Journal of Machine Learning Research 22 (2021) 1-54 Submitted 10/20; Revised 8/21; Published 8/21

Interpretable Deep Generative Recommendation Models

Huafeng Liu HUAFENG@BJTU.EDU.CN
School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China
Department of Mathematics, The University of Hong Kong, Hong Kong SAR, China

Liping Jing* LPJING@BJTU.EDU.CN
Jingxuan Wen JINGXUAN@BJTU.EDU.CN
Pengyu Xu PENGYUXU@BJTU.EDU.CN
Jiaqi Wang JIAQI.WANG@BJTU.EDU.CN
Jian Yu JIANYU@BJTU.EDU.CN
School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China

Michael K. Ng MNG@MATHS.HKU.HK
Department of Mathematics, The University of Hong Kong, Hong Kong SAR, China

Editor: Samy Bengio

Abstract

User preference modeling in recommendation system aims to improve customer experience through discovering users' intrinsic preference based on prior user behavior data. This is a challenging issue because user preferences usually have complicated structure, such as inter-user preference similarity and intra-user preference diversity. Among them, inter-user similarity indicates different users may share similar preference, while intra-user diversity indicates one user may have several preferences. In literatures, deep generative models have been successfully applied in recommendation systems due to its flexibility on statistical distributions and strong ability for non-linear representation learning. However, they suffer from the simple generative process when handling complex user preferences. Meanwhile, the latent representations learned by deep generative models are usually entangled, and may range from observed-level ones that dominate the complex correlations between users, to latent-level ones that characterize a user's preference, which makes the deep model hard to explain and unfriendly for recommendation. Thus, in this paper, we propose an **In**terpretable **D**eep **G**enerative **R**ecommendation **M**odel (**InDGRM**) to characterize inter-user preference similarity and intra-user preference diversity, which will simultaneously disentangle the learned representation from observed-level and latent-level. In InDGRM, the *observed-level disentanglement on users* is achieved by modeling the user-cluster structure (i.e., inter-user preference similarity) in a rich multimodal space, so that users with similar preferences are assigned into the same cluster. The *observed-level disentanglement on items* is achieved by modeling the intra-user preference diversity in a prototype learning strategy, where different user intentions are captured by item groups (one group refers to one intention). To promote disentangled latent representations, InDGRM adopts structure and sparsity-inducing penalty and integrates them into the generative procedure, which has ability to enforce each latent factor focus on a limited subset of items (e.g., one item group) and benefit *latent-level disentanglement*. Meanwhile, it can be efficiently inferred by minimizing its penalized upper bound with the aid of local variational optimization technique. Theoretically, we analyze the generalization error bound of InDGRM to

*. Corresponding author.

guarantee its performance. A series of experimental results on four widely-used benchmark datasets demonstrates the superiority of InDGRM on recommendation performance and interpretability.[1]

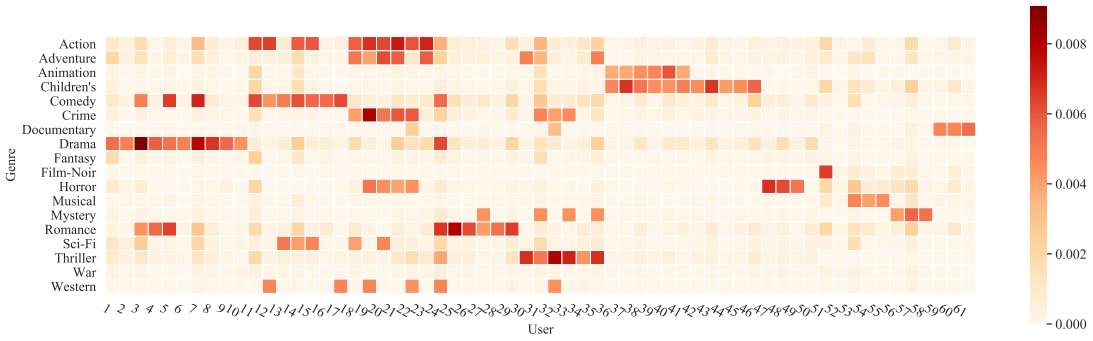
Keywords: Recommendation System, Collaborative Filtering, Deep Generative Model, Interpretable Machine Learning, Latent Factor Model

1. Introduction

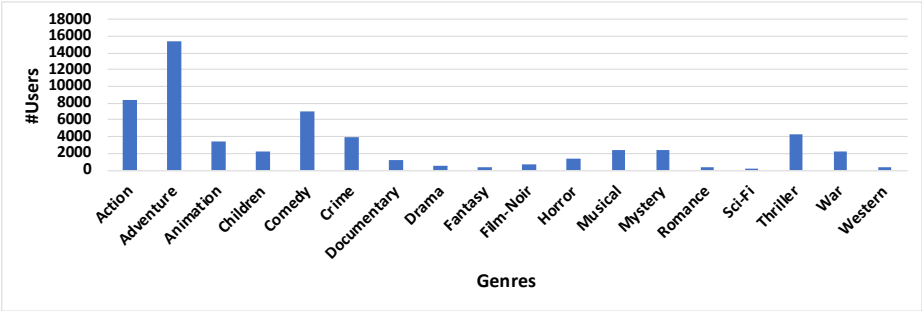
Interpretability can be defined as "the degree to which a human can understand the cause of a decision". Making machine learning models explainable helps understand why the models succeed or fail, and could give us better intuition about the problem and higher trust in the solution. More fundamental need for the explainability stems from an incompleteness in the problem formalization. Explainable recommendation refers to personalized recommendation algorithms that address the problem of why - they not only provide users or system designers with recommendation results, but also explanations to clarify why such items are recommended (Zhang and Chen, 2020). In this way, it helps to improve the transparency, persuasiveness, effectiveness, trustworthiness, and user satisfaction of recommendation systems. It also facilitates system designers to diagnose, debug, and refine the recommendation algorithm. Even though significant progress has been made for recommendation, they hardly realize the underlying reasons behind the users' decision making processes since user preferences are highly similar and diverse, and may range from inter-user preference similarity that governs preference relationship among users to intra-user preference diversity that characterizes a user's diverse preference when executing an intention.

Learning representations that reflect users preference, based chiefly on user behavior, has been a central theme of research on recommendation systems (Liang et al., 2018; Liu et al., 2019b; Ma et al., 2019). However, complex user preference is hardly to model and the learned representations are highly entangled and may range from observed-level ones that dominate the complex correlations between users and items and latent-level ones that characterize a user's preference. On the one hand, the latent representations of different users are assumed independent to each other when building the model. Actually, in real applications, different users may interact with each other due to inter-user preference similarity on behaviors. By using *Movielens 20M*[2] dataset as an example, we investigate the genres that each user likes[3]. Figure 1(a) demonstrates the genres distribution related to 61 users for each genre[4]. We can see that although different users have different genre distributions, users show a strong correlation on genres. Furthermore, for each user, we sort the genres in descending order according to the number of genres interactions, and take the top 20% genres as the main preference of users. Figure 1(b) demonstrates the number of users who like each genre together in terms of genres. We can see that there are a large number of users with the same

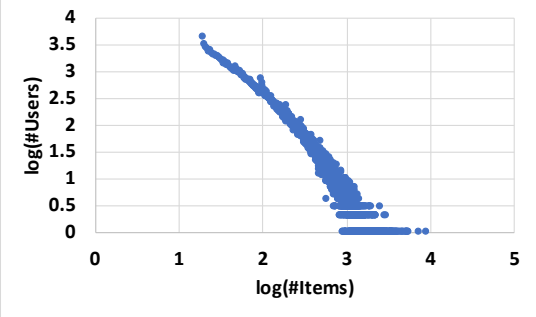
1. A preliminary version of this work was presented in *Proceedings of the Web Conference (WWW)*, 2020 (Liu et al., 2020).
2. https://grouplens.org/datasets/movielens/
3. In *Movielens 20M* dataset, each user interacted with several items and each item is labeled by one or more genre labels.
4. We count the interaction proportion of each user on different genres, and get the distribution of genres. For each type of genres, we select the top 10 users to display. Considering that some users have more interactions on multiple genres, a total of 61 users are statistically displayed.



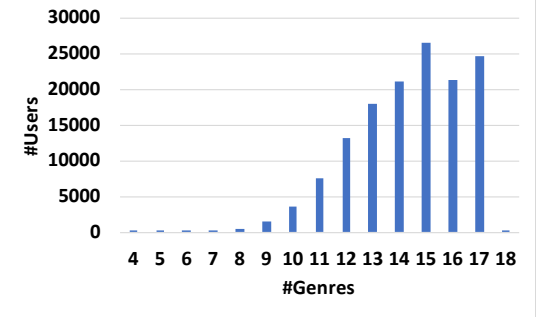
(a) Genres distribution related to top 10 users for each genre



(b) The number of users who like each genre together in terms of genres



(c) The number of interacted items along all users



(d) The size of genres corresponding to items which are interacted by each user

Figure 1: Some data distributions in *Movielens 20M* dataset.

preferences. Although different users may have different preference, users with the same preference genre show a strong correlation. Thus, ignoring correlation between users may destroy the data distribution estimation. Meanwhile, such correlations between users are indeed helpful to explain the recommendation result, which has been proven by structural user models (Balog et al., 2019).

On the other hand, the user representations can be highly entangled in the latent-level, which preserves the confounding of the factors and is prone to mistake the relationship between the latent factors and the observed user behavior, and further leads to non-robust recommendation and low interpretability. In this case, the system designers can not inter-

pret the learned latent factors, and the end-user can not understand the final recommendation results. It is interesting to note that the intra-user preference diversity where the preferences of a user are always diverse but sparsely distributed with respect to the whole item space (Zhou et al., 2018; Liu et al., 2019a, 2020). Figure 1(c) shows the distribution about the number of interacted items along all users. Many users are only interested with a few items, which leads to a power-law distribution. This statistical observation confirms that the interested item data set is sparsely distributed. Figure 1(d) demonstrates the size of genres corresponding to items which are interacted by each user. It can be found that most of the users are interested in more than ten genres, which indicates that users have diverse interests. Thus, it is necessary to disentangle the latent representations so that different factors can capture different preferences for each user. Such intra-user preference diversity modeling is critical to improve the recommendation performance (Ma et al., 2019).

In this paper, based on our prior user preference modeling work (Liu et al., 2020), we propose a new Interpretable Deep Generative Recommendation Model (InDGRM) to characterize user behavior from both inter-user preference similarity and intra-user preference diversity modeling, and achieve latent-level and observed-level disentanglements for interpretable recommendation. In InDGRM, inter-user preference similarity indicates strong correlations among a subset of users who have similar preference, and the latent user representation is modeled by a mixture prior in a rich multimodal space, and further achieve *observed-level disentanglement on users* by separating users into different preference groups. Intra-user preference diversity represents user's intrinsic diverse preference since one user might be interested in different kinds items, which is model by identifying the item groups via learning a set of prototypes to achieve *observed-level disentanglement on items*, based on which the user intention related with each item is inferred, and then capturing the preference of a user about the different intentions separately. We promote disentangled latent representations by introducing structure and sparsity-inducing penalty into a generative procedure, which enforces each latent factor to influence a limited subset of items (i.e., item groups) and achieve *latent-level disentanglement*. To effectively handle the optimizing process, we adopt Wasserstein auto-encoder (WAE) framework (Tolstikhin et al., 2018) to measure the true preference data distribution and the generated data distribution. InDGRM can be efficiently inferred via the local variational optimization technique. Theoretically, we provide its generalization error bound to guarantee its performance. A series of experiments are conducted on four real-world datasets, and the results have demonstrated that InDGRM outperforms the state-of-the-art baselines in terms of several popular evaluation metrics. And the learned disentanglement on latent representation and observed behavior is demonstrated to be interpretable.

2. Related Work

In this section, we briefly review two different areas which are highly relevant to the proposed method, interpretable recommendation and user preference modeling.

2.1 Interpretable Recommendation

Interpretable recommendation can not only provide interpretation to a user why the items are recommended, but also facilitate system designers to diagnose, debug, and refine the

recommendation models. Early interpretable approaches to personalized recommender systems mostly focused on content-based or collaborative filtering (CF) based recommendation (Adomavicius and Tuzhilin, 2005; Pazzani and Billsus, 2007; Sarwar et al., 2001; Konstan et al., 1997). Content-based recommendation attempts to model user and/or item profiles with various available content information, such as the price, color, brand of the goods in e-commerce, or the genre, director, duration of movies in review systems. Since the item contents are much understandable to the users, they are always used to explain why an item is recommended out of other candidates in content-based recommendation. However, collecting content information in different application scenarios is expensive and time-consuming. The most popular CF-based recommendation includes user-based CF and item-based CF (Konstan et al., 1997). Those methods can provide intuitive interpretations, such as "*customers who bought this item also bought...*" in user-based CF and "*the item is similar to your previously loved items*" in item-based CF. Leveraging the neighborhood style explanation mechanism, Abdollahi and Nasraoui (2016) presented an explainable matrix factorization (EMF) by extending matrix factorization-based CF model via an explainability regularizer. Even though EMF provides explanation, it prefers to the popular items recommendation. Inspired by the successes of attention-based deep learning methods, researchers presented neural attentive interpretable recommendation model (NAIRS) (Yu et al., 2019) to extend the traditional item similarity model (Kabbur et al., 2013). In our recent work (Liu et al., 2019c), an influence-based interpretable recommendation was proposed to modeling the influence of historical interactions. Recently, Ma et al. (2019) focused on disentangled representation for interpretable recommendation. Although the above methods are able to provide interpretable results, the learned deep latent features are too entangled in preference space to investigate internal recommendation mechanism.

Another stream of work seeks for recommendation interpretations from auxiliary information. For example, textual reviews and tag information are studied in additional to the basic recommender model. To make use of textual reviews, topic model is integrated with matrix factorization to determine the explicit review-aware item features which are aligned to the latent factor for explanations generation (Mcauley and Leskovec, 2013; Zhang et al., 2014). Recently, due to the powerful ability in representation learning, deep learning is adopted in recommendation to model textual content and generate the explanations (Seo et al., 2017; Chen et al., 2018b; Donkers et al., 2017; Chen et al., 2018a). For instance, Seo et al. (2017) introduced an interpretable convolutional neural network to learn the item feature from users' review information. Donkers et al. (2017) combined the user-item interaction and review information in a unified LSTM framework. Moreover, social trust information has been proved an alternative view of user preference to improve trustworthiness and transparency for recommendation (Park et al., 2018), and the utilization of tags for explainable recommendation has been particularly well studied (Balog et al., 2019). We in this work target on making interpretable recommendation from implicit feedback data only. The key is to infer the interpretable disentangled factor from observed feedback data, while related work discussed here aims at linking the auxiliary information with the recommendation decisions. Our study thus has a different problem setting and is generally applicable to systems where the auxiliary information is unavailable.

2.2 User Preference Modeling

User preference modeling in recommendation systems aims to explore users' intrinsic preferences to improve recommendation performance. Most existing recommender systems represent a user's preference with a feature vector, which is assumed to be fixed or followed in the same distribution when predicting this user's preferences for different items, e.g., latent factor models (Mnih and Salakhutdinov, 2008; Salakhutdinov et al., 2007; Xue et al., 2017). However, the same vector or distribution cannot accurately capture a user's varying preferences on all items, especially when considering the diverse characteristics of various items.

In order to capture the correlations among users and diverse preferences, recently, researchers proposed several localized latent factor models (Lee et al., 2016; Wu et al., 2016; Zhang et al., 2017; Liu et al., 2019a, 2020, 2021a,b) by exploiting local structure of the large-scale preference matrix. The main idea focuses on dividing the whole preference matrix into several submatrices so that each submatrix contains a set of like-minded users and the items that they are interested in. To sufficiently reduce the approximation error, the original preference matrix is partitioned several times to get a set of approximated preference matrices, and reconstructed with them and the corresponding weights in an ensemble manner. In literature, a simple way to capture local structure is randomly selecting users/items to form submatrices with similar interests (Mackey et al., 2011), but it can not guarantee that the users in the same submatrix share the common interests and the items have the similar categories. To address this problem, several works (Lee et al., 2016; Chen et al., 2015; Beutel et al., 2015; Wang et al., 2016; Zhang et al., 2017) are proposed to effectively partition preference matrix. Lee et al. (2016) proposed a local low-rank matrix approximation method using a kernel smoothing nearest neighbors method to acquire local structure and represent rating matrix as a weighted sum of several local low-rank matrices. In Beutel et al. (2015), Bayesian co-clustering was proposed to determine the local structure and a concise model was designed for matrix approximation in an additive strategy. Zhang et al. (2017) proposed a heuristic anchor-point selecting method to enhance local low-rank matrix approximation. However, these methods assign each user/item to only one single cluster, which make them can not handle the users with multiple interests well. Thus, researchers introduced an affiliation score to characterize the strength between user/item and the corresponding submatrices (Zhang et al., 2013; Wu et al., 2016). Our latest work DGLGM aims to learn global and local user representation in a deep generative model and achieve promising recommendation performance (Liu et al., 2020). However, above methods modeling user preference solely focus on partitioning user well and neglect interpretability of recommendation results.

