user/item in different layers, we introduce *interlayer links*. Specifically, the interlayer links between layer α and layer β can be represented by a $|V| \times |V|$ matrix $\boldsymbol{C}^{\alpha\beta}$. By arranging nodes in each layer in the same order, we have $\boldsymbol{C}^{\alpha\beta} = \boldsymbol{I}$, i.e., the identity matrix. Finally, by regarding the whole multiplex network $\mathcal{M}$ as a graph with N nodes, where $N = |V| \times m$, it can be represented by the following $N \times N$ supra-adjacency matrix

$$\tilde{\boldsymbol{A}} = \begin{pmatrix} \boldsymbol{A}^1 & \boldsymbol{C}^{12} & \cdots & \boldsymbol{C}^{1m} \\ \boldsymbol{C}^{21} & \boldsymbol{A}^2 & \cdots & \boldsymbol{C}^{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{C}^{m1} & \boldsymbol{C}^{m2} & \cdots & \boldsymbol{A}^m \end{pmatrix} = \begin{pmatrix} \boldsymbol{A}^1 & \boldsymbol{I} & \cdots & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{A}^2 & \cdots & \boldsymbol{I} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{I} & \boldsymbol{I} & \cdots & \boldsymbol{A}^m \end{pmatrix}.$$

3.2 Multiplex Graph Neural Networks

We propose our model, Multiplex Graph Neural Networks (MGNN) for multi-behavior recommendation. In the scenario of multi-behavior recommendation, a user may interact with an item in some but not all behaviors, which demonstrates the commonality and specificity of user preference w.r.t different behaviors. Therefore, in MGNN, we assume that the embedding of each node in the multiplex network can be separated into a shared embedding and a behavior-specific embedding.

3.2.1 Shared embeddings. To obtain the common information of each user/item in the multiplex network, MGNN constructs a *quotient graph*, which aggregates the interaction pattern of each layer into a single graph. Quotient graph is a technique of network analysis. Suppose nodes in a graph can be partitioned into several clusters, the *quotient graph* w.r.t this partition is defined as a weighted graph, in which each node represents a cluster and each edge represents the average connectivity between two clusters. In the multiplex network $\mathcal{M}$, we partition the entire N nodes into $|V|$ clusters according to which user/item they represent, and therefore the resulting quotient graph represents the average connectivity between a user and an item by aggregating their interactions under different behaviors. Formally, the adjacency matrix of the quotient graph w.r.t this partition is computed as

$$A_Q = \Gamma^{-\frac{1}{2}} S^T \tilde{A} S \Gamma^{-\frac{1}{2}}, \quad (1)$$

where $S \in \mathbb{R}^{N\times|V|}$ is the characteristic matrix of the partition, i.e., $S_{vi} = 1$ if node v represents user/item i and otherwise $S_{vi} = 0$, and Γ, the normalization matrix, is an $|V| \times |V|$ diagonal matrix with $[\gamma_1, \cdots, \gamma_i, \cdots, \gamma_{|V|}]$ on the main diagonal, where γ_i denotes the number of layers in which the user/item i is active.

After obtaining the quotient graph, we apply a graph neural network on it to learn the shared embeddings for users and items. Specifically, in our model, we adopt Graph Convolutional Networks (GCN) [2], due to its simplicity and effectiveness. Given the quotient graph with adjacency matrix A_Q and the initial shared embeddings, i.e., $E_U \in \mathbb{R}^{|\mathcal{U}|\times d}$ and $E_I \in \mathbb{R}^{|\mathcal{I}|\times d}$, where d is the dimension of the embedding space, an L-layer GCN updates the shared embeddings in the following way:

$$X_s^{(l)} = \sigma(\hat{D}_Q^{-\frac{1}{2}} \hat{A}_Q \hat{D}_Q^{-\frac{1}{2}} X_s^{(l-1)} W_s^{(l)}), \quad l = 1, 2, \cdots, L, \quad (2)$$

where $X_s^{(0)} = \begin{bmatrix} E_U \\ E_I \end{bmatrix}$, $\sigma(\cdot)$ is an activation function, $W_s^{(l)} \in \mathbb{R}^{d\times d}$ is a trainable weight matrix, $\hat{A}_Q = A_Q + I$, and $\hat{D}_Q$ is a diagonal matrix with $\hat{D}_{Q_{ii}} = \sum_j \hat{A}_{Q_{ij}}$. We take $X_s^{(L)}$ as the final shared embeddings.

