

# DBRec: Dual-Bridging Recommendation via Discovering Latent Groups

Jingwei Ma  
The University of Queensland  
jingwei.ma@uq.edu.au

Jiahui Wen\*  
National University of Defense  
Technology  
wen\_jiahui@outlook.com

Mingyang Zhong  
Central Queensland University;  
Chongqing Meiqi Industry Co.  
m.zhong@cqu.edu.au

Liangchen Liu  
The University of Queensland  
l.liu9@uq.edu.au

Chaojie Li  
Aliexpress, Alibaba group  
Hang Zhou, Zhe Jiang  
Cjlee.cqu@163.com

Weitong Chen  
The University of Queensland  
w.chen9@uq.edu.au

Yin Yang  
Hamad Bin Khalifa University  
yyang@hbku.edu.qa

Hongkui Tu  
National University of Defense  
Technology  
tuhkjet@foxmail.com

Xue Li  
The University of Queensland  
xueli@itee.uq.edu.au

## ABSTRACT

In recommender systems, the user-item interaction data is usually sparse and not sufficient for learning comprehensive user/item representations for recommendation. To address this problem, we propose a novel dual-bridging recommendation model (DBRec). DBRec performs latent user/item group discovery simultaneously with collaborative filtering, and interacts group information with users/items for bridging similar users/items. Therefore, a user's preference over an unobserved item, in DBRec, can be bridged by the users within the same group who have rated the item, or the user-rated items that share the same group with the unobserved item. In addition, we propose to jointly learn user-user group (item-item group) hierarchies, so that we can effectively discover latent groups and learn compact user/item representations. We jointly integrate collaborative filtering, latent group discovering and hierarchical modelling into a unified framework, so that all the model parameters can be learned toward the optimization of the objective function. We validate the effectiveness of the proposed model with two real datasets, and demonstrate its advantage over the state-of-the-art recommendation models with extensive experiments.

## CCS CONCEPTS

• **Computer systems organization** → **Embedded systems; Redundancy; Robotics**; • **Networks** → **Network reliability**.

\*Corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

CIKM '19, November 3–7, 2019, Beijing, China

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-6976-3/19/11...\$15.00

<https://doi.org/10.1145/3357384.3357892>

## KEYWORDS

Recommendation, Dual-Bridging, Latent group discovery

## ACM Reference Format:

Jingwei Ma, Jiahui Wen, Mingyang Zhong, Liangchen Liu, Chaojie Li, Weitong Chen, Yin Yang, Hongkui Tu, and Xue Li. 2019. DBRec: Dual-Bridging Recommendation via Discovering Latent Groups. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM'19)*, November 3–7, 2019, Beijing, China. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3357384.3357892>

## 1 INTRODUCTION

Due to the prevalence of the Internet, recommender systems have attracted remarkable attentions in industrial and research communities. Recommender systems mainly aim to alleviate the problem of information overload for online services, and explore a tremendous amount of information and recommend interested items for each user. Collaborative Filtering (CF) is one of the most popular recommendation techniques, which leverages users' historical behaviors to collaboratively infer user preferences over items. Among the CF-based methods, Matrix Factorization (MF) is efficient in capturing user preferences and providing superior recommendation performance. The basic idea behinds matrix factorization is to embed each user/item into a low-dimensional vector, and model a user-item rating with the interaction (i.e. inner product) between the corresponding user/item vector.

Many previous works have developed MF-based methods for modeling user-item interactions. For example, neural matrix factorization [10] combines generalize matrix factorization and multiple-layer perceptron for learning the interaction function. The multiple-layer of nonlinear transformations is proved to be efficient in learning user-item interactions, and yields state-of-the-art recommendation performance. ConvNCF [9] utilizes outer product to transform each user-item embedding pair into a two-dimensional interaction map, and employs convolution and pooling layers for modeling high-order user-item correlations. [3] involves context items (e.g. items rated by users) for modeling user-item interactions,

hence user preferences over items are comprised of user-item and item-item similarities. Eventhough the aforementioned methods are shown to be effective, they suffer from the problem of data sparseness, as they are mainly based on user-item interactions, and users usually give few ratings to the items in real scenarios.

To address the data sparseness problem, existing research propose to incorporate additional information for boosting recommendation performance. Some of them [28, 29] leverage social networks for bridging similar users, and propagating user preferences along the social links. Some others resort to attribute data of different modalities to mitigate data sparseness. Typical attribute data include user profile [20], descriptive texts [4–6], images [18, 19], etc. However, these recommender models are tightly coupled with one information source or another, and the requirement of extra information sources limits their scalability.

In this work, we propose a novel method for boosting recommendation without the requirement of auxiliary data sources. To this end, we discover latent user/item groups and model user-item interactions collaboratively. The discovered latent groups are then interacted with users/items to boost recommendation performance. The rationale underlying DBRec is that, the estimation of a user’s rating over an unobserved item, in the proposed model, can be bridged by the users within the same group who have interacted with the item, or the user-rated items that share the same group with the unobserved item. The in-group users and items are abstracted with high-level group representations, and the interactions of user-item group and item-user group result in a dual-bridging architecture. In this light, the dual-bridging mechanism can mitigate data sparseness through bridging similar users/items. Also, we utilize deep neural networks for modeling user-user group (item-item group) hierarchies, so that users/items are forced to be close to their groups for learning compact representations.

Notice that, in [21], it exploits group information to boost social recommendation. However, the group information is explicitly required, hence those methods cannot be scaled to the scenarios where the group information is not explicitly available.

The contributions of this work can be concluded as follows,

- We propose to seamlessly combine collaborative filtering and latent group discovery for recommendation. The discovered group information can dual-bridge similar users and items for boosting recommendation performance.
- We propose to jointly model user-user group (item-item group) hierarchies for learning compact user/item representations.
- We validate the effectiveness of the proposed model on two real datasets, and demonstrate its advantage over the state-of-the-art recommendation models with comparison experiments and comprehensive analysis.

## 2 RELATED WORK

In this section, we review the representative state-of-the-art CF-based recommender models.

### 2.1 Neural Recommendation

In recently years, deep learning techniques have been widely applied to recommender community. The majority of works employ multiple-layer neural networks to model deep user-item interactions. For example, NeuMF [10] combines generalized matrix factorization and multiple-layer perceptron for modeling user-item similarities. ConvMF [9] proposes to use outer product to transform embeddings of each user-item pair into a two-dimensional interaction map, and then employs convolutional and pooling layer to model high-order interrelations among embedding dimensions. Cheng et al.[3] leverage item contexts (i.e. historical items rated by users) to compensate the interaction function. Ebesu et al. [7] propose an attention to aggregate neighboring users, and jointly exploit the neighborhoods with user-item interactions to derive the recommendation. However, one major shortcoming of those models is that, they are mainly based on the interaction data, and suffer from data sparseness.