Thus, in this paper, we focus on modeling user preference on both inter-user preference similarity and intra-user preference diversity and achieving interpretable recommendation by investigating observed-level and latent-level disentanglement based on our previous work (Liu et al., 2020).

3. Notations and Problem Formulation

Let calligraphic letter (e.g., $\mathcal{A}$) indicate set, lower or upper case regular letter (e.g., a or A) for scalar, lower-case bold letter (e.g., $\mathbf{a}$) for vector, and upper-case bold letter (e.g.,

A) for matrix. Suppose there are N users and M items, $\mathcal{X} = \{(i, j, r), i \in \mathcal{D}_I, j \in \mathcal{D}_J, r \in \{0, 1, \cdots, R\}$ denotes user-item preference set where (i, j, r) indicates the i-th user gives preference r to the j-th item (here preference value r is defined by non-negative integer value.), $\mathcal{D}_I$ and $\mathcal{D}_J$ are user set and item set respectively. In most scenarios, preference value r is a binary value (i.e., 0 or 1) since it is collected implicitly, while, in few areas, the explicit preference information (e.g., ratings) can be collected. Thus, in this paper, we focus on modeling user's implicit feedback to achieve interpretable recommendation. Let $\mathbf{X} \in \mathbb{N}^{N \times M}$ denote the user-item interaction matrix among N users and M items. Its element $x_{i,j} \in \{0, 1\}$ indicates whether the j-th item is interacted by the i-th user. $\mathbf{x}_i = [x_{i,1}, \cdots, x_{i,M}]^\top \in \mathbb{N}^M$ is a binary vector demonstrating the interaction history of user i on all items.

The general goal of deep generative recommendation method is to determine the latent user representation by defining proper prior and generative process to user-item interactions. In this work, we focus on learning latent user representations $\{\mathbf{z}_i\}_{i=1}^N$ ($\mathbf{z}_i \in \mathbb{R}^{d_i}$, d_i is the size of the optimal latent space for current user) to model inter-user preference similarity and intra-user preference diversity, and achieve the latent-level and observed-level disentanglement of the representations for interpretable recommendation.

Inter-user preference similarity: In recommendation platform, users with similar preference may affect each other. In this case, it is intuitive to assume that users with similar preference follow the same distribution, and their latent representations can be generated in same way. Specifically, the users can be separated into several preference groups, where users in the same group share the same interests (preference). This grouping operation is helpful to achieve observed-level disentanglement on users. Given one user's interaction behavior $\mathbf{x}_i$, the corresponding latent user representation $\mathbf{z}_i \in \mathbb{R}^{d_i}$ can be modeled in a rich multimodal space. In this space, the whole preference space is partitioned into K disjoint clusters, such that users in the same group are close to each other, while users in different groups are far away from each other in terms of user's preference. Meanwhile, different user groups can be spanned by their own optimal latent space. Here the users are grouped into K disjoint clusters to capture the inter-user preference similarity.

Intra-user preference diversity: One user might be interested in different kinds of items, especially when the system contains a large set of items with various types. In literatures, there is another line of work to handle preference diversity by introducing overlapping clusters (Lee et al., 2016). In this work, we investigate the intra-user preference diversity by introducing item groups. Specifically, a set of multi-hot vectors $\mathbf{C} = \{\mathbf{c}_j\}_{j=1}^M$ are used to indicate the item-category membership. For the j-th item, $\mathbf{c}_j = [c_{j,1}, c_{j,2}, \cdots, c_{j,G}]^\top \in \mathbb{N}^G$, and $c_{j,g} = 1$ if item j belongs to category g, otherwise $c_{j,g} = 0$. $\mathbf{C}$ is helpful to achieve observed-level disentanglement on items. To capture the essential user preference in each item category (group), the interaction behavior data of user i is split into G groups, denoted as $\mathbf{x}_i = \{\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}, \cdots, \mathbf{x}_i^{(G)}\}$, where $\mathbf{x}_i^{(g)}$ contains the interaction data corresponding to the items in group g. Meanwhile, in the latent space for user representation, each latent factor can be connected to the explicit item group for latent-level disentanglement. Here, a structure and sparsity mapping matrix $\mathbf{O}_i = [\mathbf{o}_i^{(1)}, \mathbf{o}_i^{(2)}, \cdots, \mathbf{o}_i^{(G)}] \in \mathbb{R}^{d_i \times G}$ between factors and item groups is designed for each user.

Modeling inter-user preference similarity and intra-user preference diversity is expected to properly capture the user preference and output more accurate recommendation results,

so as to achieve observed-level disentanglement on users and items, as well as latent-level disentanglement on representations for interpretable recommendation. In the next section, we will describe the model in detail.

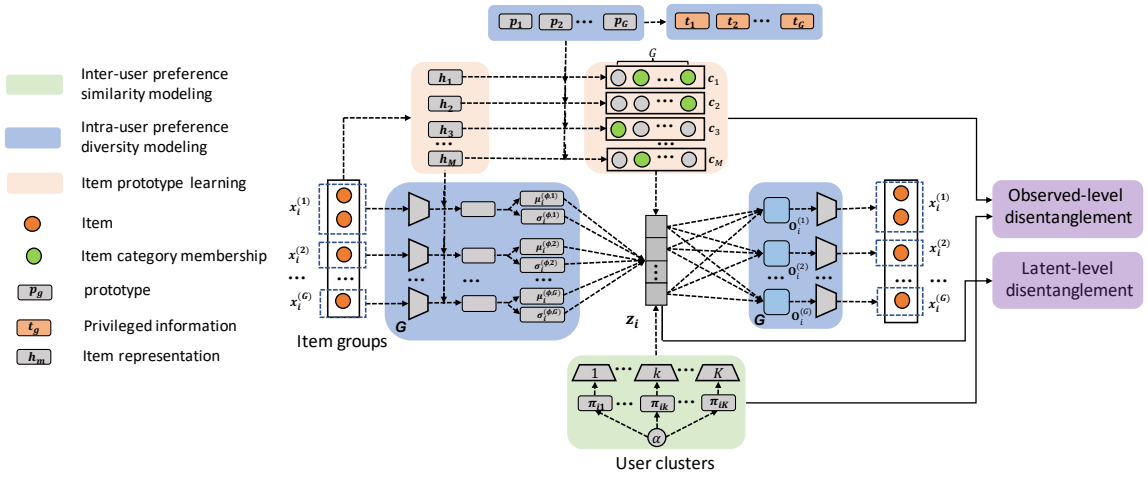


Figure 2: Architecture of the proposed InDGRM by modeling both inter-user preference similarity and intra-user preference diversity to achieve observed-level disentanglement and latent-level disentanglement. Inter-user preference similarity indicates strong correlation among a subset of users who have similar preference, which is achieved by learning a non-parametric mixture prior with several components to capture the correlation among users (user clusters). Intra-user preference diversity indicates the inherent diversity preferences of a single user, which is achieved by learning a set of item group prototypes, based on which the user intention related with each item is inferred, and then capturing the preference of a user about the different intentions separately. Both inter-user preference similarity and intra-user preference diversity modeling strategies focus on achieve disentanglement in observed-level and latent-level, i.e., users clusters, item groups, and disentangled latent representations.

4. The Proposed Method

In this section, an Interpretable Deep Generative Recommendation Model (InDGRM) will be presented by investigating user behaviors from the views of *inter-user preference similarity* and *intra-user preference diversity*. Our goal is to achieve interpretable mechanism for deep learning architecture from two perspectives, *observed-level disentanglement* and *latent-level disentanglement*. The architecture of InDGRM is given in Figure 2.

InDGRM consists of three modules. One module is to capture the inter-user preference similarity and achieve observed-level disentanglement on users via user grouping technique with multimodal prior on latent user representations. The second module is to characterize the intra-user preference diversity and achieve observed-level disentanglement on items via

item prototype learning technique. The third module aims to disentangle the latent factor with the aid of item groups and achieve latent-level disentanglement for interpretable recommendation. In InDGRM, the behavior data of each user can be generated via a hierarchical generative model with two layers of latent variables as follows:

$$p_G(\mathbf{x}_i) = \sum_{k=1}^{K} \pi_{i,k} \mathbb{E}_{p_\theta(\mathbf{C})} \left[\int \prod_{g=1}^{G} p_\theta \left(\mathbf{x}_i^{(g)} | \mathbf{z}_i, \mathbf{o}_i^{(g)}, \mathbf{C} \right) p_\theta(\mathbf{z}_i|k) p_\theta(k|\boldsymbol{\pi}_i) p_\theta(\mathbf{o}_i^{(g)}|\gamma) p(\gamma) d\gamma d\mathbf{z}_i \right] \tag{1}$$

Here $p_\theta(\mathbf{C})$ indicates the item-group correlation distribution which aims to assign items into different groups. $\boldsymbol{\pi}_i \in \mathbb{R}^K$ is the prior cluster probability for the i-th user and $\sum_{k=1}^{K} \pi_{i,k} = 1$ (K is the number of clusters). $p_\theta(k|\boldsymbol{\pi}_i)$ is the categorical distribution parameterized by $\boldsymbol{\pi}_i$, and θ is the learnable parameters. The discrete latent variable k indicates the corresponding cluster that the i-th user belongs to. According to the assigned user cluster k, the latent user representation $\mathbf{z}_i \in \mathbb{R}^{d_i}$ can be obtained by $p_\theta(\mathbf{z}_i|k)$. The sparsity mapping $\{\mathbf{o}_i^{(g)}\}_{g=1}^{G}$ ($\mathbf{o}_i^{(g)} \in \mathbb{R}^{d_i}$) from latent factors to item groups is modeled by a sparsity-inducing penalty $p_\theta(\mathbf{o}_i^{(g)}|\gamma)p(\gamma)$ (γ is a variable to control the sparsity, which can be sampled from a *Gamma* distribution), which will lead to a disentangled latent representation. Finally, the user-item interaction behavior data $\mathbf{x}_i$ can be generated through the pre-defined distribution parameterized with latent representation $\mathbf{z}_i$, item group assignment $\mathbf{C}$ and sparsity mapping $\{\mathbf{o}_i^{(g)}\}_{g=1}^{G}$. Next, we will describe the detailed implementation of above generative procedure.

4.1 Inter-user Preference Similarity Modeling

The general goal of deep generative recommendation method is to determine the latent user representation $\mathbf{z}_i$ by defining proper prior and generative process from latent representation to observed user-item interaction data. Among it, the prior is usually crucial for recommendation data generation. In literatures, most researches restrict user latent representation in a univariate *Gaussian* space, which means that all users come from a single preference pattern. As aforementioned, different users may have different habits, while some of them may be effected to each other due to their similar preference. In this case, a simple prior, e.g., *Gaussian* distribution, becomes unreasonable to model such complicated user structure.

In order to properly capture the inter-user preference similarity and achieve observed-level disentanglement on users, the mixture model is adopted as the prior distribution of latent representation, because it has been proved was a universal approximator for any continuous density function and able to characterize the recommendation data well (Ma et al., 2019; Liu et al., 2020). Nevertheless, the existing methods have to predefine the number of mixture components K and share the same latent space size in all components, which often leads to a priori inaccuracy. Such hyperparameter setting definitely affects the final recommendation performance. For instance, if K is too small, the mixture model may not be able to well capture the complicated local structure from the wide-ranging sources. On the other hand, if K is too large, it will be time-consuming to learn all components even though most of them may make small contribution. Moreover, without a proper prior assumption for the mixing coefficient, the mixture model may be unstable and result in overfitting (Chen et al., 2016). Different components have their own structure or own space, therefore, it is necessary to determine the optimal feature subspace for each component.

4.1.1 User group structure identification

To capture the group structure among users, a deep mixture generative recommendation model is designed, with which the latent user representation can be generated via

$$\mathbf{z}_i \sim \sum_{k=1}^{K} \pi_{i,k} p_\theta(\mathbf{z}_i|k) p_\theta(k|\boldsymbol{\pi}_i). \tag{2}$$

Here K is the number of components in mixture model, which is usually taken as a predefined parameter. However, it is hard to previously set proper value for K. To automatically determine the number of components, we exploit non-parametric Bayesian technique which provides an elegant solution on automatic adaptation of model capacity. Such adaptivity can be obtained by defining stochastic processes on rich measure spaces such as Dirichlet Process (Neal, 2000), Indian Buffet Process (Ghahramani and Griffiths, 2006) and etc. In this work, the Dirichilet Process is adopted. More specifically, the latent variable $\mathbf{z}_i$ is generated via a hierarchical structure with two layers according to $p_\theta(\mathbf{z}_i|k)p_\theta(k|\boldsymbol{\pi}_i)$ and user-component membership $\boldsymbol{\pi}_i$. Among it, $\boldsymbol{\pi}_i$ is sampled from Griffiths-Engen-McCloskey (GEM) distribution (Pitman, 2002) via $\boldsymbol{\pi}_i \sim GEM(\alpha)$ (with $\sum_{k=1}^{\infty} \pi_{i,k} = 1$), which is a special case of Dirichlet Process (DP). $GEM(\cdot)$ distribution is able to construct an infinite data partition and can be efficiently constructed via a stick-breaking process. Considering a stick with unit length, it can be broken into an infinite number of segments $\{\pi_{i,k}\}_{k=1}^{\infty}$ by the following process with parameter $v_k \sim \mathcal{B}eta(1, \alpha)$:

$$\pi_{i,k} = \begin{cases} v_1 & \text{if} \quad k = 1, \\ v_k \prod_{l=1}^{k-1}(1 - v_l) & \text{for } k > 1. \end{cases} \tag{3}$$

The precision parameter α controls the number of significant sticks that have appreciable weights $\pi_{i,k}$. This process provides insights for developing variational approximate inference algorithms (Blei and Jordan, 2006).

In each component, the data is assumed following a *Gaussian* distribution. Then, the prior of user representation $\mathbf{z}_i$ along all components can be formulated as:

$$\begin{aligned} \mathbf{z}_i &\sim \sum_{k=1}^{\infty} \pi_{i,k} \mathcal{N}\left(\boldsymbol{\mu}_i^{(k)}, \boldsymbol{\sigma}_i^{(k)} \mathbf{I}\right) \\ \boldsymbol{\pi}_i &\sim GEM(\alpha) \quad \text{with} \sum_{k=1}^{\infty} \pi_{i,k} = 1 \end{aligned} \tag{4}$$

here $\boldsymbol{\mu}_i^{(k)}$ and $\boldsymbol{\sigma}_i^{(k)}\mathbf{I}$ are mean and variance of *Gaussian* distribution related to the k-th component. For the users with similar preference, they are expected to have similar user-component distribution $\boldsymbol{\pi}_i$. In this case, their latent representation $\mathbf{z}_i$ will approach to each other.

4.1.2 Optimal subspace determination

For each component corresponding to one preference cluster, to capture its own structure, its latent feature space including space size is automatically determined. For convenient

computing and disentangled representation, the latent features in $\mathbf{z}_i$ are assumed independent to each other and each feature has zero mean. Then, the *Gaussian* distribution $\mathcal{N}(\boldsymbol{\mu}_i^{(k)}, \boldsymbol{\sigma}_i^{(k)}\mathbf{I})$ related to the k-th component can be modeled as:

$$\begin{aligned} &\mathcal{N}\left(\boldsymbol{\mu}_i^{(k)}, \boldsymbol{\sigma}_i^{(k)}\mathbf{I}\right) = \prod_{l=1}^{d^{(k)}} \mathcal{N}\left(0, \lambda_l^{(k)}\right) = \mathcal{N}\left(0, \boldsymbol{\Lambda}^{(k)}\right) \\ &\lambda_l^{(k)} \sim \mathcal{IG}(\eta_a, \eta_b) \end{aligned} \tag{5}$$

where $\boldsymbol{\mu}_i^{(k)} \in \mathbb{R}^{d^{(k)}}$ and $\boldsymbol{\Lambda}^{(k)} = \text{diag}(\lambda^{(k)}) \in \mathbb{R}^{d^{(k)} \times d^{(k)}}$ are the mean vector and covariance matrix respectively. In the covariance matrix, its l-th diagonal element ($\lambda_l^{(k)}$) is sampled from *Inverse Gamma* distribution with parameters η_a and η_b. To determine the optimal dimensionality $d^{(k)}$, we take advantage of the automatic relevance determination (ARD) technique (Neal, 1996; Wipf and Rao, 2007). Since each latent feature is assumed having zero mean, the feature with small variance $\{\lambda_l^{(k)}\}$ will shrink to zero. In this case, the latent features with small variance can not contribute to characterize users, thus, we can remove them. In other words, only the latent features with larger variance (empirically greater than 0.001) are useful to form the latent space. A good by-product is that each component can contain its own latent space size ($d^{(k)}$).