3.2.2 node embeddings. Let $E_N \in \mathbb{R}^{N\times d}$ denote the behavior-specific embeddings. For each node in the multiplex network, the behavior-specific embedding and the corresponding final shared embedding are concatenated to form the initial node embedding, which is denoted as $X \in \mathbb{R}^{N\times 2d}$. These embeddings can be refined by applying an L-layer GCN on the entire multiplex network:

$$X^{(l)} = \sigma(\hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}} X^{(l-1)} W^{(l)}), \quad l = 1, 2, \cdots, L, \quad (3)$$

where $X^{(0)} = X$, $\hat{A} = \tilde{A} + I$, and $\tilde{A}$ is the supra-adjacency matrix of the multiplex network.

Remark. We further analyze the behavior of Eq.(3) from the perspective of neighborhood aggregation. For a node v in layer α, which represents a user or an item in that layer, Eq.(3) updates the embedding of this node as

$$x_v^{\alpha\,(l)} = \sigma(W^{(l)} \sum_w \frac{1}{\alpha_{vw}} x_w^{\alpha\,(l-1)}), \quad w \in \{v \cup \mathcal{N}^\alpha(v) \cup C(v)\}, \quad (4)$$

where α_{vw} is the normalized weight, $\mathcal{N}^\alpha(v)$ is the set of neighbors of node v in layer α, and $C(v)$ is the set of nodes that correspond to the same user/item as node v. Therefore, MGNN aggregates the embedding of not only v's neighbors in layer α (as GCN does), but also nodes in other different layers that corresponds to the same user/item as node v, which allows information to be transfered across different layers. By leveraging the information from different layers, MGNN is able to learn better representation for each node, especially for virtual nodes and nodes with few interactions in a certain layer. This is the main difference between MGNN and existing multiplex network embedding methods.

3.2.3 Prediction and Optimization. We concatenate all representations learned by the second L-layer GCN to obtain the final node embedding:

$$x_v^\alpha = [x_v^{\alpha\,(0)}\, x_v^{\alpha\,(1)} \cdots x_v^{\alpha\,(L)}] \quad (5)$$

Given a user u, an item i and a behavior α, we conduct inner product to predict the user's preference for the item w.r.t behavior α:

$$\hat{y}_{ui}^\alpha = {x_u^\alpha}^T x_i^\alpha. \quad (6)$$

For each target behavior $\alpha \in \mathcal{T}$, we adopt the pairwise BPR loss [6], and jointly learn the representation of each user/item w.r.t each behavior with the following loss function:

$$\mathcal{L} = \sum_{\alpha\in\mathcal{T}} \lambda^\alpha (\sum_{(u,i,j)\in\mathcal{D}^\alpha} -\ln \sigma\left(\hat{y}_{ui}^\alpha - \hat{y}_{uj}^\alpha\right) + \lambda_\theta ||\Theta||_2^2), \quad (7)$$

where σ is the sigmoid function, i.e., $\sigma(x) = \frac{1}{1+e^{-x}}$, $\mathcal{D}^\alpha$ is the dataset with negative samples, Θ denotes all trainable parameters, λ_θ controls the strength of L_2 regularization, and λ^α controls the importance of behavior α in the joint training with $\sum_\alpha \lambda^\alpha = 1$.

4 EXPERIMENTS

4.1 Experimental Settings

4.1.1 Datasets. We conduct extensive experiments on two ordinal-behavior datasets, i.e. *YooChoose*[1] and *Steam*[2], and one generic-behavior dataset, *Douban Book*[3]. We discard users and items with fewer than five interactions. The basic statistics of the three dataset are summarized in Table 1. In experiment, all types of behaviors are regarded as the target behaviors. For each user, we randomly sample two interacted items to form the validation set and the test set. To evaluate the performance, we adopt two widely-used metrics, ***HR@K*** and ***NDCG@K***. $K = 20$ for the ordinal-behavior datasets, and $K = 50$ for Douban Book since it is quite sparse.