### 2.2 Recommendation with Side Information

To alleviate data sparseness and boost recommendation performance, existing works incorporate side information of different modalities for recommendation. The auxiliary information reflects item characteristics, and can bridge similar items (users) for alleviating data sparseness. In most cases, they employ deep neural networks to extract high-level representations for the extra information, and populate them into CF-based methods for recommendation. The extra data sources that are commonly exploited include user profile [20], texts [4], images [18] and demands [14, 15]. However, the aforementioned methods are usually coupled with one type of information source or another, and lack the scalability when they are deployed for recommendation in different scenarios. Notice that group information has been exploited for group recommendation. For example, in [21], the authors address group recommendation problem, and aggregate representations of group members with an attention mechanism to obtain group representation. However, the group information needs to be specified in the data, and hence those methods are inapplicable in our case.

In [7], the authors combine latent factor model and neighborhood-based structure, and propose an attention mechanism to find similar users based on the specific user and item. However, they simply aggregate similar users for estimating the rating scores, therefore the neighboring users are not sufficiently leveraged for bridging unobserved user-item pairs. Chen et al. [2] aggregate social friends with an attention mechanism given the specific items, and interact the aggregated neighborhoods with the item to derive the recommendation. However, the interactions between the neighborhoods and the target items exert too strict regularization between the users and their social friends.

### 2.3 Community Detection

The proposed latent group discovery is related to community detection [1] that identifies groups of densely connected nodes in a graph. However, those methods [13] explicitly require proximity information among the nodes for discovering latent communities, and they are inapplicable in the case when the link information is not available.

**Table 1: Descriptions of the symbols used in this paper**

Symbols	Descriptions
$m, n$	number of users and items
$u_i, v_j$	$i$ -th user and $j$ -th item
$U, V$	set of users and items
$\mathbf{R}$	rating matrix
$r_{ij}$	rating score of $u_i$ over $v_j$
$\hat{r}_{ij}$	estimated rating score of $u_i$ over $v_j$
$\mathbf{u}_i, \mathbf{v}_j$	embedding of $u_i$ and $v_j$
$d$	dimension of user and item embeddings
$G_u, G_v$	set of user and item groups
$\mathbf{G}_u, \mathbf{G}_v$	embedding matrix of user and item groups
$\mathbf{g}_s^u$	embedding of $s$ -th user group
$\mathbf{g}_t^v$	embedding of $t$ -th item group
$a_i$	group label of user $u_i$
$b_j$	group label of item $v_j$
$d_g$	dimension of user and item group embeddings
$\mu_i$	soft latent group representation of $u_i$
$\mathbf{u}'_i$	reconstruction of $\mathbf{u}_i$
$\mathbf{v}'_j$	reconstruction of $\mathbf{v}_j$
$\{\mathbf{u}'_s   s = 1, \dots, p\}$	$p$ negative samples for $\mathbf{u}'_i$ in learning user groups
$\{\mathbf{v}'_t   t = 1, \dots, p\}$	$p$ negative samples for $\mathbf{v}'_j$ in learning item groups
$\phi_{\{uv, ug, vg, u, v\}}^{\{1, \dots, L\}}$	$L$ layers neural network for modeling user-item, user-group, item-group, user and item hierarchies

By contrast, we propose to discover latent user/item groups and model user-item interactions collaboratively, the proposed latent group discovery and collaborative filtering can be mutually beneficial to each other. On one hand, users/items with similar rating activities are more likely to have similar representations in collaborative filtering, which facilitates the user/item clustering process. On the other hand, user/item groups can bridge similar users/items and boost recommendation performance.

## 3 THE PROPOSED MODEL

### 3.1 Preliminaries

In a recommendation problem, we have a set of  $m$  user  $U = \{u_1, \dots, u_m\}$ , a set of  $n$  items  $V = \{v_1, \dots, v_n\}$  and a rating matrix  $\mathbf{R} \in \{0, 1\}^{m \times n}$ , where each element  $r_{ij} \in \{0, 1\}$  equals 1 if user  $u_i$  has an observed interaction (e.g. clicked, viewed, etc.) with item  $v_j$  and 0 otherwise. The task of item recommendation is to estimate the rating score  $\hat{r}_{ij}$  for unobserved user-item interactions, and recommend for each user the items with high estimated rating scores. In this paper, we also propose to discover latent user groups  $G_u = \{1, \dots, k\}$  and item groups  $G_v = \{1, \dots, k\}$  for bridging similar users and items, where  $k$  is the pre-specified number of groups for users and items.

In this paper, two-dimension matrices are denoted with bold uppercase symbols, single-dimension vectors are denoted bold lowercase symbols, and scalars are represented with lowercase symbols. The symbols used in this paper are summarized in Table.1, other intermediate symbols are described in the corresponding sections.

### 3.2 Basic model

Inspired by previous works [10, 23], we employ multiple neural networks as the basic model for capturing deep user-item interactions. Without loss of generality, given the embedding vectors  $\mathbf{u}_i \in \mathbb{R}^d, \mathbf{v}_j \in \mathbb{R}^d$  for user  $u_i$  and item  $v_j$  respectively, where  $d$  is the number of dimensions of user and item embedding spaces. The corresponding rating score of  $u_i$  over  $v_j$  can be estimated as:

$$\begin{aligned} \mathbf{z}_0 &= [\mathbf{u}_i; \mathbf{v}_j; \mathbf{u}_i \circ \mathbf{v}_j] \\ \mathbf{z}_{ij} &= \phi_L(\dots \phi_1(\mathbf{z}_0) \dots) \\ \phi_l(\mathbf{z}_{l-1}) &= \sigma_l(\mathbf{W}_l \mathbf{z}_{l-1} + \mathbf{b}_l), l \in [1, L] \\ \hat{r}_{ij} &= \frac{1}{1 + \exp(-\mathbf{w}_{uv}^T \mathbf{z}_{ij})} \end{aligned} \quad (1)$$

where  $[\cdot]$  and  $\circ$  are the concatenation and element-wise multiply operation respectively.  $L$  is the number of hidden layers in the neural network, and  $\mathbf{W}_l, \mathbf{b}_l, \sigma_l$  are the weight matrix, bias vector and non-linear activation function of the  $l$ -th layer, respectively.  $\mathbf{w}_{uv}$  is the parameter vector for transforming high-level user-item interaction vector  $\mathbf{z}_{ij}$  into a logit, which is then used for rating score estimation.

### 3.3 Latent group discovery

In many online services, some users (items) have the same interests (characteristics) and constitute a group. Users/items in a group share the same group representation, and it can bridge similar users/items and mitigate data sparseness. However, in most cases, the group information is not explicitly available. In this paper, we propose to discover latent user/item groups and model user-item interactions collaboratively. Specifically, we propose to learn user and item group embeddings, and introduce embedding matrix  $\mathbf{G}_u \in \mathbb{R}^{k \times d_g}$  and  $\mathbf{G}_v \in \mathbb{R}^{k \times d_g}$  for user and item group respectively, where  $k$  is the number of user/item group as specified previously, and it is much smaller than  $m$  and  $n$ .  $d_g$  is the dimension of user/item group embeddings, and it is assumed to be smaller than the user/item dimension  $d$  as group embeddings represent users/items at a more general level of granularity.