Note that the generative process is built on the truncated stick-breaking process at step K, which has been proved to closely approximate a true Dirichlet Process as long as K is chosen to be large enough (Ishwaran and James, 2001). Empirically K may be initialized to some value from tens to hundreds based on the model complexity. The useless dimensions will gradually be pruned automatically. For our case, small $\boldsymbol{\pi}_{ik}$ indicates that it is very unlike for some entries belonging to the corresponding cluster, thus, we can prune such clusters because they make few contribution for generation process. As a consequence, the users can be clustered into K groups without a complicated model selection procedure. Meanwhile, user clustering is able to achieve observed-level disentanglement on users.

4.2 Intra-user Preference Diversity Modeling

To capture the preference diversity for each user, all interacted items are automatically grouped via prototype learning technique.

4.2.1 Prototype-based item group assignment

To sufficiently exploit group structure among items, G category prototypes $\{\mathbf{p}_g\}_{g=1}^G$ are introduced. Meanwhile, a *Multinomial* distribution $p_\theta(\mathbf{c}_j)$ over G groups is used to model the j-th item along all groups, where $\mathbf{c}_j = [c_{j,1}, c_{j,2}, \cdots, c_{j,G}]^\top$ is a multi-hot vector drawn from

$$\begin{aligned} &\mathbf{c}_j \sim \mathcal{M}ult(N_j, \mathbf{s}_j) \\ &s_{j,g} \propto \exp\left(\frac{\cos(\mathbf{h}_j, \mathbf{p}_g)}{\tau}\right). \end{aligned} \tag{6}$$

where $\{\mathbf{p}_g\}_{g=1}^G$ indicates the G category prototypes and $\{\mathbf{h}_j\}_{j=1}^M$ indicate the item latent representations. The correlation between each pair of item and prototype is measured via

cosine similarity[5] $\cos(\mathbf{h}_j, \mathbf{p}_g) = \frac{\mathbf{h}_j^\top \mathbf{p}_g}{||\mathbf{h}_j|| ||\mathbf{p}_g||}$ and normalized via a softmax function to produce a probability vector $\mathbf{s}_j = [s_{j,1}, s_{j,2}, \cdots, s_{j,G}]^\top \in \mathbb{S}^{G-1}$ in a $(G-1)$-simplex. Among it, $s_{j,g}$ quantifies the correlation between the j-th item and the g-th group. The larger the correlation value is, the higher probability that the j-th item belongs to the g-th group. The hyperparameter τ aims to scales the similarity from $[-1, 1]$ to $[-\frac{1}{\tau}, \frac{1}{\tau}]$. In experiments, τ is set to be 0.1 for more skewed distribution. N_j is the number of groups to which the j-th item belongs.

With the aid of the above variables, the multi-hot group membership vector $\mathbf{c}_j$ can be sampled from a *Multinomial* distribution with probability $\mathbf{s}_j$. According to the distribution of $\mathbf{c}_j$, i.e., $p_\theta(\mathbf{c}_j)$, we can get the item-group distribution $p_\theta(\mathbf{C}) = \prod_{j=1}^{M} p_\theta(\mathbf{c}_j)$ under the assumption that all items are drawn from independent and identically distributed (*i.i.d.*). The item j will be assigned to the group $g*$ if $g* = \arg\max_g \{c_{j,g}|_{g=1}^{G}\}$.

4.2.2 Prototype enhancing with privileged information

According to the cognitive science(Hampton, 1993), the prototypes are expected to demonstrate some concepts or categories, which are usually average or best exemplars. Prototypes provide a concise representation for an entire group (category) of entities, providing means to anticipate hidden properties and interact with novel stimuli based on their similarity to prototypical members of their group. A prototype learning system is a system of cognitive processes together with their underlying neural structures that enables one to learn a category prototype from a set of data points(Zeithamova, 2012). The prototype learning on pure interacted data is hard to achieve this goal. Fortunately, in the real-world applications, item is usually labeled with special characteristics, such as genre information of movies(Harper and Konstan, 2015), category information of goods(Ma et al., 2019) and etc. Such kind of privilege information has been proved to be effective for prototype learning (Vapnik and Izmailov, 2015). Thus, in this work, they are taken as prior knowledge and incorporated in the prototype learning process.

Our main idea is to construct the triple set from item category or genre information, and model the alignment between prototype relations and the given category relations. Specifically, the category description can be collected and preprocessed as semantical representations $\{\mathbf{t}_g\}_{g=1}^{G}$ via word embedding technique. The triplet of categories can be constructed, where three categories in one set have the following indicator $\bar{s}_{j,k,g} = \mathbb{1}(\cos(\mathbf{t}_g, \mathbf{t}_j) \geq \cos(\mathbf{t}_g, \mathbf{t}_k))$ to demonstrate the ranking order of prototype j and k when given prototype g. With the help of $\bar{s}_{j,k,g}$, the prototypes triplet ($\mathbf{p}_g$, $\mathbf{p}_j$, $\mathbf{p}_k$) can be

5. Here cosine similarity is adopted instead of the inner product similarity which is used by most existing deep learning methods (He et al., 2017), because it is crucial for preventing mode collapse. Empirically, with inner product, the majority of the items are highly likely to be assigned into a single category $\mathbf{p}_g$ with an extremely large norm. In contrast, cosine similarity avoids this degenerate case due to the normalization. Moreover, cosine similarity is related with the Euclidean distance on the unit hypersphere, and the Euclidean distance is a proper metric that is more suitable for inferring the cluster structure, compared to inner product.

constrained by the following *Bernoulli* distribution:

$$
\begin{aligned}
s_{j,k,g} &\propto \exp\left\{\frac{\cos(\mathbf{h}_j, \mathbf{p}_k)}{\tau} - \frac{\cos(\mathbf{h}_j, \mathbf{p}_g)}{\tau}\right\} \\
\bar{s}_{j,k,g} &\sim \mathcal{B}(s_{j,k,g})
\end{aligned}
\tag{7}
$$

where the ranking order likelihood $s_{j,k,g}$ is modeled via a logistic function which maps the similarity difference to probability. Incorporating similarity order, instead of direct similarity, will enforce the hyperspherical prototypes have the same similarity ranking order as the priors $\{\mathbf{t}_g\}_{g=1}^{G}$, which is much more reasonable in real applications (Mettes et al., 2019).

A good by-product of item grouping is to achieve observed-level disentanglement on items. Specially, the observed interactions behavior data of the i-th user ($\mathbf{x}_i$) can be separated into G groups, $\mathbf{x}_i = [\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)} \cdots, \mathbf{x}_i^{(G)}]$. $\mathbf{x}_i^{(g)}$ contains the interaction data related to the items which belong to the g-th group.

4.3 Factor-to-group Sparse Mapping

As we known, the latent representations learned by deep generative models are usually entangled, which makes the deep model hard to explain and non-transparent for recommendation. A straight but efficient way to handle this issue is to disentangle the latent factors. In our work, the item groups with category information provide high-level and interpretable concepts. Thus, we can connect the latent features of $\mathbf{z}_i$ with the item groups for interpretable recommendation model construction. Inspired by the sparse factor analysis (Tibshirani, 1996; Tank et al., 2018), the latent features can be well explained by encouraging sparse mappings from latent features to item groups. Specifically, a group-specific factor mapping vector $\mathbf{o}_i^{(g)} = [o_{i,1}^{(g)}, o_{i,2}^{(g)}, \cdots, o_{i,d_i}^{(g)}]^\top \in \mathbb{R}^{d_i}$ is introduced between the latent representation and item group for the current user i, where d_i is the optimal latent space size (as shown in the right part of Figure 2).

Note that the l-th element of the mapping vector, i.e., $o_{i,l}^{(g)}$, indicates the influence of the items in group g on the l-th latent factor for user i. To make the influence more explainable, $o_{i,l}^{(g)}$ is set to be 0 if the g-th item group have no influence on the l-th latent factor. Meanwhile, each latent factor should only focus on as few items as possible, i.e., the mapping vector should be sparse (Anindya et al., 2015). To implement this, a hierarchical distribution with Bayesian prior $p(\gamma)$ is introduced to model $o_{i,l}^{(g)}$ as follows:

$$
\begin{aligned}
\gamma^2 &\sim \Gamma\left(\frac{a+1}{2}, \frac{b^2}{2}\right) \\
o_{i,l}^{(g)} &\sim \mathcal{N}(0, \gamma^2)
\end{aligned}
\tag{8}
$$

where sparsity variance γ^2 is sampled from *Gamma* distribution parametrized by shape parameter $\frac{(a+1)}{2}$ and rate parameter $\frac{b^2}{2}$. Among it, the sparsity can be controlled by rate parameter $\frac{b^2}{2}$, larger b implying more sparse in $\mathbf{o}_i^{(g)}$. Note that deep structure allows rescaling of the parameters across layer boundaries without affecting the end behavior of the networks.

With the aid of mapping vectors $\{\mathbf{o}_i^{(g)}\}_{g=1}^G$, the generative process of InDGRM has the ability to capture the group-specific latent feature. Obviously, such latent feature learning is helpful to achieve disentangled latent representation. In addition, since each item group is related to one type of user preferences, such group-specific latent feature is beneficial to model intra-user preference diversity.

4.4 Interaction Data Generation

Once having the latent representation $\mathbf{z}_i$ for user i, sparse mapping vectors $\{\mathbf{o}_i^{(g)}\}_{g=1}^G$ and item representations $\{\mathbf{h}_j\}_{j=1}^M$, the interaction behavior data $\mathbf{x}_i$ can be generated. To sufficiently exploit the preference diversity, $\mathbf{x}_i$ is split into G groups (as shown in Subsection 4.2). For each user, all interaction behaviors $[x_{i,j}^{(g)}]_{1\leq g\leq G, 1\leq j\leq M}$ share the same user latent representation $\mathbf{z}_i$. In this case, the user-item interaction information $\mathbf{x}_i = \{\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}, \cdots, \mathbf{x}_i^{(G)}\}$ with group structure can be generated by:

$$\mathbf{x}_i \sim \prod_{j=1}^{M}\prod_{g=1}^{G} \mathcal{B}\left(x_{i,j}^{(g)}; \frac{\sum_{g=1}^{G} c_{j,g} f_{\theta^{(g)}}\left(\cos(\mathbf{h}_j, \mathbf{z}_i)/\tau + \mathbf{o}_i^{(g)}\mathbf{z}_i^\top\right)}{\sum_{j=1}^{M}\sum_{g=1}^{G} c_{j,g} f_{\theta^{(g)}}\left(\cos(\mathbf{h}_j, \mathbf{z}_i)/\tau + \mathbf{o}_i^{(g)}\mathbf{z}_i^\top\right)}\right) \tag{9}$$

Among them, the user-item interaction behavior $x_{i,j}^{(g)}$ can be decoded with an individual *Bernoulli* distribution. Here function $f_{\theta^{(g)}}(\cdot)$ indicates neural network parameterized by $\theta^{(g)}$ that estimates how much the user with a given preference is interested in item group g. $\cos(\mathbf{h}_j, \mathbf{z}_i)/\tau$ implies that item representation $\mathbf{h}_j$ will be disentangled if $\mathbf{z}_i$ is disentangled, as the two's dimensions are aligned. $\mathbf{o}_i^{(g)}\mathbf{z}_i^\top$ connects the latent user representation and item groups to achieve latent-level disentanglement. Consider the computational complexity, here we use sampled softmax (Jean et al., 2015) to estimate $\sum_{j=1}^{M}\sum_{g=1}^{G} c_{j,g} f_{\theta^{(g)}}\left(\cos(\mathbf{h}_j, \mathbf{z}_i)/\tau + \mathbf{o}_i^{(g)}\mathbf{z}_i^\top\right)$ based on a few sampled items when M is vary large. The whole generative process of InDGRM is summarized in Algorithm 1.

4.5 The Interpretability of InDGRM

The proposed InDGRM model aims to capture the complicated user preference pattern and simultaneously provide interpretable recommendation. The interpretability of InDGRM can be guaranteed from two aspects:

1. *Observed-level disentanglement on users and items*: By introducing mixture prior to model inter-user preference similarity on latent user representation, InDGRM allows dynamic allocation of statistical capacity among components, where users will be assigned into different components (clusters) so that users with similar preference belong to one component. In this case, users in the same component can be taken as neighborhood to each other. Based to this operation, InDGRM is able to capture user correlations structure and achieve observed-level disentanglement on users. Furthermore, this kind of disentanglement alleviates data sparsity by allowing a near cold-start user to borrow information from other users from the same cluster. By introducing prototype-based item grouping to model intra-user preference diversity,

Algorithm 1 InDGRM Generative Process

1. Randomly initialize hyperparameters $\alpha, \eta_a, \eta_b, a, b$, item representations $\{\mathbf{h}_j\}_{j=1}^N$, group representations $\{\mathbf{p}_g\}_{g=1}^G$;
2. For each item $j \in [1, M]$:
    1) For each item group $g \in [1, G]$:
        a) Draw $s_{j,g} \propto \exp\left(\frac{\cos(\mathbf{h}_j, \mathbf{p}_g)}{\tau}\right)$
        b) if privilege information
            i) Draw $s_{j,k,g} \propto \exp\left\{\frac{\cos(\mathbf{h}_j, \mathbf{p}_k)}{\tau} - \frac{\cos(\mathbf{h}_j, \mathbf{p}_g)}{\tau}\right\}$
            ii) Draw $\bar{s}_{j,k,g} \sim \mathcal{B}(s_{j,k,g})$
    2) Draw $\mathbf{c}_j \sim \mathcal{M}ult(N_j, \mathbf{s}_j)$
3. For each user $i \in [1, N]$:
    1) Draw $\boldsymbol{\pi}_i \sim GEM(\alpha)$
    2) For each latent dimension $l \in [1, d^{(k)}]$
        a) Draw $\lambda_l^{(k)} \sim \mathcal{IG}(\eta_a, \eta_b)$
    3) Draw use representation $\mathbf{z}_i \sim \sum_{k=1}^{\infty} \pi_{i,k} \mathcal{N}\left(\boldsymbol{\mu}_i^{(k)}, \boldsymbol{\sigma}_i^{(k)} \mathbf{I}\right) = \sum_{k=1}^{\infty} \pi_{i,k} \prod_{l=1}^{d^{(k)}} \mathcal{N}\left(0, \lambda_l^{(k)}\right)$
    4) For each interaction related to user i:
        a) Draw $\gamma^2 \sim \Gamma\left(\frac{a+1}{2}, \frac{b^2}{2}\right)$
        b) For each item group $g \in [1, G]$ and each latent dimension $l \in [1, d^{(k)}]$:
            i) Draw $o_{i,l}^{(g)} \sim \mathcal{N}(0, \gamma^2)$
        c) Draw the rating $\mathbf{x}_i \sim \prod_{j=1}^{M} \prod_{g=1}^{G} \mathcal{B}\left(x_{i,j}^{(g)}; \frac{\sum_{g=1}^{G} c_{j,g} f_{\theta^{(g)}}\left(\cos(\mathbf{h}_j, \mathbf{z}_i)/\tau + \mathbf{o}_i^{(g)} \mathbf{z}_i^\top\right)}{\sum_{j=1}^{M} \sum_{g=1}^{G} c_{j,g} f_{\theta^{(g)}}\left(\cos(\mathbf{h}_j, \mathbf{z}_i)/\tau + \mathbf{o}_i^{(g)} \mathbf{z}_i^\top\right)}\right)$

InDGRM captures item groups to reflect the diverse interests of single user so as to achieve observed-level disentanglement on items.

2. *Latent-level disentanglement on representations*: By introducing the column-wise latent-to-group sparse mapping matrix from latent user representation to item groups, each latent factor will be automatically associated with item group. The mapping matrix can be taken as a bipartite graph on latent feature set and item group set, from which we can quickly identify the relationships among these two sets. When we have category prior for each group, the latent feature can be directly explained according to the category information of the most related groups. A good by-product of factor-to-group mapping is that the latent feature can be easily disentangled in the latent space. Specially, by penalizing mapping between the latent features and item groups, for each user, we can force the model to learn a latent representation where the correlation among latent features is as small as possible. Furthermore, the cosine similarity measurement between user representation and item representation (in Eq. (9)) also encourages latent-level disentanglement on item representations when their dimensions are aligned well.