4.1.2 Baselines and Hyper-parameter Settings. We study the performance of the following methods: **BPRMF** [6], **GCN** [2], **CMF** [7], **MELL** [4], **NMTR** [1], the proposed model **MGNN** and two variants of **MGNN**, i.e., $\mathbf{MGNN}_{specific}$ (which discards shared embeddings and only contains the behavior-specific embeddings) and $\mathbf{MGNN}_{intralayer}$ (which discards all interlayer links and only contains intralayer links of the multiplex network). For **BPRMF** and **GCN**, we train a separate model for each behavior.

The embedding size is fixed to 32 and the batch size to 1024 for all methods. Other hyper-parameters of all methods are tuned by grid search on the validation set. As for MGNN, the learning rate of the Adam optimizer is $3e^{-4}$, $\lambda_\Theta = 1e^{-5}$ and $\lambda^\alpha = \frac{1}{m}$. Besides, the number of layers of GCN is 2 for all datasets.

4.2 Performance Comparison

The result are summarized in Table 2. Since NMTR is designed for ordinal behaviors and cannot be applied to generic behaviors, we do not compare it with other methods on Douban Book dataset.

As for the ordinal-behavior datasets (i.e., YooChoose and Steam), the proposed QGNN model marginally outperforms the state-of-the-art NMTR method and significantly outperforms other baselines. As for the generic-behavior dataset Douban Book, compared with the best results of baselines, the proposed MGNN model achieves an improvement of 4.5%~11.4% w.r.t HR@50 and 3.5%~10.2% w.r.t NDCG@50. These results demonstrate that MGNN is able to model

[1] https://www.kaggle.com/chadgostopp/recsys-challenge-2015
[2] https://cseweb.ucsd.edu/~jmcauley/datasets.html
[3] https://github.com/7thsword/MFPR-Datasets/

Table 1: Basic statistics of the three datasets.

Dataset	#user	#item	Dataset	Behavior	#interactions	Dataset	Behavior	#interactions
YooChoose	509,126	19,034	YooChoose	click	2,292,077	Douban Book	wish	162,565
				purchase	1,045,390		reading	71,662
Steam	24,110	8,696	Steam	purchase	2,434,501		read	174,726
				play	1,557,517		tag	162,070
Douban Book	12,850	22,040		review	56,358		comment	151,758
				recommend	51,244		rated	190,590

Table 2: Results of multi-behavior recommendation methods w.r.t HR and NDCG. The best results of baselines are underlined, and the overall best results (based on Wilcoxon signed rank test and $p < 0.05$) are highlighted in bold.

Dataset	YooChoose				Steam							
Behavior	click		purchase		purchase		play		review		recommend	
Metrics	HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG
BPRMF	0.233	0.0966	0.248	0.0963	0.223	0.108	0.243	0.127	0.383	0.196	0.389	0.197
GCN	0.236	0.0983	0.254	0.102	0.235	0.116	0.257	0.134	0.361	0.173	0.370	0.182
CMF	0.241	0.101	0.278	0.125	0.249	0.129	0.267	0.141	0.376	0.178	0.382	0.183
MELL	0.259	0.114	0.309	0.132	0.270	0.137	0.280	0.139	0.394	0.210	0.403	0.220
NMTR	0.268	0.118	0.319	0.140	0.280	0.139	0.296	0.150	0.405	0.221	0.419	0.226
MGNN$_{specific}$	0.238	0.101	0.290	0.129	0.252	0.130	0.250	0.132	0.399	0.222	0.406	0.226
MGNN$_{intralayer}$	0.276	0.128	0.317	0.153	0.298	0.152	0.307	0.159	0.377	0.206	0.390	0.216
MGNN	**0.281**	**0.130**	**0.332**	**0.161**	**0.302**	**0.155**	**0.315**	**0.163**	**0.416**	**0.233**	**0.433**	**0.237**