For learning user group embeddings, as an example, each user embedding is input into a non-linear transformation to obtain a vector that has the dimension of  $k$ , which is equal to the number of user groups. The value of each dimension of the vector can be viewed as the weight that the user belongs to the corresponding group, and the weighted sum of the user group embedding matrix is regarded as the soft latent group representation of the user. The user embedding is then reconstructed from the soft latent group representation. Therefore, user group embeddings are learned by transforming user embeddings into their reconstructions with least possible of information loss, and preserving most of the information in the  $k$  embedded user groups. In addition, users with similar representations are supposed to have similar soft latent group representations, and are clustered automatically into the same user group. The user group learning process is illustrated in Fig.1, and we detail the learning process in the following subsections.

**3.3.1 Soft latent group representation.** Each user embedding  $\mathbf{u}_i$  is fed into a non-linear transformation to obtain a vector  $\beta_i \in \mathbb{R}^k$ , as shown in the following equation.

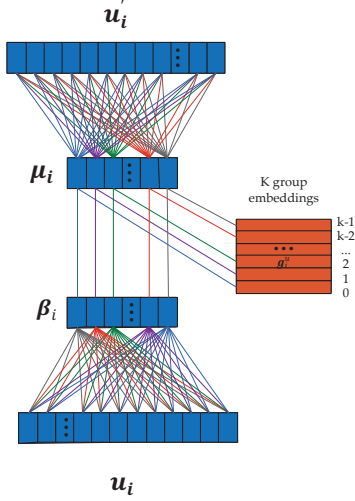


Figure 1: Illustration of user group learning process.

$$\beta_i = \text{softmax}(\mathbf{W}_u \mathbf{u}_i + \mathbf{b}_u) \quad (2)$$

where  $\mathbf{W}_u \in \mathbb{R}^{k \times d}$  is the weighted matrix parameter, and  $\mathbf{b}_u \in \mathbb{R}^k$  is the bias vector.  $\beta_i$  can be viewed as the activations toward the user group embeddings, and each dimension of  $\beta_i$  is the probability that the user belongs to the related group. The weighted sum of the user group embeddings with respect to  $\beta_i$  is then regarded as the soft latent group representation of user  $u_i$ .

$$\mu_i = \sum_{s=1}^k \beta_{i,s} \mathbf{g}_s^u \quad (3)$$

where  $\beta_{i,s}$  is the  $s$ -th dimension of vector  $\beta_i$ , and  $\mathbf{g}_s^u$  is the  $s$ -th row of embedding matrix  $\mathbf{G}_u$ , representing the embedding of  $s$ -th user group.  $\mu_i \in \mathbb{R}^{d_g}$  is the soft latent group representation of user  $u_i$ . Therefore, users with similar embeddings are more likely to have similar activations toward the group embeddings, and they are automatically clustered into the same group in an unsupervised manner.

**3.3.2 Reconstruction user embeddings.** We have obtained the soft latent group representations for the users. Now we describe how to calculate the reconstruction of the embedding for each user  $u_i$  from his/her soft latent group representations  $\mu_i$ . We reconstruct user embeddings from their related soft latent group representations  $\mu_i$ . The underlying reason is that if two users have similar soft latent group representations, then they are more likely within the same user group and the reconstructions of their embeddings are assume to be similar. In this paper, we employ non-linear transformations to reconstruct user embeddings, as shown follows:

$$\mathbf{u}_i' = \text{sigmoid}(\mathbf{W}_{u'} \mu_i + \mathbf{b}_{u'}) \quad (4)$$

where  $\mathbf{W}_{u'} \in \mathbb{R}^{d \times d_g}$  is the weighted matrix parameter, and  $\mathbf{b}_{u'} \in \mathbb{R}^d$  is the bias vector. At this point, the reconstruction of user embeddings is similar to an autoencoder, and obtaining the soft latent group representations can be viewed as the encoder stage of an

autoencoder that takes user embeddings as input and maps them to the latent space while preserving group information. Eq.(4) is similar to the decoder stage of an autoencoder that maps the soft latent group representation to the reconstruction  $\mathbf{u}_i'$  of the same shape as  $\mathbf{u}_i$ , as illustrated in Fig.1.

**3.3.3 Learning objective.** Learning user groups is equivalent to minimizing the reconstruction error of  $\mathbf{u}_i'$  from  $\mathbf{u}_i$ . We employ the contrastive max-margin objective function that is commonly used in previous work [12, 27]. Specifically, for each user embedding  $\mathbf{u}_i$ , we randomly sample  $p$  users as negative users. The embeddings of the negative samples are represented as  $\{\mathbf{u}_{i_s} | s = 1, 2, \dots, p\}$ , and the learning objective is to make the reconstructed embedding  $\mathbf{u}_i'$  similar to its origin embedding  $\mathbf{u}_i$  while different from those negative embeddings. Therefore, the objective of user group learning  $\mathcal{L}_u$  is defined as a hinge loss that maximize the cosine similarity between  $\mathbf{u}_i'$  and  $\mathbf{u}_i$  and simultaneously minimize that between  $\mathbf{u}_i'$  and the negative samples:

$$\mathcal{L}_u = \sum_{u_i} \sum_{s=1}^p \max(0, 1 - \mathbf{u}_i'^T \mathbf{u}_i + \mathbf{u}_i'^T \mathbf{u}_{i_s}) \quad (5)$$

Similarly, the objective of item group learning can be formulated as:

$$\mathcal{L}_v = \sum_{v_j} \sum_{t=1}^p \max(0, 1 - \mathbf{v}_j'^T \mathbf{v}_j + \mathbf{v}_j'^T \mathbf{v}_{j_t}) \quad (6)$$

where  $\mathbf{v}_j'$  is the reconstructed embedding from  $\mathbf{v}_j$ . The calculation of  $\mathbf{v}_j'$  is similar to that of  $\mathbf{u}_i'$  and we leave out the detail description here to avoid redundancy.  $\{\mathbf{v}_{j_t} | t = 1, 2, \dots, p\}$  are  $p$  negative samples for  $\mathbf{v}_j'$  when computing objective  $\mathcal{L}_v$ .

## 3.4 Hierarchies modeling

The intuition of hierarchy modeling is that users/items should be close to their groups to learn compact representations. To this end, take user-user group hierarchy for example, we introduce a latent variable  $\epsilon_i$  for each user  $u_i$ , and minimize the distance between  $\mathbf{u}_i + \epsilon_i$  and its group representation  $\mu_i$  in the embedding space [8]. In this paper, we propose to utilize deep neural networks to model the user-user group (item-item group) hierarchies.