In a word, in InDGRM, the observed-level and latent-level disentanglement schemes are properly designed and seamlessly integrated into the recommendation model via a unified generative process, which will enforce each other to improve recommendation performance and achieve model interpretability.

4.6 Inference Learning for InDGRM

Deep generative models target to minimizing certain discrepancy measures between the true data distribution and the generative distribution (Kingma and Welling, 2013; Goodfellow et al., 2014). *Kullback-Leibler* (KL) divergence (equivalently maximizing the marginal data log-likelihood) or *Jensen-Shannon* (JS) divergence are commonly used to estimate the model parameters. However, user behavior data is characterized by few frequently occurring items and a large amount of tail items, where data can be actually characterized by a low dimensional manifold. In this case, thus, we adopt Wasserstein distance (Villani, 2008) to measure the difference between two distributions rather than KL-divergence or JS-divergence, because it has the ability to preserve the transitivity in latent space due to the much weaker topology (Tolstikhin et al., 2018; Liu et al., 2019b). Meanwhile, the generative process can be implemented under the auto-encoder framework from the optimal transport point of view (Tolstikhin et al., 2018; Arjovsky et al., 2017).

Motivated by above techniques, the parameters in InDGRM can be determined by minimizing Wasserstein distance between the ground truth distribution and the model generated distribution on user-item interaction data. It can be approximated by optimizing the following bound:

$$\begin{aligned}\mathcal{L}(\mathbf{x}_i;\theta,\phi,\mathbf{O}) = &\inf_{q_\phi(\mathbf{z}_i,k|\mathbf{x}_i,\mathbf{C})\in\mathcal{Q}_{\mathcal{Z}\times\mathcal{K}}} \mathbb{E}_{p(\mathbf{x}_i)}\mathbb{E}_{q_\phi(\mathbf{z}_i,k|\mathbf{x}_i,\mathbf{C})}[c(\mathbf{x}_i,\hat{\mathbf{x}}_i)] \\ &+ \lambda_r \mathbb{E}_{p(\mathbf{x}_i)}[D\left(q_\phi(\mathbf{z}_i,k|\mathbf{x}_i,\mathbf{C})||p_\theta(\mathbf{z}_i|k)p_\theta(k|\boldsymbol{\pi}_i)\right)] - \lambda_o \sum_g ||\mathbf{o}_i^{(g)}||_2.\end{aligned} \tag{10}$$

where $\mathbf{x}_i$ and $\hat{\mathbf{x}}_i$ are the original and generated feedback data respectively. $c(x, y)$ is the cost function to measure the distance between x and y, here the squared cost function $c(x, y) = ||x - y||_2^2$ is used. $\inf_{q_\phi(\mathbf{z}_i,k|\mathbf{x}_i,\mathbf{C})\in\mathcal{Q}_{\mathcal{Z}\times\mathcal{K}}} \mathbb{E}_{p(\mathbf{x}_i)}\mathbb{E}_{q_\phi(\mathbf{z}_i,k|\mathbf{x}_i,\mathbf{C})}[c(\mathbf{x}_i, \hat{\mathbf{x}}_i)]$ is the tractable upper bound of Wasserstein distance $\mathcal{W}(p(\mathbf{x}_i), p_G(\mathbf{x}_i))$, and $\mathcal{Q}_{\mathcal{Z}\times\mathcal{K}}$ is the set of all conditioned distributions over $\mathbf{z}_i$. Variational distribution $q_\phi(\mathbf{z}_i, k|\mathbf{x}_i, \mathbf{C})$ in the inference model can be factorized as $q_\phi(\mathbf{z}_i|k, \mathbf{x}_i, \mathbf{C})$ and $q_\phi(k|\mathbf{x}_i, \mathbf{C})$. Since any parametrization of $q_\phi(\mathbf{z}_i, k|\mathbf{x}_i, \mathbf{C})$ reduces the search space of the infimum, the above objective function can be taken as an upper bound of the true Wasserstein distance. $D(\cdot)$ is an arbitrary divergence between prior and posterior, two mixture of *Gaussian* distributions, which can be efficiently and effectively approximated (Hershey and Olsen, 2007). The last penalty term on $\mathbf{O}_i$ aims to constrain its values in he collapsed bound and encourage it sparse. $\lambda_r > 0$ and $\lambda_o > 0$ are two hyperparameters.

As we mentioned before, prototype learning can be promoted via introducing privilege information (Eq. (7)). Thus, the final objective can be formulated as follow:

$$\begin{aligned}\mathcal{L}(\mathbf{x}_i; \theta, \phi, \mathbf{O}) = &\inf_{q_\phi(\mathbf{z}_i,k|\mathbf{x}_i,\mathbf{C})\in\mathcal{Q}_{\mathcal{Z}\times\mathcal{K}}} \mathbb{E}_{p(\mathbf{x}_i)}\mathbb{E}_{q_\phi(\mathbf{z}_i,k|\mathbf{x}_i,\mathbf{C})}[c(\mathbf{x}_i, \hat{\mathbf{x}}_i)] \\ &+ \lambda_r \mathbb{E}_{p(\mathbf{x}_i)}[D\left(q_\phi(\mathbf{z}_i, k|\mathbf{x}_i, \mathbf{C})||p_\theta(\mathbf{z}_i|k)p_\theta(k|\boldsymbol{\pi}_i)\right)] - \lambda_o \sum_g ||\mathbf{o}_i^{(g)}||_2 \\ &+ \lambda_p \frac{1}{|\mathcal{P}|} \sum_{(j,k,g)\in\mathcal{P}} -\bar{s}_{j,k,g} \log s_{j,k,g} - (1 - \bar{s}_{j,k,g}) \log(1 - s_{j,k,g}).\end{aligned} \tag{11}$$

where the last term $\lambda_p \frac{1}{|\mathcal{P}|} \sum_{(j,k,g)\in\mathcal{P}} -\bar{s}_{j,k,g} \log s_{j,k,g} - (1-\bar{s}_{j,k,g}) \log(1-s_{j,k,g})$ is derived from Eq. (7) in Section 4.2.2. $\mathcal{P}$ denotes the set of all ranking order triplets for prototype and λ_p is a hyperparameter controlling the effect of prototype enhancing. Intuitively, this term optimizes for hyperspherical prototypes to have the same ranking order as the semantic priors.

The second term with KL divergence in Eq. (10) and Eq. (11) can be rewritten as

$$\begin{aligned}&\mathbb{E}_{p(\mathbf{x}_i)}\left[D(q_\phi(\mathbf{z}_i, k|\mathbf{x}_i, \mathbf{C})||p_\theta(\mathbf{z}_i|k)p_\theta(k|\boldsymbol{\pi}_i))\right] \\ &= \mathbb{E}_{p(\mathbf{x}_i)}\left[\mathbb{E}_{q_\phi(\mathbf{z}_i,k|\mathbf{x}_i,\mathbf{C})}\left[\ln \frac{q_\phi(\mathbf{z}_i, k|\mathbf{x}_i, \mathbf{C})}{q_\phi(\mathbf{z}_i, k|\mathbf{C})} + \ln \frac{q_\phi(\mathbf{z}_i, k|\mathbf{C})}{p_\theta(\mathbf{z}_i|k)p_\theta(k|\boldsymbol{\pi}_i)}\right]\right] \\ &= \mathbb{E}_{p(\mathbf{x}_i)}[D(q_\phi(\mathbf{z}_i, k|\mathbf{x}_i, \mathbf{C})||q_\phi(\mathbf{z}_i, k|\mathbf{C}))] + \mathbb{E}_{q_\phi(\mathbf{z}_i|\mathbf{x}_i,\mathbf{C})}\left[\ln \frac{q_\phi(\mathbf{z}_i, k|\mathbf{C})}{p_\theta(\mathbf{z}_i|k)p_\theta(k|\boldsymbol{\pi}_i)}\right] \\ &= I(\mathbf{x}_i, \mathbf{z}_i) + D(q_\phi(\mathbf{z}_i, k|\mathbf{C})||p_\theta(\mathbf{z}_i|k)p_\theta(k|\boldsymbol{\pi}_i)).\end{aligned} \tag{12}$$

$I(\mathbf{x}_i, \mathbf{z}_i)$ is the mutual information between $\mathbf{x}_i$ and $\mathbf{z}_i$. Minimizing it is equivalent to applying the information bottleneck principle on the learning process, which encourages $\mathbf{z}_i$ to ignore as much noise in the input as it can, so that the latent representation mainly focus on the essential information. The second term can encourage independence among latent features with the aid of proper prior. For instance, a prior with independent features, $p_\theta(\mathbf{z}_i|k) = \prod_{l=1}^{d} p_\theta(z_{i,l}|k)$ (as shown in Eq.(5)) is adopted here. Penalizing this KL term will make the posterior approach to the prior and preserve such independent properties. Meanwhile, this term seamlessly integrate preference diversity modeling and user correlation modeling.

As mentioned before, to capture the inter-user preference similarity, each user latent representation is assumed following a Mixture of *Gaussian* distribution. By sufficiently exploiting the relationship between user clusters and item groups, the variational posterior $q_\phi(\mathbf{z}_i|k,\mathbf{x}_i,\mathbf{C})$ can be approximated by

$$q_\phi(\mathbf{z}_i|k,\mathbf{x}_i,\mathbf{C}) = \sum_{k=1}^{K} \tilde{\pi}_{i,k}\mathcal{N}\left(\boldsymbol{\mu}_i^{(\phi,k)}, \boldsymbol{\sigma}_i^{(\phi,k)}\mathbf{I}\right). \tag{13}$$

In mixture model, each component is a *Gaussian* distribution with the following mean $\boldsymbol{\mu}_i^{(\phi,k)}$ and standard deviation $\boldsymbol{\sigma}_i^{(\phi,k)}$:

$$\begin{aligned}
\boldsymbol{\mu}_i^{(\phi,k)} &= \mathbf{W}_\mu^{(k)}\left[\frac{\sum_{j=1}^{M}\sum_{g=1}^{G} c_{j,g}\mathbf{h}_j}{\sqrt{\sum_{j=1}^{M}\sum_{g=1}^{G} c_{j,g}}}, f_\phi\left(\{\mathbf{x}_i^{(g)}\}_{g=1}^{G}\right)\right], \\
\boldsymbol{\sigma}_i^{(\phi,k)} &= \text{softplus}\left(\mathbf{W}_\sigma^{(k)}\left[\frac{\sum_{j=1}^{M}\sum_{g=1}^{G} c_{j,g}\mathbf{h}_j}{\sqrt{\sum_{j=1}^{M}\sum_{g=1}^{G} c_{j,g}}}, f_\phi\left(\{\mathbf{x}_i^{(g)}\}_{g=1}^{G}\right)\right]\right),
\end{aligned} \tag{14}$$

where $\mathbf{W}_\mu^{(k)}$ and $\mathbf{W}_\sigma^{(k)}$ are learnable weight matrices related to the k-th component. $f_\phi(\cdot)$ is the neural network to extract the high-level features from user behavior information with group structure. $\{\tilde{\pi}_{i,k}\}_{k=1}^{K}$ are the group-specific attention weights, which can be computed via:

$$\tilde{\pi}_{i,k} = \text{softmax}\left(\mathbf{W}_2\text{tanh}\left(\mathbf{W}_1\left[\mathbf{x}_i, \{\mathbf{x}_{i'}\}_{i'\in\mathcal{U}_k}, \{\mathbf{h}_j\}_{x_{ij}=1}\right]\right)\right). \tag{15}$$

Here $\mathcal{U}_k$ indicates users set related to the k-th user cluster and $\mathbf{h}_j$ is representation for the j-th item. $\mathbf{W}_1$ and $\mathbf{W}_2$ are learnable weight metrices.

To mirror the prior, we parameterize variational distribution $q_\phi(k|\mathbf{x}_i,\mathbf{C})$ via mixture prior $p_\theta(\mathbf{z}_i|k)$ $p_\theta(k|\boldsymbol{\pi}_i)$. Specifically,

$$q_\phi(k|\mathbf{x}_i,\mathbf{C}) = \frac{\pi_{i,k}\mathcal{N}\left(\boldsymbol{\mu}_i^{(k)}, \boldsymbol{\sigma}_i^{(k)}\mathbf{I}\right)}{\sum_{k'=1}^{K}\pi_{i,k'}\mathcal{N}\left(\boldsymbol{\mu}_i^{(k')}, \boldsymbol{\sigma}_i^{(k')}\mathbf{I}\right)} \tag{16}$$

is a categorical distribution over K user clusters. The information loss induced by the factorized approximation can be mitigated by forcing its dependency on the posterior $p_\theta(k|\mathbf{z}_i)$ and non-formative prior $p_\theta(k|\boldsymbol{\pi}_i)$.

For the group lasso penalty on the latent-to group matrix $\mathbf{O}_i$ in Eq.(10) or Eq.(11), we update it via proximal gradient descent method, which is efficient for the separable objectives with both differentiable and potentially non-differentiable components. Specifically, the proximal operator on vector $\lambda_o\sum_g||\mathbf{o}_i^{(g)}||_2$ can be formulated as:

$$\mathbf{o}_i^{(g)} = \frac{\mathbf{o}_i^{(g)}}{||\mathbf{o}_i^{(g)}||_2}\max\left(0, ||\mathbf{o}_i^{(g)}||_2 - \eta_o\lambda_o\right).$$

Geometrically, this operator reduces the norm of $\mathbf{o}_i^{(g)}$ by $\eta_o\lambda_o$, and shrinks $\mathbf{o}_i^{(g)}$ with $||\mathbf{o}_i^{(g)}||_2 \leq \eta_o\lambda_o$ to zero. Obviously, this updating is cheap for computing and leads to machine-precision zeros.

4.7 Local Variational Optimization for InDGRM

In the objective function, Eq.(10) or Eq.(11), $\{\theta, \phi, \mathbf{O}\}$ are the parameters to be optimized. Among it, the latent representation $\mathbf{z}_i$ can be obtained via reparametrization trick, $\mathbf{z}_i = \boldsymbol{\mu}_i^{(\phi,g)} + \epsilon_n \odot \boldsymbol{\sigma}_i^{(\phi,g)}$, where $\epsilon_n \sim \mathcal{N}(0, \mathbf{I})$. Thus, the posterior is be controlled by following variational parameters:

$$\psi(\mathbf{x}_i) = \left\{ \{\boldsymbol{\mu}_i^{(\phi,g)}\}_{g=1}^G, \{\boldsymbol{\sigma}_i^{(\phi,g)}\}_{g=1}^G \right\}.$$

In this case, given $\mathbf{x}_i$, our main goal is to find the optimal variational parameters $\psi(\mathbf{x}_i)$, which can be implemented by maximizing $\mathcal{L}(\mathbf{x}_i; \theta, \phi, \mathbf{O}, \psi(\mathbf{x}_i))$ on the neural network with parameters ϕ.

A simple and popular way is random sampling $\psi(\mathbf{x}_i)$. However, it has been proven that such strategy can not guarantee optimal variational parameters and may yield a much looser upper bound for the original generative model designed on Wasserstein distance (between prior and posterior distributions) (Krishnan et al., 2017). Therefore, in this work, the optimal variational parameters ψ^* are automatically determined, which has ability to obtain a tighter upper bound $\mathcal{L}(\mathbf{x}_i; \theta, \phi, \mathbf{O}, \psi^*(\mathbf{x}_i))$ as follows.

$$\mathcal{L}(\mathbf{x}_i; \theta, \phi, \mathbf{O}) \geq \mathcal{L}(\mathbf{x}_i; \theta, \phi, \mathbf{O}, \psi^*(\mathbf{x}_i)) \geq \mathcal{W}(p(\mathbf{x}_i), p_G(\mathbf{x}_i)) \tag{17}$$

Algorithm 2 Learning with local variational optimization for InDGRM

Input: Implicit user-item interaction matrix $\mathbf{X}$ with N users and M items. Randomly initialized parameters $\{\theta, \phi, \mathbf{O}\}$.