Dataset	Douban Book											
Behavior	wish		reading		read		tag		comment		rated	
Metrics	HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG
BPRMF	0.0705	0.0203	0.121	0.0370	0.131	0.0403	0.116	0.0373	0.142	0.0430	0.128	0.0384
GCN	0.0693	0.0197	0.125	0.0374	0.129	0.0415	0.117	0.0377	0.145	0.0441	0.132	0.0406
CMF	0.0721	0.0214	0.132	0.0401	0.140	0.0446	0.126	0.0388	0.150	0.0464	0.140	0.0425
MELL	0.0765	0.0223	0.140	0.0422	0.143	0.0450	0.123	0.0375	0.155	0.0490	0.142	0.0438
MGNN$_{specific}$	0.0723	0.0210	0.127	0.0386	0.137	0.0423	0.120	0.0383	0.145	0.0436	0.142	0.0430
MGNN$_{intralayer}$	0.0789	0.0225	0.138	0.0411	0.149	0.0473	0.136	0.0406	0.156	0.0486	0.145	0.0444
MGNN	**0.0852**	**0.0234**	**0.155**	**0.0465**	**0.153**	**0.0493**	**0.140**	**0.0418**	**0.162**	**0.0507**	**0.154**	**0.0478**

the collective effect of multiple behaviors. Besides, although MELL is not originally designed for recommendation, it perform slightly better than traditional recommendation methods on this dataset, which demonstrates that multiplex network is an effective representation of the multi-behavior data.

4.3 Ablation Analysis

We further compare the results of MGNN and its variants to verify the effectiveness of the components in our proposed MGNN model. Firstly, compared with MGNN, the MGNN$_{specific}$ model performs much worse on all three datasets, which shows the effectiveness of the shared embeddings in MGNN. Secondly, by comparing the results of MGNN and MGNN$_{intralayer}$, we observe that these two methods perform closely, except for the *review* and *recommend* behavior of the Steam dataset. We believe the main reason is that the interactions of these two behaviors are quite sparse, and therefore without the interlayer links, information can not be transfered across different layers and MGNN$_{intralayer}$ has no enough information to learn good representation for nodes in these behaviors.

5 CONCLUSION

In this paper, we propose a general model named Multiplex Graph Neural Networks (MGNN) for multi-behavior recommendation problem. MGNN tackles the problem from the perspective of multiplex network. Extensive experiments on three dataset show that MGNN is able to model the collective effect of multiple behaviors and outperforms several existing multi-behavior recommendation methods. For future work, we plan to further investigate multi-behavior recommendation problem with explicitly negative-feedback behaviors (e.g. skip behaviors in music recommendation).

6 ACKNOWLEDGMENT

We thank the support of National Natural Science Foundation of China No. 61672449.

REFERENCES

[1] Chen Gao, Xiangnan He, Dahua Gan, Xiangning Chen, Fuli Feng, Yong Li, Tat-Seng Chua, and Depeng Jin. 2019. Neural Multi-task Recommendation from Multi-behavior Data. In *35th IEEE International Conference on Data Engineering, ICDE*. 1554–1557.

[2] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR*.

[3] Babak Loni, Roberto Pagano, Martha A. Larson, and Alan Hanjalic. 2016. Bayesian Personalized Ranking with Multi-Channel User Feedback. In *Proceedings of the 10th ACM Conference on Recommender Systems*. 361–364.

[4] Ryuta Matsuno and Tsuyoshi Murata. 2018. MELL: Effective Embedding Method for Multiplex Networks. In *Companion of the The Web Conference 2018 on The Web Conference 2018, WWW*. 1261–1268.

[5] Chanyoung Park, Donghyun Kim, Jiawei Han, and Hwanjo Yu. 2019. Unsupervised Attributed Multiplex Network Embedding. *CoRR* abs/1911.06750 (2019).

[6] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI 2009*. 452–461.

[7] Ajit Paul Singh and Geoffrey J. Gordon. 2008. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 650–658.

[8] Mengting Wan and Julian J. McAuley. 2018. Item recommendation on monotonic behavior chains. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 86–94.

[9] Linchuan Xu, Xiaokai Wei, Jiannong Cao, and Philip S. Yu. 2019. Multi-task network embedding. *Int. J. Data Sci. Anal.* 8, 2 (2019), 183–198.