We present the user-user group hierarchy modeling in Fig.2. As illustrated in the figure, we feed the vector  $\mathbf{u}_i + \epsilon_i$  into a multiple-layer neural network, and obtain a high-level representation  $\mathbf{z}^{u_i}$  that has the same dimensions as the group representations. The similarities between  $\mathbf{z}^{u_i}$  and all the group representations are measured with the inner product operation, and the resulted logits are transformed into a posterior probability vector with a softmax function, as shown in Eq.(7).

$$\begin{aligned} \mathbf{z}_0^{u_i} &= \mathbf{u}_i + \epsilon_i \\ \mathbf{z}^{u_i} &= \phi_L^u(\dots \phi_1^u(\mathbf{z}_0^{u_i}) \dots) \\ a_i &= \text{argmax}_s(\beta_{i,s}), \quad s = 1, 2, \dots, k \\ \hat{r}^{u_i} &= \frac{\exp(\mathbf{g}_{a_i}^{u_i T} \mathbf{z}^{u_i})}{\sum_{s=1}^k \exp(\mathbf{g}_s^{u_i T} \mathbf{z}^{u_i})} \end{aligned} \quad (7)$$

where  $\beta_{i,s}$  is described in Eq.(3),  $a_i$  is the label of the user group that user  $u_i$  has the maximum activation, and  $\mathbf{g}_{a_i}^{u_i}$  is the corresponding

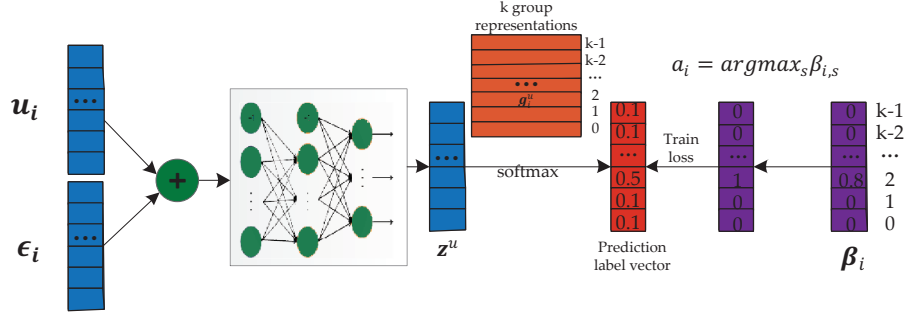


Figure 2: Hierarchy modeling of user and user group.

group embedding, hence  $\mathbf{g}_{a_i}^u$  can be viewed as the hard latent group representation of  $u_i$ .  $\hat{r}^{u_i}$  is the posterior probability that user  $u_i$  belongs to group  $a_i$ , and therefore bridging the gap between a user and his/her group is equivalent to maximizing the corresponding posterior probability. In other words, the neural network takes a user and his/her group representations as input, and produces a predicted distribution over the groups. The training objective is defined as the minimization between the predicted distribution and the distribution of actual labels [25], and it is demonstrated to be better than other loss functions [26]. The objective function of user-user group hierarchy can be defined as follows:

$$\mathcal{L}_{uu} = - \sum_{u_i} \log \frac{\exp(\mathbf{g}_{a_i}^u T \mathbf{z}^{u_i})}{\sum_{s=1}^k \exp(\mathbf{g}_s^u T \mathbf{z}^{u_i})} \quad (8)$$

Similarly, the objective function of item-item group hierarchy can be defined as:

$$\mathcal{L}_{vv} = - \sum_{v_j} \log \frac{\exp(\mathbf{g}_{b_j}^v T \mathbf{z}^{v_j})}{\sum_{t=1}^k \exp(\mathbf{g}_t^v T \mathbf{z}^{v_j})} \quad (9)$$

where  $b_j$  is group label of item  $v_j$ , and it is calculated in the same way as  $a_i$ .  $\mathbf{g}_{b_j}^v$  is the corresponding group embedding of  $v_j$ . The neural network for modeling item-item group hierarchy has the same architecture as that in Fig.2, and  $\mathbf{z}^{v_j}$  can be viewed as the high-level representation of item  $v_j$  extracted from the neural network.

### 3.5 Dual-bridging collaborative filtering

Once we have obtained group representations, we can incorporate them for boosting recommendation performance. The underlying reason is that, the user-item group (item-user group) interactions can bridge similar users/items and alleviate data sparseness. For example, even though a user's interest in an item is unknown, the item can still be ranked higher in the recommendation list, as long as the user has positively interacted with some other items sharing the same group with the unobserved item, or users within the same group have positively rated the unobserved item. Notice that, the collaborative filtering and latent group discovery are iteratively performed, and they mutually benefit each other for better learning representations. First, collaborative filtering learns similar user/item representations based on the user-item interaction data, and this facilitates the clustering process for discovering users/items with similar representations. In turn, the latent group

information can bridge similar users (items), and boost recommendation performance.

The general architecture of the proposed model is presented in Fig.3. As shown in the figure we introduce three neural networks for modeling user-item, user-item group, user group-item interactions respectively. The rating score of a user  $u_i$  over an item  $v_j$  is predicted as follows,

$$\begin{aligned} \mathbf{z}_0^{uv} &= [\mathbf{u}_i; \mathbf{v}_j; \mathbf{u}_i \circ \mathbf{v}_j] \\ \mathbf{z}_{ij}^{uv} &= \phi_L^{uv}(\dots \phi_1^{uv}(\mathbf{z}_0^{uv}) \dots) \end{aligned} \quad (10)$$

$$\begin{aligned} \mathbf{z}_0^{ug} &= [\mathbf{u}_i; \mathbf{g}_{b_j}^v; \mathbf{u}_i^T \mathbf{M}_u \mathbf{g}_{b_j}^v] \\ \mathbf{z}_{ij}^{ug} &= \phi_L^{ug}(\dots \phi_1^{ug}(\mathbf{z}_0^{ug}) \dots) \end{aligned} \quad (11)$$

$$\begin{aligned} \mathbf{z}_0^{vg} &= [\mathbf{g}_{a_i}^u; \mathbf{v}_j; \mathbf{v}_j^T \mathbf{M}_v \mathbf{g}_{a_i}^u] \\ \mathbf{z}_{ij}^{vg} &= \phi_L^{vg}(\dots \phi_1^{vg}(\mathbf{z}_0^{vg}) \dots) \end{aligned} \quad (12)$$

$$\hat{r}_{ij} = \frac{1}{1 + \exp[-(\mathbf{w}_{uv}^T \mathbf{z}_{ij}^{uv} + \mathbf{w}_{ug}^T \mathbf{z}_{ij}^{ug} + \mathbf{w}_{vg}^T \mathbf{z}_{ij}^{vg})]} \quad (13)$$

where  $\mathbf{M}_u \in \mathbb{R}^{d \times d_g}$  is a transformation matrix for measuring the similarity between  $\mathbf{u}$  and  $\mathbf{g}_{b_j}^v$  in a common latent space, while  $\mathbf{M}_v \in \mathbb{R}^{d \times d_g}$  is a matrix for measuring the similarity between  $\mathbf{v}_j$  and  $\mathbf{g}_{a_i}^u$ .  $\phi_{\{1, \dots, L\}}^{\{uv, ug, vg\}}$  are the respective neural networks for modeling user-item, user-item group, item-user group interaction, and  $\mathbf{w}_{\{uv, ug, vg\}}$  are the parameter vectors for transforming high-level representations (i.e.  $\mathbf{z}_{ij}^{uv}, \mathbf{z}_{ij}^{ug}, \mathbf{z}_{ij}^{vg}$ ) of different interaction spaces into the final predictive logit. Therefore, with the dual-bridging architecture, an item can be ranked higher in the recommendation list for a user, as long as the user's preference over the item is properly modeled in any one of the three interaction spaces. The training objective of the proposed dual-bridging collaborative filtering can be defined as follows,

$$\mathcal{L}_{uv} = - \sum_{(u_i, v_j) \in \mathcal{D}} r_{ij} \log \hat{r}_{ij} + (1 - r_{ij}) \log(1 - \hat{r}_{ij}) \quad (14)$$

where  $\mathcal{D}$  is the training set.