1: **for** $t = 1, 2, \cdots, T$
2: Sample a user $\mathbf{x}_i$ and set $\psi_0 = \psi(\mathbf{x}_i)$.
3: Learning item group assignment $\mathbf{C}$ and prototypes via Eq.(6) and Eq.(7).
4: Approximate $\psi^L \approx \psi^*(\mathbf{x}_i) = \min_\psi \mathcal{L}_G(\mathbf{x}_i; \theta^t, \phi^t, \mathbf{O}^t, \psi(\mathbf{x}_i))$.
5: **for** $\tau = 0, \cdots, L-1$
6: $\psi^{\tau+1} = \psi^\tau + \eta_\psi \nabla_{\psi^\tau} \mathcal{L}_G(\mathbf{x}_i; \theta^t, \phi^t, \mathbf{O}^t, \psi^\tau)$.
7: Update θ: $\theta^{t+1} \leftarrow \theta^t + \eta_\theta \nabla_{\theta^t} \mathcal{L}(\mathbf{x}_i; \theta^t, \phi^t, \mathbf{O}^t, \psi^L)$.
8: Update ϕ: $\phi^{t+1} \leftarrow \phi^t + \eta_\phi \nabla_{\phi^t} \mathcal{L}(\mathbf{x}_i; \theta^t, \phi^t, \mathbf{O}^t, \psi^L)$.
9: Update $\mathbf{O}$: $\mathbf{O}^{t+1} \leftarrow \mathbf{O}^t + \eta_o \nabla_{\mathbf{O}^t} \mathcal{L}(\mathbf{x}_i; \theta^t, \phi^t, \mathbf{O}^t, \psi^L)$.
10: **for** $g = 1, 2, \cdots, G$
11: Update $\mathbf{o}_i^{(g)}$ with $\mathbf{o}_i^{(g)} = \frac{\mathbf{o}_i^{(g)}}{||\mathbf{o}_i^{(g)}||_2} \max\left(0, ||\mathbf{o}_i^{(g)}||_2 - \eta_o \rho\right)$.
12: Reduce certain dimension of $\mathbf{z}_i$ if corresponding ARD variance is less than ϵ_λ.

Output: The learned optimal parameters $\{\theta, \phi, \mathbf{O}\}$.

To obtain the optimal ψ^* for further updating model parameters $\{\theta, \phi, \mathbf{O}\}$, we adopt a local variational optimization (LVO) method to estimate parameters $\psi = \psi(\mathbf{x}_i)$ with the aid of the inference network. Its main idea is to optimize ψ via gradient descent, where one initializes with ψ^0 and takes successive steps of $\psi^{\tau+1} = \psi^\tau + \eta_\psi \nabla_{\psi^\tau} \mathcal{L}(\mathbf{x}_i; \theta, \phi, \mathbf{O}, \psi^\tau)$, where the gradient with respect to ψ can be approximated via Monte Carlo sampling. As shown in line 5-7 of Algorithm 2, the resulting ψ^L can approach the optimal variational

parameters. Note that L in inner optimization is a predefined value, usually few steps are enough to find the optimal ψ. With ψ^L, we can update the parameters θ, ϕ and $\mathbf{O}$ via stochastic back-propogation, as shown in line 8, 9 and 10, where the learning rates $\eta_\psi, \eta_\theta, \eta_\phi$ and η_o obtained via ADAM (Kingma and Ba, 2014). The factor-to-group mapping vector $\mathbf{o}_i^{(g)}$ is updated by the proximal stochastic gradient descent method, as shown in line 11-13. The whole training procedure for InDGRM with local variational optimization strategy is summarized in Algorithm 3.

5. Theoretical Analysis of InDGRM

In this section, we focus on theoretically analyzing the proposed InDGRM model including its generalization error bound and relationship with existing methods. In the first we analyze the generalization error bound of the proposed InDGRM to underscore the characteristics of estimation error in terms of model parameters. Then we prove that InDGRM is superior to the existing recommendation methods.

5.1 Generalization Error Bound

Motivated by Arora et al. (2017) and Lee et al. (2016), the generalization of interaction data generation process can be defined by measuring the difference between generative data distribution $\hat{\mathcal{X}}_p$ and real data distribution $\mathcal{X}_p$. Our main idea is check whether the population distance between $\mathcal{X}_p$ and $\hat{\mathcal{X}}_p$ is close to the empirical distance between the empirical distributions ($\mathcal{X}_e$ and $\hat{\mathcal{X}}_e$).

Definition 1 *For the empirical version of the true distribution ($\mathcal{X}_e$) with N training examples, a generative distribution ($\hat{\mathcal{X}}_e$) generalizes under the distance $d(\cdot,\cdot)$ between distributions with generalization error $\delta_1 > 0$ if the following holds with high probability,*

$$|E_p(\hat{\mathbf{X}}) - E_e(\hat{\mathbf{X}})| \leq \delta_1 \tag{18}$$

where

$$\begin{aligned} E_p(\hat{\mathbf{X}}) &= \mathbb{E}_{\mathbf{x}^{(p)}\sim\mathcal{X}_p, \mathbf{x}^{(p)}\sim\hat{\mathcal{X}}_p} d\left(\mathbf{x}^{(p)}, \hat{\mathbf{x}}^{(p)}\right), \\ E_e(\hat{\mathbf{X}}) &= \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M} d\left(x_{i,j}^{(e)}, \hat{x}_{i,j}^{(e)}\right) \mid x_{i,j}^{(e)} \sim \hat{\mathcal{X}}_e, \hat{x}_{i,j}^{(e)} \sim \hat{\mathcal{X}}_e \end{aligned} \tag{19}$$

$E_p(\hat{\mathbf{X}})$ indicates the population distance between the true and generative distributions ($\mathcal{X}_p$ and $\hat{\mathcal{X}}_p$). $E_e(\hat{\mathbf{X}})$ is the empirical distance between the true and generative distributions ($\mathcal{X}_e$ and $\hat{\mathcal{X}}_e$).

Since the proposed InDGRM with single user cluster and item group can be seem as a generalized matrix completion model, the *Frobenius norm* between the ground truth data ($\mathbf{X}^{(e)}$) and the generative data ($\hat{\mathbf{X}}^{(e)}$) with N users and M items, is used as the metric to establish the error bound of the proposed method, i.e. $d(x_{ij}^{(e)}, \hat{x}_{i,j}^{(e)}) = ||x_{i,j}^{(e)} - \hat{x}_{i,j}^{(e)}||_2^2$. In the

Algorithm 3 The training procedure for InDGRM with local variational optimization strategy.

input: user-item interaction matrix $\mathbf{X} \in \mathbb{R}^{M \times N}$.
parameters: The parameters $\{\theta, \phi, \mathbf{O}\}$ include: user representations $\{\mathbf{z}_i\}_{i=1}^N \in \mathbb{R}^{N \times d_i}$, item representations $\{\mathbf{h}_j\}_{j=1}^M \in \mathbb{R}^{M \times d_i}$, group prototypes $\{\mathbf{p}_g\}_{k=g}^G \in \mathbb{R}^{K \times d_i}$, sparse mapping $\{\mathbf{O}_i\}_{i=1}^N \in \mathbb{R}^{N \times d_i \times G}$, and the parameters of neural networks.
function PrototypeGroupAssignment()
  if privilege information
    $\bar{s}_{j,k,g} = \mathbb{1}(\cos(\mathbf{t}_g, \mathbf{t}_j) \geq \cos(\mathbf{t}_g, \mathbf{t}_k)), \quad (i,j,k) \in \mathcal{P}$
    $s_{j,k,g} \propto \exp\left\{\frac{\cos(\mathbf{h}_j, \mathbf{p}_k)}{\tau} - \frac{\cos(\mathbf{h}_j, \mathbf{p}_g)}{\tau}\right\}$
    $\mathcal{PI} = \frac{1}{|\mathcal{P}|}\sum_{(j,k,g)\in\mathcal{P}} -\bar{s}_{j,k,g}\log s_{j,k,g} - (1-\bar{s}_{j,k,g})\log(1-s_{j,k,g})$
  $s_{j,g} \leftarrow \exp\left(\frac{\cos(\mathbf{h}_j, \mathbf{p}_g)}{\tau}\right), \quad g = 1, 2, \cdots, G$
  $\mathbf{c}_j \sim$ Gumble-Softmax$([s_{j,1}, \cdots, s_{j,G}])$
  return $\{\mathbf{c}_j\}_{j=1}^M, \mathcal{PI}$
function Encoder$(\mathbf{x}_i, \{\mathbf{c}_j\}_{j=1}^M)$
  $\boldsymbol{\mu}_i^{(\phi,k)} = \mathbf{W}_\mu^{(k)}\left[\frac{\sum_{j=1}^M\sum_{g=1}^G c_{j,g}\mathbf{h}_j}{\sqrt{\sum_{j=1}^M\sum_{g=1}^G c_{j,g}}}, f_\phi\left(\{\mathbf{x}_i^{(g)}\}_{g=1}^G\right)\right],$
  $\boldsymbol{\sigma}_i^{(\phi,k)} = \text{softplus}\left(\mathbf{W}_\sigma^{(k)}\left[\frac{\sum_{j=1}^M\sum_{g=1}^G c_{j,g}\mathbf{h}_j}{\sqrt{\sum_{j=1}^M\sum_{g=1}^G c_{j,g}}}, f_\phi\left(\{\mathbf{x}_i^{(g)}\}_{g=1}^G\right)\right]\right)$
  $\tilde{\pi}_{i,k} = \text{softmax}\left(\mathbf{W}_2\text{tanh}\left(\mathbf{W}_1\left[\mathbf{x}_i, \{\mathbf{x}_{i'}\}_{i'\in\mathcal{U}_k}, \{\mathbf{h}_j\}_{x_{ij}=1}\right]\right)\right)$
  $\mathbf{z}_i = \sum_{k=1}^K \tilde{\pi}_{i,k}\mathcal{N}\left(\boldsymbol{\mu}_i^{(\phi,k)}, \boldsymbol{\sigma}_i^{(\phi,k)}\mathbf{I}\right)$
  return $\mathbf{z}_i$, $\mathcal{KL} = \sum_{k=1}^K \mathcal{KL}\left(\mathcal{N}(\boldsymbol{\mu}_i^{(\phi,k)}, \boldsymbol{\sigma}_i^{(\phi,k)}\mathbf{I})||\mathcal{N}(0, \boldsymbol{\Lambda}^{(k)}\right)$
function Decoder$(\mathbf{z}_i, \{\mathbf{c}_j\}_{j=1}^M)$
  $\mathcal{SP} = \sum_g ||\mathbf{o}_i^{(g)}||_2$
  for $g = 1, 2, \cdots, G$ // For simplicity, we omit sup-script (g) and denote $p_{i,j}^{(g)}$ as $p_{i,j}$.
    $p_{i,j} \leftarrow \frac{\sum_{g=1}^G c_{j,g} f_{\theta(g)}\left(\cos(\mathbf{h}_j, \mathbf{z}_i)/\tau + \mathbf{o}_i^{(g)}\mathbf{z}_i^\top\right)}{\sum_{j=1}^M\sum_{g=1}^G c_{j,g} f_{\theta(g)}\left(\cos(\mathbf{h}_j, \mathbf{z}_i)/\tau + \mathbf{o}_i^{(g)}\mathbf{z}_i^\top\right)}, \quad j = 1, 2, \cdots, M$
    $[\hat{x}_{i,1}, \hat{x}_{i,2}, \cdots, \hat{x}_{i,M}] \leftarrow$ Softmax$([p_{i,1}, p_{i,2}, \cdots, p_{i,M}])$
  $\mathcal{RC} = ||\mathbf{x}_i - \hat{\mathbf{x}}_i||_F^2$
  return $\mathcal{RC} + \lambda_o\mathcal{SP}$
$\{\mathbf{c}_j\}_{j=1}^M, \mathcal{PI} \leftarrow$ PrototypeGroupAssignment().
$\mathbf{z}_i$, $KL \leftarrow$Encoder$(\mathbf{x}_i, \{\mathbf{c}_j\}_{j=1}^M)$
$\mathcal{RC} + \lambda_o\mathcal{SP} \leftarrow$ Decoder$(\mathbf{z}_i, \{\mathbf{c}_j\}_{j=1}^M)$
if privilege information
  $\mathcal{L}\left(\mathbf{x}_i; \theta, \phi, \mathbf{O}, \psi^*(\mathbf{x}_i)\right) = \mathcal{RC} + \lambda_r\mathcal{KL} + \lambda_o\mathcal{SP} + \lambda_p\mathcal{PI}$
else
  $\mathcal{L}\left(\mathbf{x}_i; \theta, \phi, \mathbf{O}, \psi^*(\mathbf{x}_i)\right) = \mathcal{RC} + \lambda_r\mathcal{KL} + \lambda_o\mathcal{SP}$
$\theta, \phi, \mathbf{O} \leftarrow$ Update θ, ϕ and $\mathbf{O}$ to minimize $\mathcal{L}\left(\mathbf{x}_i; \theta, \phi, \mathbf{O}, \psi^*(\mathbf{x}_i)\right)$, using the proposed local variational optimization method.

following content, we abbreviate $\mathbf{X}^{(e)}$ as $\mathbf{X}$ and $\hat{\mathbf{X}}^{(e)}$ as $\hat{\mathbf{X}}$, and $d\left(x_{i,j}, \hat{x}_{i,j}\right) = ||x_{i,j} - \hat{x}_{i,j}||_2^2$ is used as shorthand for $d(x_{ij}^{(e)}, \hat{x}_{i,j}^{(e)}) = ||x_{i,j}^{(e)} - \hat{x}_{i,j}^{(e)}||_2^2$.

According to the generative process of InDGRM, it can be seen that the whole interaction matrix is generated by $K \times G$ components with different properties. Here we define $\mathbf{X}^{(k,g)}$ as the (k,g)-th submatrix, $N^{(k,g)}$ is the number of users in submatrix $\mathbf{X}^{(k,g)}$ and $M^{(k,g)}$ is the number of items. Firstly, we establish the error bound of the proposed InDGRM, i.e., the generative error of the proposed deep generative recommendation model is bounded, so that we can still find optimal generative model for recommendation by optimizing the proposed optimization problem.

Theorem 2 *For any* $\mathbf{X} \in \mathbb{N}^{N \times M}$ *(*$N, M > 2$*),*

$$E_p(\hat{\mathbf{X}}) \leq E_e(\hat{\mathbf{X}}) + \sqrt{\frac{\log \delta_1}{-2NM} \left(\max_{(i,j)} d_{i,j}\right)^2} \tag{20}$$

holds with probability at least $1 - \delta_1$ *over uniformly choosing an empirical version of* $\mathbf{X}$*. Here,* $d_{i,j} = d\left(x_{i,j}, \hat{x}_{i,j}\right)$*.*

Proof Since the entries in $\mathbf{X}$ are chosen independently and uniformly, it is reasonable to assume each $d_{i,j} = d(x_{i,j}, \hat{x}_{i,j})$ is a random variable and satisfies

$$p(\zeta \geq d_{i,j} \geq 0) = 1$$

where $\zeta = \max_{(i,j)} d_{i,j}$. Hence, based on the Hoeffding Inequality, we have $p(|E_p(\hat{\mathbf{X}}) - E_e(\hat{\mathbf{X}})| \geq \epsilon) \leq \exp\left(\frac{-2NM\epsilon^2}{\zeta^2}\right)$. By setting $\epsilon = \sqrt{\frac{\log \delta_1}{-2NM}\zeta^2}$, we have

$$p\left(|E_p(\hat{\mathbf{X}}) - E_e(\hat{\mathbf{X}})| \leq \sqrt{\frac{\log \delta_1}{-2NM}\zeta^2}\right) \geq 1 - \delta_1 \tag{21}$$

i.e.,

$$p\left(E_p(\hat{\mathbf{X}}) \leq E_e(\hat{\mathbf{X}}) + \sqrt{\frac{\log \delta_1}{-2NM} \left(\max_{(i,j)} d_{i,j}\right)^2}\right) \geq 1 - \delta_1 \tag{22}$$

Therefore, the errors of interaction data generative model are bounded. ■

Then, we theoretically prove the generalization error bound of InDGRM related to a single user cluster and a single item group, $\mathbf{X}^{(k,g)}$. To make the theorem more convincing, we make several standard assumptions: (1) each submatrix $\mathbf{X}^{(k,g)}$ related to the (k)-th user cluster and the (g)-th item group is incoherent (Candès and Recht, 2009; Sun and Luo, 2016); (2) $\mathbf{X}^{(k,g)}$ is well-conditioned (Candès and Recht, 2009); (3) the number of users is larger than items in each submatrix ($N^{(k,g)} \geq M^{(k,g)}$). As shown in Theorem 7 in Candes and Plan (2011), a theoretical bound to generalized matrix reconstruction model is existing as follows.

Theorem 3 *If* $\mathbf{X}^{(k,g)}$ *is well-conditioned and incoherent such that* $|\mathcal{D}^{(k,g)}| \geq C\mu^2 M^{(k,g)} d^{(k,g)} \log^6 M^{(k,g)}$, *then with high probability* $1-(M^{(k,g)})^{-3}$, $\mathbf{X}^{(k,g)}$ *satisfies*

$$\begin{aligned} ||\mathbf{X}^{(k,g)} - \hat{\mathbf{X}}^{(k,g)}||_F &\leq 4\varepsilon\sqrt{\frac{(2+\rho^{(k,g)})M^{(k,g)}}{\rho^{(k,g)}}} + 2\varepsilon \\ &= 4\sqrt{\frac{(2+\rho^{(k,g)})M^{(k,g)}}{\rho^{(k,g)}}} + 2 \end{aligned} \tag{23}$$

where $\rho^{(k,g)} = \frac{|\mathbf{X}^{(k,g)}|}{N^{(k,g)}M^{(k,g)}}$ *indicates the density of the observed entries* ρ *in every submatrix is consistent with each other.* $\varepsilon = \max(\hat{\mathbf{X}}) - \min(\hat{\mathbf{X}}) = 1$ *and* $d^{(k,g)}$ *is the optimal dimensionality of latent representation for the* (k,g)*-th submatrix,* $\mathcal{D}^{(k,g)} \subset \mathcal{D}$ *is the set of observed feedback in submatrix* $\mathbf{X}^{(k,g)}$.

This theorem indicates that the generative error of each submatrix $\hat{\mathbf{X}}^{(k,g)}$ is bounded. Then, based on Theorem 3, we can analyze the generalization error bound of the proposed InDGRM method on whole interaction matrix via the following theorem.

Theorem 4 *If matrix related to* K *user clusters and* G *item groups satisfied Theorem 3, then with high probability* $1-\delta_2$, $\hat{\mathbf{X}}$ *is divided into* $K \times G$ *submatrices satisfies*

$$p\left(||\mathbf{X} - \hat{\mathbf{X}}||_F \leq \frac{\omega}{\sqrt{NM}}\left(4\sqrt{\frac{(2+\rho)NKG}{\rho}} + 2KG\right)\right) \geq 1-\delta_2 \tag{24}$$

where $\delta_2 = (2KGM)^{-3}$ *and* ω *indicates the average weight,* $\rho = \frac{|\mathbf{X}|}{NM}$ *is the density of the observed entries in* $\mathbf{X}$.

Proof For each user-item pair (i,j), an implicit feedback $x_{i,j}$ is equal to $\hat{x}_{i,j} + z$ where z is a random variable whose absolute error is bounded by

$$||W \otimes (\mathbf{X} - \hat{\mathbf{X}})||_\infty \leq \omega(\max(\mathbf{X}) - \min(\mathbf{X})) = \omega\varepsilon = \omega \tag{25}$$

where W indicates the mixture weights. $\varepsilon = \max(\mathbf{X}) - \min(\mathbf{X}) = 1$ since we focus on implicit feedback data. By applying Theorem 3 to implicit preference reconstruction problem with bounded noise, we get with probability greater than $1 - v^{-3}$ that every mixture model to approximate $\mathbf{X}$ will satisfy

$$||W \otimes (\mathbf{X} - \hat{\mathbf{X}})||_F \leq \frac{\omega}{\sqrt{NM}}\left(4\sqrt{\frac{\gamma(2+\rho)}{\rho}} + 2\right) \tag{26}$$

where $v = \max(N,M)$, $\gamma = \min(N,M)$. For a submatrix, there are $K \times G$ different components $\mathbf{X}^{(k,g)}$ and obviously $\sum_k \sum_g \gamma^{(k,g)} \leq N$. Using Cauchy-Schwarz inequality, we get

$$\sum_{(k,g)} \sqrt{\gamma^{(k,g)}} \leq \sqrt{KG \sum_{(k,g)} \gamma^{(k,g)}} \leq \sqrt{KGN} \tag{27}$$

Therefore, we can bound the reconstruction error as follow:

$$\begin{aligned} ||W \otimes (\mathbf{X} - \hat{\mathbf{X}})||_F &\overset{(a)}{\leq} \sum_{(k,g)} ||W^{(k,g)} \otimes (\mathbf{X}^{(k,g)} - \hat{\mathbf{X}}^{(k,g)})||_F \\ &\overset{(b)}{\leq} \sum_{(k,g)} \frac{\omega}{\sqrt{NM}} \left(4\sqrt{\frac{\gamma^{(k,g)}(2+\rho)}{\rho}} + 2 \right) \\ &\overset{(c)}{\leq} \frac{\omega}{\sqrt{NM}} \left(4\sqrt{\frac{2KGN(2+\rho)}{\rho}} + 2KG \right) \end{aligned} \tag{28}$$

in which (a) holds due to the triangle inequality of Frobenius norm; and (b) holds due to (26); and (c) holds due to (27). Since for all user-item pairs, we have

$$E_e(\hat{\mathbf{X}}) = \frac{||\mathbf{X} - \hat{\mathbf{X}}||_F}{\sqrt{NM}} \leq \frac{||W \otimes (\mathbf{X} - \hat{\mathbf{X}})||_F}{\sqrt{NM}} \tag{29}$$

Combining (28) and (29), we established the error bound of $\hat{\mathbf{X}}$ as stated above. In order to adjust the confidence level, we take a union bound of events $||W \otimes (\mathbf{X} - \hat{\mathbf{X}})||_F \geq \frac{\omega}{\sqrt{NM}} \left(4\sqrt{\frac{\gamma(2+\rho)}{\rho}} + 2 \right)$ for each mixture component, then we have

$$\sum_{(k,g)} \left(v^{(k,g)} \right)^{-3} \geq \left(KG \sum_{(k,g)} v^{(k,g)} \right)^{-3} \geq (2KGM)^{-3} \tag{30}$$

thus, the error bound holds with probability at least $1 - (2KGM)^{-3}$. ■

Remark 5 *According to above theorems, we can underscore the characteristics of estimation error in terms of parameters such as the training set size of each submatrix $|\mathcal{D}^{(k,g)}|$, optimal dimensions of each user representation d_i, and the number of submatrix $K \times G$. Considering the basic assumption of each submatrix is followed by Candes and Plan (2011), our model with K user clusters and G item groups requires a lower training set size, i.e., $\sum_{(k,g)} |\mathcal{D}^{(k,g)}| \leq |\mathcal{D}|$. This property also guarantees that our model is able to achieve competitive performance in more sparse scenario. And we observe empirically that this phenomenon does occur with cold-start scenario (see Table 3). Furthermore, our method is able to model user representations in each submatrix with the optimal subspace dimensions since the introduction of ARD technique, which further relaxes the requirement of the size of training set.*

5.2 Relations to Existing Methods

Generative model aims to reconstruct the input interaction data including the observed data and the unobserved data. In this scenario, the proposed method is closely related to global-based methods (e.g., PMF (Mnih and Salakhutdinov, 2008)) and local-based methods

(e.g., LLORMA (Lee et al., 2016), DGLGM (Liu et al., 2020)). In this section, we will theoretically compare them from the perspective of generalization bound.

The objective function of the traditional matrix reconstruction methods (Mnih and Salakhutdinov, 2008) can be formulated as follow:

$$\mathcal{L}_g = \frac{1}{|\mathcal{D}|} \sum_{(i,j)\in\mathcal{D}} ||x_{i,j} - \hat{x}_{i,j}||_2^2. \tag{31}$$

Here we ignore the regularization term. For local-based matrix reconstruction method with S local structures (Lee et al., 2016), its objective function can be written

$$\mathcal{L}_l = \sum_{s=1}^{S} \frac{w_s}{|\mathcal{D}^{(s)}|} \sum_{(i,j)\in\mathcal{D}^{(s)}} ||x_{i,j}^{(s)} - \hat{x}_{i,j}^{(s)}||_2^2, \tag{32}$$

where $\mathcal{D}^{(s)}$ is the observed entries set related to the s-th local structure and w_s indicates the importance of the corresponding local structure with $\sum_{s=1}^{S} w_s = 1$ and the interactions in the same local structure have the same weight. Next, let's prove that the local-based method (Eq. (32)) can achieve lower generalization error bound than global-based method (Eq. (31)), i.e., the later has ability to obtain potentially better generalization performance than the former.

Theorem 6 *Let $\mathbb{E}[\mathcal{L}_g] = \mathbb{E}[\mathcal{L}_l] = \mu$. For any $\epsilon > 0$, if $p(|\mathcal{L}_l - \mu| \le \epsilon) \ge 1 - \delta_l$ and $p(|\mathcal{L}_g - \mu| \le \epsilon) \ge 1 - \delta_g$, then $\delta_l \le \delta_g$.*

Proof Based on Markov's inequality[6] and Hoeffding's Lemma[7], for any $\epsilon, t > 0$, we have

$$\begin{aligned} p(\mathcal{L}_g - \mu \ge \epsilon) &= p\left(e^{t(\mathcal{L}_g - \mu)} \ge e^{t\epsilon}\right) \\ &\overset{\text{(a)}}{\le} \frac{\mathbb{E}[e^{t(\mathcal{L}_g - \mu)}]}{e^{t\epsilon}} \\ &\overset{\text{(b)}}{\le} \frac{\frac{1}{8}t^2(a-b)^b}{e^{t\epsilon}}, \end{aligned} \tag{33}$$

where μ is the expectation of $\mathcal{L}_g$, $a = \sup\{\mathcal{L}_g - \mu\}$ and $b = \inf\{\mathcal{L}_g - \mu\}$. (a) holds due to Markov's inequality, and (b) holds due to Hoeffding's Lemma. Similarly, we have

$$p(\mu - \mathcal{L}_g \ge \epsilon) \le \frac{e^{\frac{1}{8}t^2(a-b)^2}}{e^{t\epsilon}}. \tag{34}$$

Combing the above two inequalities, we have

$$p(|\mu - \mathcal{L}_g| \le \epsilon) \ge 1 - \frac{2e^{\frac{1}{8}t^2(a-b)^2}}{e^{t\epsilon}}, \tag{35}$$

6. (Markov's Inequality) Let x be a real-valued non-negative random variable. Then, for any $\epsilon > 0$, $p(x \ge \epsilon) \le \frac{\mathbb{E}[x]}{\epsilon}$.
7. (Hoeffding's Lemma) Let x be a real-valued random variable with zero mean and $p(x \in [a, b]) = 1$. Then, for any $z \in \mathbb{R}$, $\mathbb{E}[e^{zx}] \le \exp(\frac{1}{8}z^2(b-a)^2)$.

i.e.,

$$\delta_g = \frac{2e^{\frac{1}{8}t^2(a-b)^2}}{e^{t\epsilon}}. \tag{36}$$

Let $\mathcal{L}_g^{(s)} = \frac{1}{\mathcal{D}^{(s)}} \sum_{(i,j)\in\mathcal{D}^{(s)}} ||x_{i,j} - \hat{x}_{i,j}||_2^2$ ($s \in \{1, 2, \cdots, S\}$). Then, we know that $\mathcal{L}_l$ and $\mathcal{L}_g$ have the same expectation μ, Therefore, we can derive δ_l for $\mathcal{L}_l$ as follows:

$$\begin{aligned} p\left(\sum_{s=1}^{S} w_s \mathcal{L}_l^{(s)} - \mu \geq \epsilon\right) &\overset{\text{(a)}}{\leq} \frac{\mathbb{E}[e^{t(\sum_s w_s \mathcal{L}_l^{(s)} - \mu)}]}{e^{t\epsilon}} \\ &\overset{\text{(b)}}{\leq} \sum_s \frac{w_s \mathbb{E}[e^{t(\mathcal{L}_l^{(s)} - \mu)}]}{e^{t\epsilon}} \\ &\overset{\text{(c)}}{\leq} \sum_s \frac{w_s e^{\frac{1}{8}t^2(a_s - b_s)^2}}{e^{t\epsilon}}, \end{aligned} \tag{37}$$

where (a) holds due to $\frac{\mathbb{E}[e^{t(\sum_s w_s \mathcal{L}_l^{(s)} - \mu)}]}{e^{t\epsilon}} = \frac{\mathbb{E}[e^{t(\sum_s w_s (\mathcal{L}_l^{(s)} - \mu))}]}{e^{t\epsilon}}$ and Markov's inequality, (b) holds due to the convexity of exponential function, and (c) holds due to Hoeffding's Lemma. Let $a_s = \sup\{\mathcal{L}_l^{(s)} - \mu\}$ and $b_s = \inf\{\mathcal{L}_l^{(s)} - \mu\}$, we have

$$\delta_l = 2\sum_s \frac{w_s e^{\frac{1}{8}t^2(a_s - b_s)^2}}{e^{t\epsilon}}. \tag{38}$$

Considering $\mathcal{D}^{(s)} \subset \mathcal{D}$, we know $\sup\{\mathcal{L}_l^{(s)}\} \leq \sup\{\mathcal{L}_g\}$ and $\inf\{\mathcal{L}_l^{(s)}\} \geq \inf\{\mathcal{L}_g\}$. Therefore, $\sup\{\mathcal{L}_l^{(s)} - \mu\} \leq \sup\{\mathcal{L}_g - \mu\}$ and $\inf\{\mathcal{L}_l^{(s)} - \mu\} \geq \inf\{\mathcal{L}_g - \mu\}$, i.e., $a_s \leq a$ and $b_s \geq b$. Then, we have $(a-b)^2 \geq (a_s - b_s)^2$, i.e.,

$$\frac{e^{\frac{1}{8}t^2(a_s - b_s)^2}}{e^{t\epsilon}} \leq \frac{e^{\frac{1}{8}t^2(a-b)^2}}{e^{t\epsilon}} \quad \text{for} \quad \forall s \in \{1, \cdots, S\}. \tag{39}$$

Then, under the constraint $\sum_s w_s = 1$, we can conclude that $\delta_l \leq \delta_g$. ■

Remark 7 *The above theorem demonstrates that $\mathcal{L}_l$ will be close to its expectation μ with higher probability than $\mathcal{L}_g$. Note that Theorem 6 can be applied to general local-based matrix reconstruction methods. However, different local structure determination strategies can derive different δ_l, i.e., a well-designed local structure determination strategy can achieve better generalization performance than random splitting because a well-designed local structure determination strategy can achieve more accurate submatrix partition.*

Similar to the local-based method, we can rephrase the objective function of the proposed InDGRM method by

$$\mathcal{L} = \frac{1}{|\mathcal{D}|} \sum_{(i,j)\in\mathcal{D}} \sum_{s=1}^{S} \frac{w_s}{|\mathcal{D}^{(s)}|} \sum_{(i,j)\in\mathcal{D}^{(s)}} w_{i,j}^{(s)} ||x_{i,j}^{(s)} - \hat{x}_{i,j}^{(s)}||_2^2 \tag{40}$$

where w_s is the local-specific weight. Similarly, all weights are constrained by $\sum_{s=1}^{S} w_s = 1$. $w_{i,j}^{(s)}$ indicates the entry-specific weight in the s-local structure. We can prove that $\mathcal{L}$ can achieve lower generalization error bound in the following Theorem.

Theorem 8 *Let $\mathbb{E}[\mathcal{L}] = \mathbb{E}[\mathcal{L}_l] = \mu$. For any $\epsilon > 0$, if $p(|\mathcal{L}_l - \mu| \leq \epsilon) \geq 1 - \delta_l$ and $p(|\mathcal{L} - \mu| \leq \epsilon) \geq 1 - \delta$, then $\delta \leq \delta_l$.*

Proof According to the definition of $\mathcal{L}$ and $\mathcal{L}^{(s)}$, we can get $\mathbb{E}[\mathcal{L}] = \mathbb{E}[\mathcal{L}_l] = \mu$.