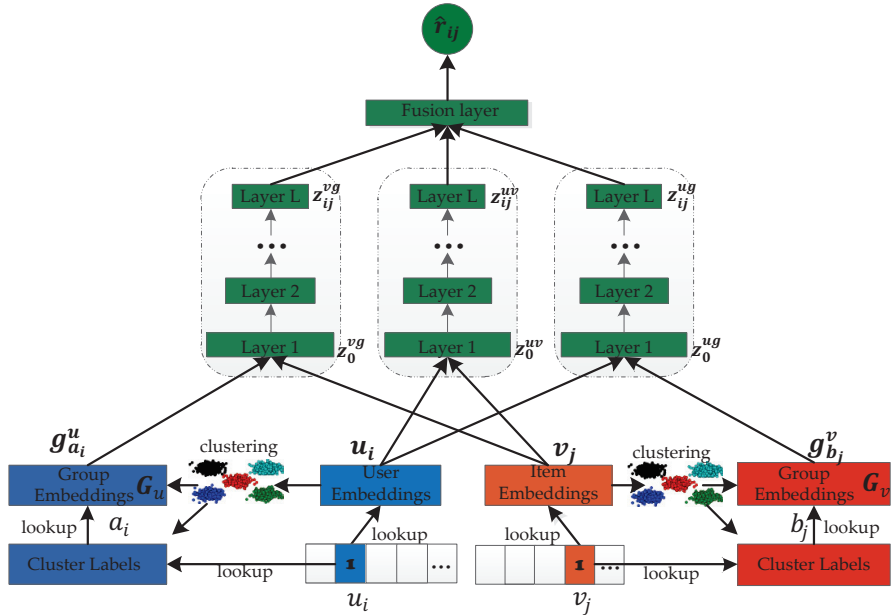


Figure 3: Illustration of the dual-bridging collaborative filtering architecture.

### 3.6 Model learning

Due to the nature of implicit feedback and the large number of items in the recommendation task [11], negative sampling [24] is employed to approximate  $\mathcal{L}_{uv}$ . As for  $\mathcal{L}_{uu}$  and  $\mathcal{L}_{vv}$ , even though the computation involves the summation over the interactions between a user/item and all the groups (i.e.  $\sum_{k=1}^K \exp(\mathbf{g}_k^u \mathbf{z}^{u_i})$ ), the number of groups are much fewer than that of users/items, and hence the summation does not incur high computational overhead. In the proposed DBRec model, we jointly define the loss function, so that it can be optimized efficiently by backpropagation. Taking into account the loss of dual-bridging collaborative filtering, hierarchy modeling and group learning, the final objective function of DBRec can be defined as follows,

$$\mathcal{L} = \mathcal{L}_{uv} + \alpha(\mathcal{L}_{uu} + \mathcal{L}_{vv} + \mathcal{L}_u + \mathcal{L}_v) \quad (15)$$

where  $\alpha$  controls the tradeoff between the loss of dual-bridging collaborative filtering and that of hierarchy modeling and group learning. The training of the proposed model can be decomposed into two parts. In the part of collaborative filtering and hierarchy modeling, user and item embeddings are updated to fit user-item interaction data and preserve group hierarchies. While in the part of group learning, group embeddings are updated to discover representative group information.

The model parameters include user/item representations and parameters of neural networks as shown in Fig.1, Fig.2 and Fig.3. We update the object function with Adam optimizer, which is a variant of Stochastic Gradient Descent with a dynamically tuned learning rate and updates parameters every step along the gradient direction with the following protocol:

$$\theta^t \leftarrow \theta^{t-1} - lr \frac{\partial \mathcal{L}}{\partial \theta} \quad (16)$$

Table 2: Statistics of the datasets.

Dataset	#user	#item	#ratings	Sparsity
Amazon	4016	7199	32190	99.89%
ML1M	6040	3706	1000209	95.53%
Gowalla	78477	528968	5380147	99.98%

where  $lr$  is the learning rate, and  $\theta$  are the model parameters and  $\frac{\partial \mathcal{L}}{\partial \theta}$  are the partial derivatives of the objective function with respect to the model parameters, and they can be automatically computed with typical deep learning libraries.

## 4 EXPERIMENTS

### 4.1 Experimental Settings

**4.1.1 Datasets.** To validate the effectiveness of the proposed model, we conduct experiments on three publicly datasets, namely Amazon<sup>2</sup>, MovieLens-1M (ML1M)<sup>1</sup> and Gowalla<sup>3</sup>. For the amazon dataset, we select the items of musical instruments for evaluation. To transform the explicit ratings of Amazon and ML1M into implicit feedback, ratings higher than 3 are regarded as positive feedback, and the others are regarded as missing values. As for Gowalla, for each user the check-in locations are viewed as positive items, while the other locations are treated as missing items. Inspired by previous works [16, 17], we filter out users with fewer than 5 positive items and the items with fewer than 2 users. The statistics of the datasets is shown in Table.2.

<sup>1</sup><https://grouplens.org/datasets/movielens/>

<sup>2</sup><http://jmcauley.ucsd.edu/data/amazon/>

<sup>3</sup><http://snap.stanford.edu/data/loc-gowalla.html>

4.1.2 *Setup.* To evaluate the proposed model, we randomly split the datasets into training set (70%), validation set (10%) and testing set (20%). In the training process, for each positive item, we randomly sample 5 items as the negative samples. In the testing process, as it is time-consuming to rank all the items for every user at each time, hence for each  $(u_i, v_j)$  pair in the testing set, we mix the testing item with 99 random items, and rank the testing item along with the 99 items for the related user. We measure the recommendation performance with the commonly used Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG), as shown follows,

$$\begin{aligned} HR &= \frac{\#hits}{\#test} \\ NDCG &= \frac{1}{\#test} \sum_{i=1}^{\#test} \frac{1}{\log_2(p_i + 1)} \end{aligned} \quad (17)$$

where  $\#hits$  is the number of testing item that appears in the the recommendation list of the related user and  $\#test$  is the total number of  $(u_i, v_j)$  pair in the testing set.  $p_i$  is the position of the testing item in the recommendation list for the  $i$ -th hit. HR measures whether the testing item is in the recommendation list, while NDCG assigns higher score to the testing item with higher position. In this paper, we truncate the ranking list at  $k \in [1, 2, \dots, 10]$  for both metrics.

4.1.3 *Baselines.* The baselines employed for performance comparison are list as follows,

- NeuMF[10]. It combines generalized MF and Multi-Layer Perception (MLP) for modeling user-item latent structures.
- ConvMF[9]. It uses outer product to transform latent vectors of a user-item pair into two-dimensional map, and applies convolutional and pooling layers to model deep user-item interactions.
- CMN [7]. It identify similar neighboring users with an attention mechanism based on the specific user-item pair, and jointly exploits the neighborhood state and user-item interactions to derive recommendation.
- DELF[3]. It proposes an attention mechanism to aggregate an additional embedding for each user/item, and then further introduce a neural network architecture to incorporate dual embeddings for recommendation.

For fair comparison, baselines in this paper mainly consist of the state-of-the-art recommendation models based on user-item interactions, and the hyper-parameters of these baselines are set according to the original works. Notice that even though CMN exploits neighborhood information for recommendation, the neighboring users are selected mainly based on their rating behaviours. Beside those baselines, we also study various variants of the proposed model, list as follows,

- DBRec-o, a variant of the proposed model that excludes user and item group learning, and it mainly relies on user-item interactions for learning user preferences over the items.
- DBRec-u, a variant of the proposed model that excludes item group learning, and it jointly learns user group and model user-group hierarchy, and interacts user group information with item for boosting recommendation.

- DBRec-i, a variant of the proposed model that excludes user group learning, and it jointly learns item group and model item-group hierarchy, and interacts item group information with users for boosting recommendation.

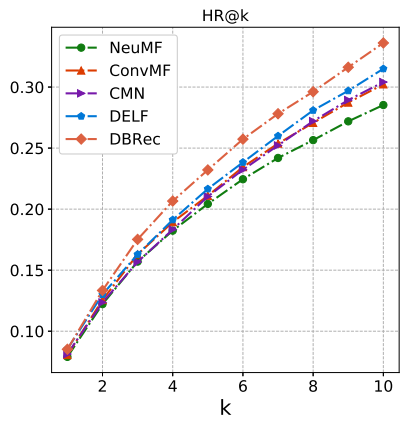
4.1.4 *Implementation.* We implement DBRec based on Tensorflow, and the sources are made publicly available<sup>3</sup> to facilitate community research. We set the batch size to 256 and the learning rate to 0.0001. The dimensions of latent factor is 128. We employ two hidden layers for the neural networks illustrated in Fig.3 and Fig.2, and the hidden units of the respective layer are [64,16] for dual-bridging collaborative filtering, and [64,128] for hierarchy modeling. The model parameters are fine tuned on the validation set. We set the number of user/item group to 5, and the trade-off weight (i.e.  $\alpha$  in Eq.(15)) for group learning and hierarchy modeling to 0.01. We analyze the sensitivity of these two hyper-parameters later in this section.

*pre-training.* The objective function is non-convex, and the learning method can be easily trapped in local optimums. In our work, we pre-train DBRec with NeuMF to obtain initial user and item embeddings in DBRec. As for the group embeddings, we perform clustering on the pre-trained user and item embeddings, and use the cluster centroids to initialize the group embeddings. Since users/items and their corresponding groups do not share the same latent space, we employ a dimensionality reduction method (i.e. t-SNE [22]) to reduce the centroids of  $d$ -dimension to that of  $d_g$  dimensions.

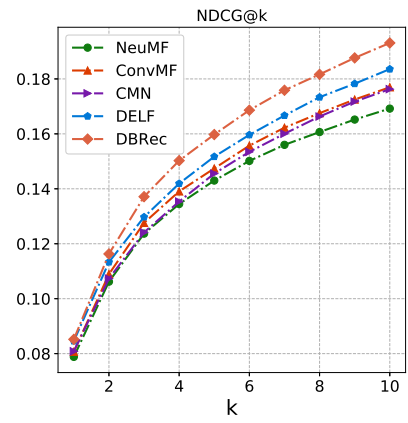
## 4.2 Experiments

4.2.1 *Recommendation comparisons.* We compare the proposed model, DBRec, with the state-of-the-art recommendation models in terms of top-K recommendation, and the results are presented in Fig.4 and Fig.5. From the figures, we have the following observations. First, DELF, ConvMF and CMN outperform NeuMF across different datasets and metrics, which is consistent with previous works [3, 7, 9]. The reason is that NeuMF mainly relies user-item interactions for exploiting user preferences, and it suffers from data sparseness due to the sparsity nature of the rating matrix. Second, the performance differences between CMN and ConvMF vary across different datasets. For example, ConvMF achieves marginally better performance than CMN on ML1M, while CMN outperforms ConvMF on Gowalla. This is because they leverage different aspects of interaction data for boosting performance. Specifically, ConvMF utilizes outer product to exploit dimension-wise correlations for recommendation, while CMN exploits informative neighboring users for bridging each user-item pair. Therefore, the performance differences between these two models are quite data-dependent. Third, the figures show that DELF is superior to CMN, and one possible explanation might be that DELF aggregates for each user/item an additional embedding based on the interacted items/users, and introduce a neural network to incorporate the dual embeddings for recommendation. In addition, DELF can identify the most informative context users/items for collaborative filtering, due to the proposed effective attention mechanism.

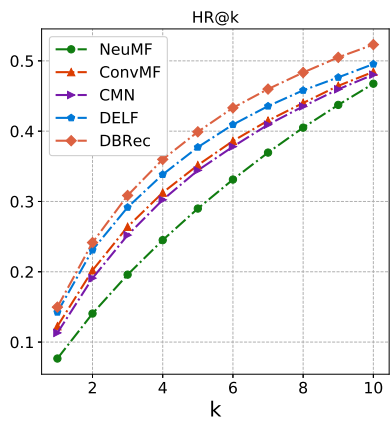
<sup>3</sup><https://github.com/uqjwen/DBRec>



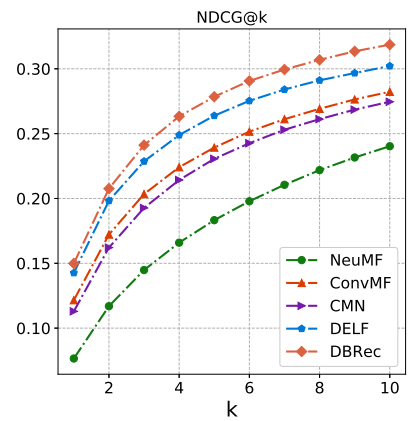
(a) Amazon



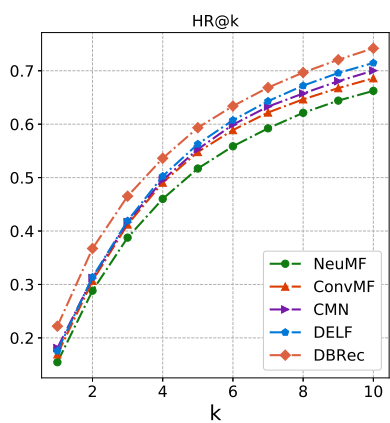
(a) Amazon



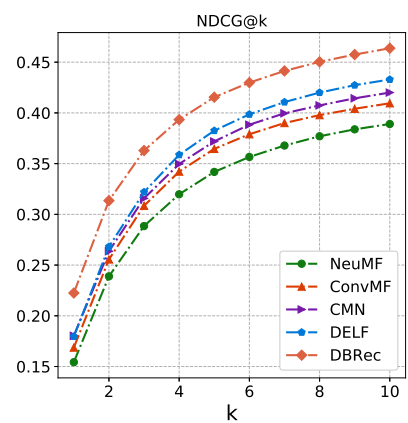
(b) ML1M



(b) ML1M



(c) Gowalla



(c) Gowalla

Figure 4: HR of different models across the datasets

Figure 5: NDCG of different models across the datasets.