Let $\mathcal{L}^{(s)} = \sum_{(i,j)\in\mathcal{D}^{(s)}} w_{i,j}^{(s)} ||x_{i,j}^{(s)} - \hat{x}_{i,j}^{(s)}||_2^2$ ($s \in \{1, 2, \cdots, S\}$). We can derive δ for $\mathcal{L}$ as follows:

$$\begin{aligned} p\left(\sum_{\mathcal{D}} \sum_{s\in\{1,2,\cdots,S\}} w_s \mathcal{L}^{(s)} - \mu \geq \epsilon\right) &\overset{(a)}{\leq} \frac{\mathbb{E}[e^{t(\sum_s w_s \mathcal{L}^{(s)} - \mu)}]}{e^{t\epsilon}} \\ &\overset{(b)}{\leq} \sum_s \frac{w_s \mathbb{E}[e^{t(\mathcal{L}^{(s)} - \mu)}]}{e^{t\epsilon}} \\ &\overset{(c)}{\leq} \sum_s \frac{w_s e^{\frac{1}{8}t^2(c_s - d_s)^2}}{e^{t\epsilon}}, \end{aligned} \tag{41}$$

where inequalities (a), (b) and (c) hold due to the Markov's inequality, the convexity of exponential function, and Hoeffding's Lemma, respectively. Let $c_s = \sup\{\mathcal{L}^{(s)} - \mu\}$ and $d_s = \inf\{\mathcal{L}^{(s)} - \mu\}$, we have

$$\delta = 2\sum_s \frac{w_s e^{\frac{1}{8}t^2(c_s - d_s)^2}}{e^{t\epsilon}}. \tag{42}$$

According to Theorem 6, we know $\sup\{\mathcal{L}^{(s)}\} \leq \sup\{\mathcal{L}_l^{(s)}\}$ and $\inf\{\mathcal{L}^{(s)}\} \geq \inf\{\mathcal{L}_l^{(s)}\}$. Therefore, $\sup\{\mathcal{L}^{(s)} - \mu\} \leq \sup\{\mathcal{L}_l^{(s)} - \mu\}$ and $\inf\{\mathcal{L}^{(s)} - \mu\} \geq \inf\{\mathcal{L}_l^{(s)} - \mu\}$, i.e., $c_s \leq a_s$ and $d_s \geq b_s$. Then, we have $(a_s - b_s)^2 \geq (c_s - d_s)^2$, i.e.,

$$\frac{e^{\frac{1}{8}t^2(c_s - d_s)^2}}{e^{t\epsilon}} \leq \frac{e^{\frac{1}{8}t^2(a_s - b_s)^2}}{e^{t\epsilon}} \quad \text{for} \quad \forall s \in \{1, \cdots, S\}. \tag{43}$$

Then, under the constraint $\sum_s w_s = 1$, we can conclude that $\delta < \delta_l$. ■

Remark 9 *Theorem 8 indicates that our InDGRM have a lower generalization bound than the existing local-based matrix reconstruction methods, because $\mathcal{L}$ will be close to its expectation μ with higher probability than $\mathcal{L}_l$. Therefore, minimizing $\mathcal{L}$ can achieve small generalization error with higher probability than minimizing $\mathcal{L}_l$.*

6. Experimental Results

In this section, we evaluate the proposed deep generative model on four datasets by comparing with the state-of-the-art recommendation methods.

	ML 20M	*Netflix*	*AliShop-7C*	*Yelp*
♯ users (N)	138,493	480,189	10,668	1,182,626
♯ items (M)	26,744	17,770	20,591	156,638
♯ Interactions	20,000,263	100,000,000	767,493	4,731,265
Density	0.54%	1.17%	0.35%	0.0026%
$\overline{M}_i$	144	208	47	4
$\overline{N}_j$	748	5,627	231	30

$\overline{M}_i$: the average number of items rated by each user
$\overline{N}_j$: the average number of users interested in each item

Table 1: Summary of experimental datasets

6.1 Experimental Setting

Datasets: In experiments, four widely-used recommendation datasets, *MovieLens 20M (ML 20M)* [8], *Netflix* [9], *AliShop-7C* [10] and *Yelp* [11], are used to validate the recommendation performance. Among them, *ML 20M* (Harper and Konstan, 2015) and *Netflix* come from movie domain, *Alishop-7C* (Ma et al., 2019) belongs to online product domain, and *Yelp* is related to local business domain. The preference scores in *ML 20M* are 10 discrete numerical values in the range of [0.5,5] with step 0.5, while the ratings in other datasets are ordinal values on the scale 1 to 5. Following Liang et al. (2018), we binarize explicit data by keeping ratings of four or higher. More detailed information is summarized in Table 1.
Metrics for ranking estimation: Recall

$$\text{Recall@}n(i) = \frac{|\mathcal{R}e(i) \cap \mathcal{T}(i)}{|\mathcal{T}(i)|}$$

is used as ranking estimation, where $\mathcal{R}e(i)$ denotes the set of recommended items to user i and $\mathcal{T}(i)$ denotes the set of favorite items of user i. Meanwhile, the normalized discounted cumulative gain (NDCG) is adopted to measure the item ranking accuracy, which can be computed by:

$$\text{NDCG@}n(i) = \frac{\text{DCG@}n(i)}{\text{IDCG@}n(i)} \qquad \text{DCG@}n(i) = \sum_{r=1}^{n} \frac{2^{rel_r} - 1}{\log_2(r+1)}$$

where IDCG is the DCG value with perfect ranking. rel_r is the graded relevance of the result at position r.
Metrics for interpretability: Two metrics, Explainability Precision (EP) and Explainability Recall (ER), are adopted to evaluate model interpretability (Abdollahi and Nasraoui, 2017). EP is defined as the proportion of explainable items in the top-n recommendation list relative to the number of recommended (top-n) items for each user. Similar to the recall metric, ER is the proportion of explainable items in the top-n recommendation list relative to the number of all explainable items for a given user. Note that an item j is explainable

8. https://grouplens.org/datasets/movielens/
9. https://www.netflixprize.com
10. https://jianxinma.github.io/disentangle-recsys.html
11. https://www.yelp.com/dataset/challenge

for user i if

$$\mathbb{E}(x_{ij}|j \in \mathcal{N}_p) = x_{ij} \times \frac{\mathcal{N}_p \cap \mathcal{I}_i}{|\mathcal{N}_p|} > \xi,$$

where $\mathcal{N}_p$ is the set of neighbors of item p. For each item, cosine similarity based on interaction behaviors is used to measure the relationship between items, top-50 items are selected as neighbors. $\mathcal{I}_i$ is the set of items that user i interacted with. Following (Abdollahi and Nasraoui, 2016), we set explainable threshold $\xi = 0.01$. Mean EP (MEP) is the average value of explainability precision over all users and mean ER (MER) is the mean explainability recall calculated over all users.

Baselines: Two kinds of recommendation methods are adopted as baselines: Traditional recommendation method (EMF (Abdollahi and Nasraoui, 2016)) and Deep generative recommendation models (Mult-VAE (Liang et al., 2018), MacridVAE (Ma et al., 2019) and DGLGM (Liu et al., 2020)).

Parameter setting: For fair comparison, the number of learnable parameters is set around $2Md_{max}$ for each method, which is equivalent to using d_{max}-dimensional representations for the M items. The initial dimensionality d_{max} is set as 150. The dropout technique (Srivastava et al., 2014) is adopted at the input layer with probability 0.5. The model is trained using Adam (Kingma and Ba, 2014) with batch size of 128 users for 200 epoch on all datasets. The regularization coefficient λ is set to 1.2 for *ML 20M* and *Netflix*, 1.5 for *AliShop-7C* and *Yelp*. λ_o is set to 1 for better disentanglement. For auto-encoder based deep methods (Mult-VAE, MacriVAE, DGLGM and our method), the hyperparameters are automatically tuned via TPE (Bergstra et al., 2011), which searches the optimal hyperparameter configuration with 200 trials on the validation set.

The held-out users strategy and five-fold cross validation are used to evaluate the recommendation performance. The 20% users are taken as held-out users and evenly separated for validation and test respectively. For each held-out user, his/her feedback data is randomly split into five equal sized subsets. Among them, four subsets are used to obtain the latent representation, and the rest subset is for evaluation in each round. Finally, the averaged results on five rounds are reported.

6.2 Results and Discussion

In this section, the proposed InDGRM is sufficiently investigated from seven views. Firstly, InDGRM is compared with baselines from *All Users* view and *Near-cold-start Users* view in terms of recommendation performance.Secondly, the interpretability of InDGRM is evaluated in terms of two metrics. Thirdly, a series of ablation experiments is conducted to illustrate the performance of the proposed model. Fourthly, we demonstrate the observed-level disentanglement on users and items. Fifthly, we analyze the factor-to-group interactions to show the disentanglement of user representations and the interpretability of InDGRM. Sixthly, the disentanglement ability on item representations are investigated to further confirm the interpretability of InDGRM. fFinally, the generalization ability and convergence of the proposed model is empirically analyzed.

Datasets	Methods	Recall					NDCG				
		@1	@5	@10	@50	@100	@1	@5	@10	@50	@100
ML 20M	EMF	0.1011	0.1985	0.3168	0.4662	0.5437	0.1026	0.2128	0.3235	0.3541	0.3764
	Mult-VAE	0.1231	0.2231	0.3320	0.5184	0.5761	0.1211	0.2453	0.3185	0.3833	0.4122
	MacridVAE	0.1322	0.2433	0.3411	0.5233	0.6032	0.1322	0.2633	0.3376	0.3790	0.4122
	DGLGM	0.1442	0.2542	0.3428	0.5205	0.6231	0.1527	0.2876	0.3391	0.3931	0.4198
	InDGRM	**0.1631**	**0.2651**	**0.3458**	**0.5264**	**0.6389**	**0.1723**	**0.3041**	**0.3426**	**0.4152**	**0.4276**
Netflix	EMF	0.1422	0.2411	0.3123	0.4143	0.5772	0.0944	0.1511	0.2234	0.2753	0.3422
	Mult-VAE	0.1675	0.2651	0.3208	0.4232	0.5996	0.1238	0.1782	0.2449	0.2985	0.3689
	MacridVAE	0.1842	0.2782	0.3521	0.4348	0.6181	0.1437	0.1916	0.2584	0.3156	0.3798
	DGLGM	0.1942	0.2953	0.3513	0.4401	0.6331	0.1653	0.2142	0.2669	0.3289	0.3763
	InDGRM	**0.2131**	**0.3153**	**0.3585**	**0.4453**	**0.6542**	**0.1841**	**0.2316**	**0.2695**	**0.3451**	**0.3831**
AliShop-7C	EMF	0.0421	0.0852	0.1152	0.2367	0.3651	0.0782	0.1123	0.1522	0.1842	0.2141
	Mult-VAE	0.0589	0.1032	0.1343	0.2459	0.3856	0.0943	0.1322	0.1632	0.2032	0.2387
	MacridVAE	0.0642	0.1152	0.1544	0.3025	0.3984	0.1085	0.1527	0.1722	0.2231	0.2925
	DGLGM	0.0743	0.1165	0.1585	0.3036	0.4132	0.1231	0.1675	0.1792	0.2431	0.2994
	InDGRM	**0.0895**	**0.1287**	**0.1622**	**0.3067**	**0.4275**	**0.1406**	**0.1873**	**0.1853**	**0.2633**	**0.3021**
Yelp	EMF	0.2322	0.3536	0.4873	0.5589	0.7431	0.4354	0.5642	0.7285	0.7378	0.7531
	Mult-VAE	0.2548	0.3753	0.5322	0.5824	0.7643	0.4558	0.5885	0.7328	0.7551	0.7689
	MacridVAE	0.2773	0.3871	0.5435	0.5903	0.7834	0.4669	0.5973	0.7386	0.7663	0.7746
	DGLGM	0.2697	0.3795	0.5458	0.5893	0.7896	0.4772	0.6085	0.7404	0.7595	0.7734
	InDGRM	**0.2846**	**0.3996**	**0.5496**	**0.5932**	**0.8132**	**0.4875**	**0.6173**	**0.7542**	**0.7762**	**0.7785**

Table 2: Comparisons of different methods in terms of ranking estimation (Recall and NDCG) from *All Users* view.

6.2.1 Recommendation Performance

The first experiment is conducted to compare the proposed InDGRM with the baselines from two views (*All Users* and *Near-cold-start Users*) on four datasets in terms of Recall and NDCG. *All Users* indicates that all users are used as the testing set. *Near-cold-start Users* view means that the users with less than 5 interacted items are involved in the testing set[12]. Table 2 shows the recommendation performance (Recall and NDCG) for *All Users* on four datasets. The best and second results are marked in bold and underline. We can see that deep methods significantly outperform the traditional recommendation approach (EMF) on *All Users*, which indicates that non-linear deep features are beneficial for improving recommendation quality. As expected, InDGRM performs better than all deep generative baselines, which confirms that modeling inter-user preference similarity and intra-user preference diversity in a proper way is beneficial to capture user's intrinsic preference and further improve the prediction accuracy.

The more sparser feedback data is, the more challenging the personalized recommendation task is. In these four datasets, the average number of *Near-cold-start Users* are 1543, 2201, 5310 and 4431526 in *ML 20M*, *Netflix*, *AliShop-7C* and *Yelp* respectively, and they are about 1.85%, 0.46%, 57.4%, 46.7% of all users in the corresponding datasets. It can be seen that *AliShop-7C* and *Yelp* are more sparse, thus we further evaluate the recommendation performance on *Near-cold-start Users*. The results obtained by the proposed model and all baselines are listed in the bottom part of Table 3. It is exciting to see that InDGRM is

12. It is not suitable to set the cold start user with less than 5 interacted items on *yelp* dataset since it is extremely sparse. We define *Near-cold-start Users* on *Yelp* as users with less than 2 interacted items.

Datasets	Methods	Recall					NDCG				
		@1	@5	@10	@50	@100	@1	@5	@10	@50	@100
ML 20M	EMF	0.0843	0.1433	0.2122	0.3431	0.4955	0.0564	0.1003	0.1933	0.2345	0.3154
	Mult-VAE	0.1054	0.1534	0.2390	0.3763	0.5432	0.0675	0.1226	0.1976	0.2557	0.3256
	MacridVAE	0.1251	0.1833	0.2339	0.3843	0.5642	0.0762	0.1323	0.1974	0.2733	0.3412
	DGLGM	0.1346	0.1963	0.2431	0.3816	0.5598	0.0721	0.1296	0.1998	0.2872	0.3314
	InDGRM	**0.1421**	**0.1985**	**0.2498**	**0.3928**	**0.5795**	**0.0898**	**0.1473**	**0.2033**	**0.2965**	**0.3433**
Netflix	EMF	0.0784	0.1342	0.1752	0.2546	0.3421	0.1021	0.1342	0.1852	0.2123	0.2655
	Mult-VAE	0.0931	0.1489	0.1945	0.2794	0.3575	0.1321	0.1489	0.1938	0.2322	0.2921
	MacridVAE	0.1034	0.1543	0.2042	0.2852	0.3678	0.1465	0.1589	0.2052	0.2456	0.3015
	DGLGM	0.1142	0.1698	0.2076	0.2893	0.3801	0.1634	0.1731	0.2136	0.2542	0.3078
	InDGRM	**0.1298**	**0.1846**	**0.2136**	**0.2954**	**0.3988**	**0.1814**	**0.1896**	**0.2144**	**0.2679**	**0.3124**
AliShop-7C	EMF	0.0132	0.0436	0.0731	0.1923	0.2653	0.0236	0.0563	0.0852	0.1432	0.1842
	Mult-VAE	0.0245	0.0576	0.0952	0.2124	0.2781	0.0353	0.0655	0.1023	0.1623	0.2042
	MacridVAE	0.0276	0.0631	0.1052	0.2241	0.2833	0.0403	0.0732	0.1103	0.1673	0.2134
	DGLGM	0.0315	0.0615	0.1084	0.2278	0.2913	0.0514	0.0833	0.1156	0.1765	0.2312
	InDGRM	**0.0397**	**0.0783**	**0.1126**	**0.2337**	**0.3042**	**0.0703**	**0.0984**	**0.1214**	**0.1896**	**0.2363**
Yelp	EMF	0.0232	0.0511	0.0788	0.1231	0.3244	0.0421	0.0873	0.1122	0.1766	0.2456
	Mult-VAE	0.0341	0.0631	0.0981	0.1328	0.3433	0.0542	0.0967	0.1352	0.1879	0.2642
	MacridVAE	0.0421	0.0745	0.1031	0.1398	0.3542	0.0672	0.1042	0.1428	0.1982	0.2762
	DGLGM	0.0542	0.0872	0.1121	0.1472	0.3648	0.0732	0.1101	0.1531	0.2098	0.2892
	InDGRM	**0.0673**	**0.0993**	**0.1289**	**0.1628**	**0.3761**	**0.0894**	**0.1315**	**0.1672**	**0.2214**	**0.3015**