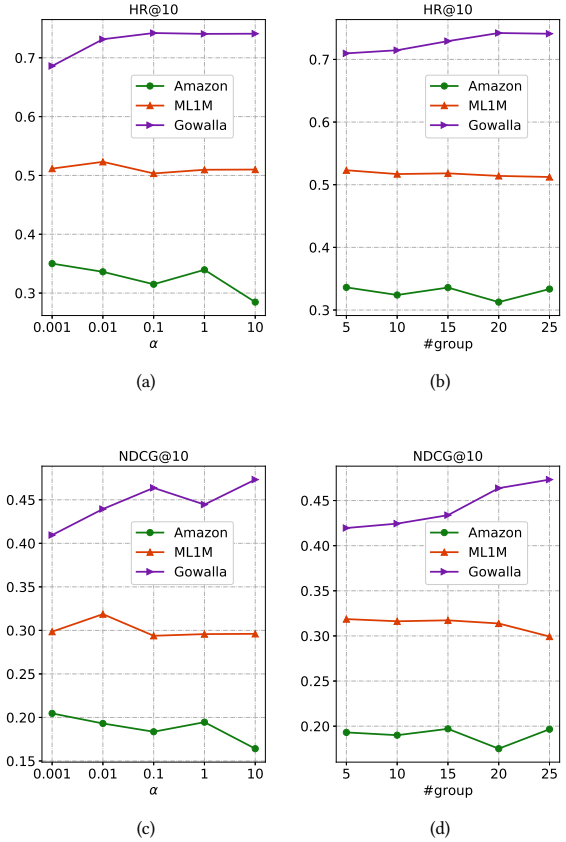


Finally, the proposed model outperforms the baselines by a large margin across the datasets for both HR and NDCG. Specifically, in terms of top-10 recommendation, DBRec outperforms the best baseline, DELF, with a relative improvement of 6.71%(HR@10) and 5.19%(NDCG@10) on Amazon, 5.59%(HR@10) and 5.46%(NDCG@10) on ML1M, and 3.84%(HR@10) and 7.11%(NDCG@10) on Gowalla, respectively. The reason behinds the advantage of DBRec over the baselines is that, we collaboratively discover latent group information for collaborative filtering, and the incorporation of group information can capture user preferences (item characteristics) at different levels of granularity and mitigate data sparseness.

**4.2.2 Efficacy of dual-bridging architecture.** In this subsection, we study the performance of various variants of the proposed model, to validate the effectiveness of the proposed dual-bridging architecture. As shown in Table.3, without latent group information, DBRec degrades to the version (DBRec-o) that is mainly based on user-item interactions, and is vulnerable to data density. However, by discovering latent groups and incorporating group information for collaborative filtering, DBRec-i and DBRec-u significantly outperform DBRec-o, demonstrating the benefit of discovering latent groups for bridging similar users/items.

DBRec combines DBRec-i and DBRec-u, and results in a dual-bridging architecture, which models user preferences and item characteristics at different levels of granularity. General group representations can alleviate the problem of data sparseness by bridging similar users/items, and achieves better recommendation performance than otherwise single-bridging model (e.g. DBRec-i, DBRec-u).

**4.2.3 Sensitivity analysis.** In this subsection, we investigate the robustness of the proposed model with respect to different hyper-parameters. Especially, we study the performance of DBRec under different pre-specified user/item group numbers, and the weight  $\alpha$  (Eq.(15)) for group learning and hierarchy modeling. In this experiment,  $\alpha$  is varied amongst [0.001,0.01,0.1,1,10], and the group number [5,10,15,20,25]. As illustrated in Fig.6, we present the performance in terms of HR@10 and NDCG@10 across the datasets. One can see from the figures that, DBRec experiences a rapid performance degradation with  $\alpha = 10$  and #group = 20 on Amazon. However, the recommendation performance is relatively stable under a wide-range choice of the hyper-parameters. As shown in Fig.6(b) and Fig.6(d), Gowalla presents a steady performance improvement with more group number, probably because larger group number is required to preserve group information due to the size of the dataset. HR@10 and NDCG@10 usually show the similar trends across the datasets except for Gowalla, which experiences a sudden drop on NDCG@10 with  $\alpha = 1$ , but the expected phenomenon is not observed on HR@10 with  $\alpha = 1$ , as shown in Fig.6(a) and Fig.6(c). The reason for the slump of NDCG on Gowalla is that, the testing items are ranked lower at the bottom of the recommendation list. Overall, this experiment demonstrates that the proposed model is invulnerable to the group number. Even though the users/items are clustered into different granularities, the group information can still boost recommendation performance as long as discriminate information provided by the user/item groups can be preserved in the clustering process.



**Figure 6: Recommendation performance of the proposed model with respect to different hyper-parameters.**

**4.2.4 Visualization.** To examine whether DBRec can discover latent groups, we visualize the learned user representations with t-SNE [22], and plot users in the same group with the same color. As shown in Fig.7, several user/item groups can be clearly identified. This experiment demonstrates that DBRec can effectively discover latent groups, and visually interpret the rationale behind the dual-bridging architecture in DBRec.

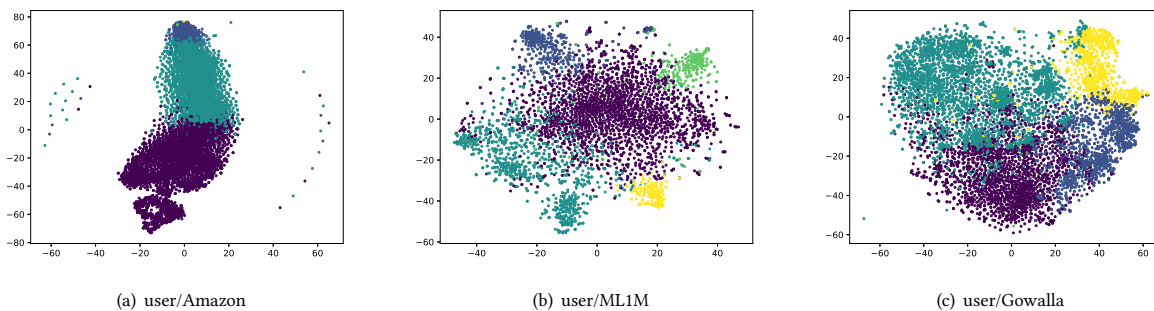
## 5 CONCLUSION

In this paper, we propose a novel dual-bridging recommendation model. The advantage of the proposed model is that it can discover latent group information for bridging similar users/items, and boost recommendation performance. We unified collaborative filtering, group learning and hierarchy modeling into a framework, so that all the sub-components can be jointly optimized and compensate each other for boosting recommendation. Extensive experiments on two real datasets demonstrate the advantage of DBRec in terms of recommendation performance, and robustness in terms of parameters sensitivity.

The proposed model in this paper mainly relies on user-item interaction data, and discover extra latent group information for

**Table 3: Performance of variants of DBRec.**

Datasets	Metrics	DBRec-o	DBRec-i	DBRec-u	DBRec
Amazon	HR@5	0.20429	0.22606	0.213	0.23212
	HR@10	0.28529	0.324	0.31281	0.33613
	NDCG@5	0.143	0.15832	0.14688	0.15975
	NDCG@10	0.1692	0.19001	0.17924	0.19309
ML1M	HR@5	0.28992	0.37152	0.36942	0.39906
	HR@10	0.46752	0.51233	0.50847	0.52311
	NDCG@5	0.18326	0.25381	0.25183	0.27845
	NDCG@10	0.24028	0.29936	0.29685	0.31865
Gowalla	HR@5	0.56284	0.5757	0.58372	0.59342
	HR@10	0.71583	0.72899	0.73358	0.74211
	NDCG@5	0.3825	0.39376	0.39768	0.41547
	NDCG@10	0.43291	0.44354	0.44637	0.46371



**Figure 7: visualizing latent user groups across the datasets**

boosting recommendation performance. In real recommendation scenarios, many attribute data are explicitly available, and integrating heterogeneous modalities of implicit and explicit information for recommendation is an interesting research topic in the future.

## 6 ACKNOWLEDGMENTS

This publication was made possible by Opinion Analysis grant 018493 from Australian Research Council Fund and NPRP grant NPRP10-0208-170408 from the Qatar National Research Fund (a member of Qatar Foundation).

## REFERENCES

- [1] Sandro Cavallari, Vincent W Zheng, Hongyun Cai, Kevin Chen-Chuan Chang, and Erik Cambria. 2017. Learning community embedding with community detection and node embedding on graphs. In *CIKM*.
- [2] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2019. Social Attentional Memory Network: Modeling Aspect- and Friend-level Differences in Recommendation. In *WSDM*.
- [3] Weiyu Cheng, Yanyan Shen, Yanmin Zhu, and Linpeng Huang. 2018. DELF: A Dual-Embedding based Deep Latent Factor Model for Recommendation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. AAAI Press.
- [4] Zhiyong Cheng, Xiaojun Chang, Lei Zhu, Rose C Kanjirathinkal, and Mohan Kankanhalli. 2019. MMALFM: Explainable recommendation by leveraging reviews and images. *ACM Transactions on Information Systems (TOIS)* 37, 2 (2019), 16.
- [5] Zhiyong Cheng, Ying Ding, Lei Zhu, and Mohan Kankanhalli. 2018. Aspect-aware latent factor model: Rating prediction with ratings and reviews. In *Proceedings of the 2018 World Wide Web Conference*. International World Wide Web Conferences Steering Committee, 639–648.
- [6] Zhiyong Cheng and Jialie Shen. 2016. On effective location-aware music recommendation. *ACM Transactions on Information Systems (TOIS)* 34, 2 (2016), 13.
- [7] Travis Ebesu, Bin Shen, and Yi Fang. 2018. Collaborative Memory Network for Recommendation Systems. In *SIGIR*.
- [8] Li Gao, Hong Yang, Jia Wu, Chuan Zhou, Weixue Lu, and Yue Hu. 2018. Recommendation with Multi-Source Heterogeneous Information. In *IJCAI*.
- [9] Xiangnan He, Xiaoyu Du, Xiang Wang, Feng Tian, Jinhui Tang, and Tat-Seng Chua. 2018. Outer Product-based Neural Collaborative Filtering. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. AAAI Press.
- [10] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee. ACM, 173–182.
- [11] Guangneng Hu, Yu Zhang, and Qiang Yang. 2018. CoNet: Collaborative Cross Networks for Cross-Domain Recommendation. In *CIKM*.
- [12] M Iyyer, A Guha, S Chaturvedi, J Boyd-Graber, and H. D III. 2016. Feuding families and former friends: Unsupervised learning for dynamic fictional relationship. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.
- [13] Mark Kozdoba and Shie Mannor. 2015. Community detection via measure space embedding. In *NIPS*.
- [14] Chaojie Li, Chen Liu, Ke Deng, Xinghuo Yu, and Tingwen Huang. 2017. Data-driven charging strategy of PEVs under transformer aging risk. *IEEE Transactions on Control Systems Technology* 26, 4 (2017), 1386–1399.
- [15] Chaojie Li, Chen Liu, Xinghuo Yu, Ke Deng, Tingwen Huang, and Liangchen Liu. 2018. Integrating demand response and renewable energy in wholesale market. In *IJCAI International Joint Conference on Artificial Intelligence*, Vol. 2018. International Joint Conferences on Artificial Intelligence, 382–388.
- [16] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. 2014. GeoMF: joint geographical modeling and matrix factorization for

- point-of-interest recommendation.. In *SIGKDD*.
- [17] Dawen Liang, Laurent Charlin, James Mcinerney, and David M. Blei. 2016. Modeling user exposure in recommendation.. In *WWW*.
  - [18] Liangchen Liu, Feiping Nie, Arnold Wiliem, Zhihui Li, Teng Zhang, and Brian C Lovell. 2018. Multi-modal joint clustering with application for unsupervised attribute discovery. *IEEE Transactions on Image Processing* 27, 9 (2018), 4345–4356.
  - [19] Liangchen Liu, Arnold Wiliem, Shaokang Chen, and Brian C Lovell. 2017. What is the best way for extracting meaningful attributes from pictures? *Pattern Recognition* 64 (2017), 314–326.
  - [20] Jingwei Ma, Guang Li, Mingyang Zhong, Xin Zhao, Lei Zhu, and Xue Li. 2018. LGA: latent genre aware micro-video recommendation on social media. *Multimedia Tools and Applications* 77, 3 (2018), 2991–3008.
  - [21] Jingwei Ma, Jiahui Wen, Mingyang Zhong, Weitong Chen, Xiaofang Zhou, and Jadwiga Indulska. 2019. Multi-source Multi-net Micro-video Recommendation with Hidden Item Category Discovery. In *International Conference on Database Systems for Advanced Applications*. Springer, 384–400.
  - [22] Laurens van der maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. In *JMLR*.
  - [23] Jarana Manotumruksa, Craig Macdonald, and Iadh Ounis. 2017. A Deep Recurrent Collaborative Filtering Framework for Venue Recommendation. In *CIKM*.
  - [24] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*.
  - [25] Byungkook Oh, Seungmin Seo, and Kyong-Ho Lee. 2018. Knowledge Graph Completion by Context-Aware Convolutional Learning with Multi-Hop Neighborhoods. In *CIKM*.
  - [26] Baoxu Shi and Weninger Tim. 2017. ProjE: Embedding Projection for Knowledge Graph Completion. In *AAAI*.
  - [27] Richard Socher, Andrej Karpathy, Quoc V. Le, Christopher D. Manning, and Andrew Y. Ng. 2014. Grounded Compositional Semantics for Finding and Describing Images with Sentences. *Transactions of the Association for Computational Linguistics* 2 (2014).
  - [28] Xiang Wang, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2017. Item Silk Road: Recommending Items from Information Domains to Social Users. *SIGIR* (2017).
  - [29] Carl Yang, Lanxiao Bai, Chao Zhang, Quan Yuan, and Jiawei Han. 2017. Bridging Collaborative Filtering and Semi-Supervised Learning: A Neural Approach for POI Recommendation. In *KDD*. ACM.