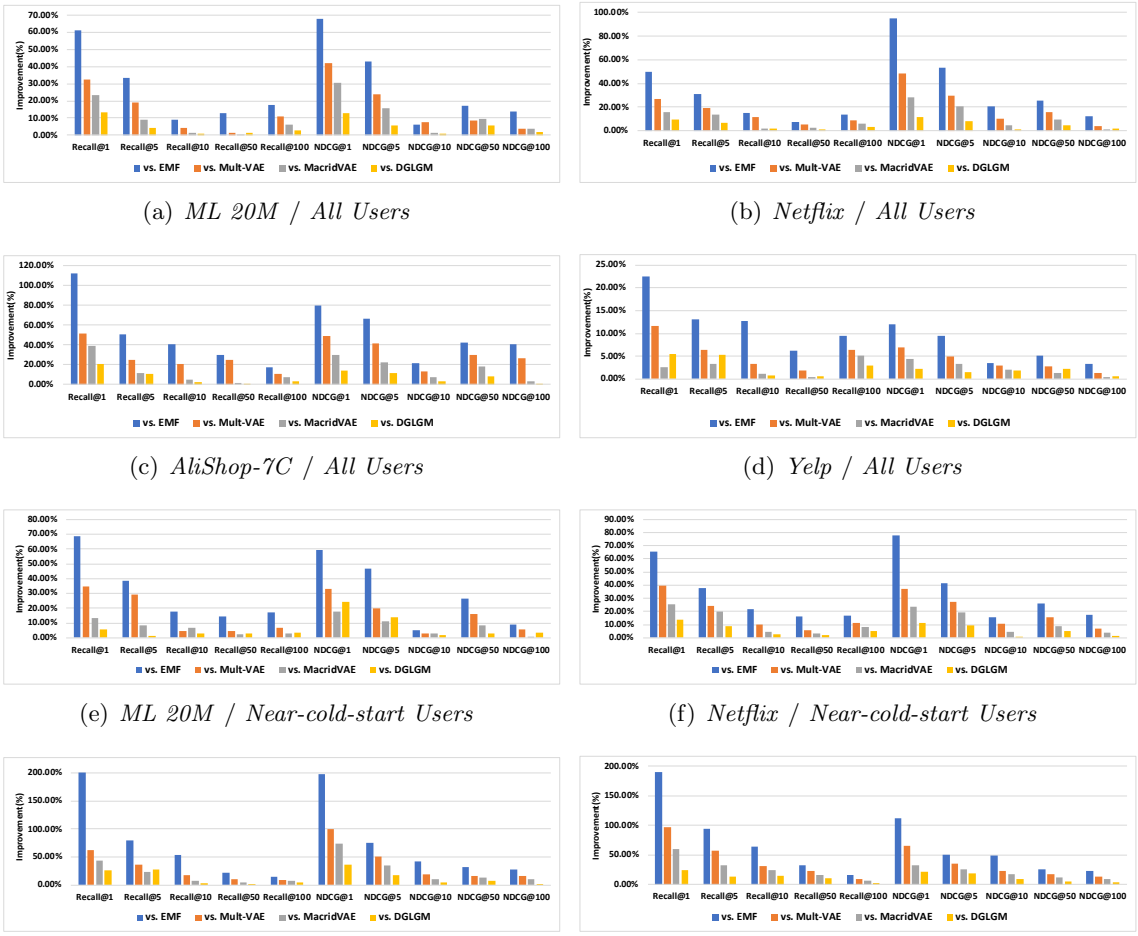
Table 3: Comparisons of different methods in terms of ranking estimation (Recall and NDCG) from *Near-cold-start Users* view.

superior to all baselines. The main reason, we believe, is that InDGRM sufficiently exploits the user correlations which is helpful to characterize the *Near-cold-start Users*. Inter-user preference similarity and intra-user preference diversity not only allow us to accurately represent the diverse interests of a user with the aid of group structure among users and items, but also alleviates data sparsity by allowing a rarely visited item to propagate information from other items of the same item group.

For demonstrating the efficiency of the proposed InDGRM method intuitively, we calculate improvements between InDGRM and four baselines, as shown in Figure 3. Although the percentage of relative improvements are small, small improvements can lead to significant differences of recommendations in practice (Koren et al., 2010). Meanwhile, we conduct paired t-test (confidence 0.95) between InDGRM and each baseline with five-fold cross-validation results. As shown in Table 4, we list the p-value obtained by InDGRM vs. four baselines on *All Users* view. The p-values in all cases are less than 10^{-5}, which indicates that our improvements are statistically significant at the 5% level. Similar statistical test results can be found in *Near-cold-start Users* view, we omit it here. Therefore, based on these observations, we can say InDGRM consistently outperforms the state-of-the-art recommendation methods and significantly improves the recommendation performance.

6.2.2 Interpretability Performance

Following Abdollahi and Nasraoui (2016) and Abdollahi and Nasraoui (2017), two metrics, Mean Explainable Precision (MEP) and Mean Explainable Recall (MER), are adopted to evaluate the performance on explainability. Larger MEP and MER indicate better interpretability performance. As shown in Table 5 and 6, InDGRM consistently performs better



(a) *ML 20M / All Users* (b) *Netflix / All Users*

(c) *AliShop-7C / All Users* (d) *Yelp / All Users*

(e) *ML 20M / Near-cold-start Users* (f) *Netflix / Near-cold-start Users*

(g) *AliShop-7C / Near-cold-start Users* (h) *Yelp / Near-cold-start Users*

Figure 3: The relative improvements of InDGRM vs. four baselines in terms of recommendation performance (Recall and NDCG) on *All Users* and *Near-cold-start Users* views.

Datasets	Methods	Recall					NDCG				
		@1	@5	@10	@50	@100	@1	@5	@10	@50	@100
ML 20M	vs.EMF	2.52E-08	7.12E-08	4.93E-06	7.66E-07	9.09E-06	6.34E-07	7.85E-05	1.41E-07	8.65E-06	5.23E-05
	vs.Mult-VAE	4.88E-07	3.78E-07	8.24E-09	4.79E-07	5.29E-05	5.52E-06	6.19E-05	3.95E-05	3.83E-08	4.42E-08
	vs.MacridVAE	1.38E-08	2.28E-07	3.57E-07	9.85E-07	4.99E-08	5.52E-06	6.19E-05	6.23E-06	3.53E-06	2.53E-08
	vs.DGLGM	6.07E-08	5.18E-06	4.67E-07	6.44E-06	1.69E-06	3.52E-05	5.89E-07	9.85E-07	4.99E-05	5.52E-06
Netflix	vs.EMF	5.32E-06	5.75E-07	7.34E-09	4.79E-07	6.21E-07	3.58E-06	5.34E-06	1.55E-06	2.42E-08	3.52E-07
	vs.Mult-VAE	8.07E-06	7.43E-08	3.66E-08	9.85E-07	4.99E-05	5.52E-06	6.19E-05	4.33E-08	2.22E-06	3.09E-08
	vs.MacridVAE	8.08E-08	3.66E-08	5.23E-09	5.31E-05	3.91E-05	2.53E-07	9.12E-06	1.03E-08	1.39E-06	5.78E-07
	vs.DGLGM	3.55E-07	3.53E-08	5.23E-07	3.33E-06	3.72E-06	3.31E-07	2.69E-05	2.55E-05	8.97E-06	6.45E-07
AliShop-7C	vs.EMF	6.95E-05	5.56E-07	3.22E-06	6.54E-06	3.33E-06	4.12E-07	2.88E-05	9.07E-06	8.51E-06	3.53E-06
	vs.Mult-VAE	8.06E-05	5.24E-05	3.52E-06	2.72E-07	3.36E-06	3.53E-07	2.88E-09	2.79E-07	5.33E-05	2.78E-08
	vs.MacridVAE	8.32E-06	2.90E-05	4.98E-05	5.33E-09	3.23E-07	3.11E-06	2.93E-05	6.63E-06	1.20E-06	4.66E-08
	vs.DGLGM	4.19E-07	3.36E-06	5.81E-05	3.82E-06	2.94E-06	2.26E-06	6.11E-08	3.81E-07	1.92E-06	3.44E-07
Yelp	vs.EMF	9.54E-07	5.93E-05	4.42E-06	2.73E-06	9.49E-06	8.58E-08	3.14E-05	4.96E-07	6.12E-06	8.60E-06
	vs.Mult-VAE	9.65E-06	7.56E-07	4.86E-07	2.83E-06	2.44E-05	2.55E-07	3.33E-08	3.73E-06	3.95E-06	8.68E-08
	vs.MacridVAE	5.23E-06	4.22E-07	4.62E-08	4.71E-06	2.79E-06	3.58E-07	3.38E-06	3.34E-07	8.74E-06	3.53E-08
	vs.DGLGM	3.65E-08	2.99E-06	3.52E-07	6.26E-08	2.27E-06	3.33E-06	6.36E-07	6.35E-07	3.47E-08	3.27E-05

Table 4: Statistical significance (p-value) obtained by InDGRM vs. four baselines on *All Users* view.

Datasets	Methods	MEP					MER				
		@1	@5	@10	@50	@100	@1	@5	@10	@50	@100
ML 20M	EMF	0.1233	0.2524	0.5121	0.6422	0.7832	0.0023	0.0046	0.0066	0.0098	0.1135
	Mult-VAE	0.1354	0.2653	0.5001	0.6555	0.7943	0.0037	0.0049	0.0062	0.0142	0.1256
	MacridVAE	0.1398	0.2734	**0.5242**	0.6653	0.8033	0.0035	0.0057	0.0068	0.0189	0.1354
	DGLGM	0.1453	0.2844	0.5232	**0.6873**	0.8142	0.0043	0.0068	0.0069	0.0197	0.1398
	InDGRM	**0.1512**	**0.2952**	0.5234	0.6793	**0.8233**	**0.0056**	**0.0075**	**0.0073**	**0.0202**	**0.1435**
Netflix	EMF	0.2313	0.3766	0.6657	0.7322	0.7932	0.0023	0.0056	0.0093	0.0145	0.0342
	Mult-VAE	0.2521	0.3872	0.6325	0.7593	0.8132	0.0028	0.0067	0.0089	0.0242	0.0433
	MacridVAE	0.2633	0.3933	0.6732	0.7633	0.8234	0.0034	0.0075	0.0094	0.0342	0.0489
	DGLGM	0.2692	0.3842	0.6643	0.7768	0.8293	0.0038	**0.0079**	0.0095	0.0387	0.0532
	InDGRM	**0.2753**	**0.3992**	**0.6801**	**0.7832**	**0.8432**	**0.0046**	0.0073	**0.0096**	**0.0424**	**0.0593**
AliShop-7C	EMF	0.1452	0.3522	0.5433	0.6892	0.7832	0.0242	0.0456	0.0742	0.0953	0.1231
	Mult-VAE	0.1569	0.3688	0.5401	0.6945	0.7985	0.0356	0.0543	0.0753	0.1032	0.1389
	MacridVAE	0.1678	0.3764	0.5552	0.7065	0.8032	0.0384	0.0598	0.0843	0.1135	0.1456
	DGLGM	0.1732	0.3734	0.5563	0.7121	0.8142	0.0396	0.0613	0.0855	0.1189	0.1498
	InDGRM	**0.1787**	**0.3875**	**0.5642**	**0.7232**	**0.8214**	**0.0406**	**0.0645**	**0.0913**	**0.1232**	**0.1534**
Yelp	EMF	0.1358	0.2533	0.3923	0.4983	0.6732	0.0142	0.0235	0.0357	0.0642	0.0983
	Mult-VAE	0.1542	0.2661	0.3778	0.5122	0.6893	0.0246	0.0342	0.0353	0.0714	0.1042
	MacridVAE	0.1688	0.2785	0.3943	0.5235	0.6832	0.0346	0.0398	0.0358	0.0798	0.1098
	DGLGM	0.1752	0.2549	0.3913	0.5212	0.6901	0.0389	0.0412	0.0358	**0.0833**	0.1123
	InDGRM	**0.1872**	**0.2911**	**0.4078**	**0.5378**	**0.6993**	**0.0477**	**0.0479**	**0.0361**	0.0819	**0.1242**

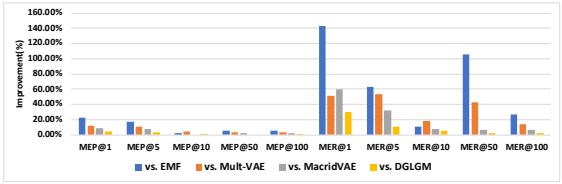
Table 5: Comparing different methods in terms of explainability metrics (MEP and MER) on *All Users* view.

Datasets	Methods	MEP					MER				
		@1	@5	@10	@50	@100	@1	@5	@10	@50	@100
ML 20M	EMF	0.0544	0.1823	0.2152	0.4222	0.5433	0.0012	0.0024	0.0037	0.0065	0.0842
	Mult-VAE	0.0636	1922	0.2251	0.4367	0.5546	0.0014	0.0027	0.0042	0.0072	0.0887
	MacridVAE	0.0698	0.1976	0.2336	0.4403	0.5598	0.0018	0.0032	0.0046	0.0078	0.0938
	DGLGM	0.0721	0.1989	0.2386	0.4451	0.5648	0.0021	0.0037	0.0051	0.0094	0.0984
	InDGRM	**0.0789**	**0.2032**	**0.2451**	**0.4526**	**0.5687**	**0.0026**	**0.0043**	**0.0054**	**0.0098**	**0.1032**
Netflix	EMF	0.1124	0.1831	0.3326	0.4257	0.5452	0.0014	0.0024	0.0045	0.0112	0.0215
	Mult-VAE	0.1255	0.1925	0.3453	0.4333	0.5544	0.0016	0.0027	0.0047	0.0124	0.0227
	MacridVAE	0.1358	0.1993	0.3533	0.4428	0.5646	0.0019	0.0033	0.0049	0.0135	0.0245
	DGLGM	0.1527	0.2043	0.3596	0.4489	0.5766	0.0023	0.0036	0.0051	0.0153	0.0263
	InDGRM	**0.1738**	**0.2176**	**0.3645**	**0.4573**	**0.5834**	**0.0026**	**0.0042**	**0.0056**	**0.0178**	**0.0276**
AliShop-7C	EMF	0.0633	0.1443	0.2353	0.3411	0.5123	0.0133	0.0225	0.0412	0.0591	0.0716
	Mult-VAE	0.0783	0.1325	0.2443	0.3537	0.5236	0.0147	0.0261	0.0451	0.0633	0.0821
	MacridVAE	0.0856	0.1489	0.2495	0.3646	0.5312	0.0166	0.0276	0.0487	0.0684	0.0895
	DGLGM	0.0945	0.1437	0.2575	0.3735	0.5463	0.0178	0.0296	0.0516	0.0742	0.0963
	InDGRM	**0.0987**	**0.1514**	**0.2651**	**0.3829**	**0.5535**	**0.0189**	**0.0314**	**0.0578**	**0.0793**	**0.1035**
Yelp	EMF	0.0256	0.1336	0.1945	0.3144	0.4877	0.0064	0.0121	0.0215	0.0401	0.0634
	Mult-VAE	0.0358	0.1453	0.2042	0.3228	0.4964	0.0121	0.0154	0.0226	0.0458	0.0712
	MacridVAE	0.0411	0.1478	0.2148	0.3328	0.5024	0.0153	0.0168	0.0231	0.0549	0.0789
	DGLGM	0.0436	0.1562	0.2235	0.3446	0.5120	0.0162	0.0185	0.0248	0.0637	0.0846
	InDGRM	**0.0458**	**0.1635**	**0.2358**	**0.3562**	**0.5258**	**0.0178**	**0.0197**	**0.0262**	**0.0691**	**0.0915**

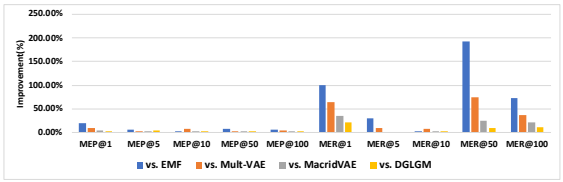
Table 6: Comparing different methods in terms of explainability metrics (MEP and MER) on *Near-cold-start Users* view.

Dataset	Methods	0-5		6-20		21-50		51-100		> 100	
		MEP	MER	MEP	MER	MEP	MER	MEP	MER	MEP	MER
ML 20M	EMF	0.4578	0.0065	0.4656	0.0065	0.4756	0.0068	0.4941	0.0069	0.5153	0.0070
	MacridVAE	0.4615	0.0066	0.4734	0.0068	0.4861	0.0069	0.4904	0.0069	0.5211	0.0070
	InDGRM	**0.4745**	**0.0068**	**0.4989**	**0.0069**	**0.4868**	**0.0071**	**0.4922**	**0.0072**	**0.5222**	**0.0071**
Netflix	EMF	0.5812	0.0088	0.6479	0.0090	0.6621	0.0093	0.6656	0.0095	0.6653	0.0096
	MacridVAE	0.6033	0.0084	0.6472	0.0086	0.6615	0.0089	0.6714	0.0090	0.6731	0.0097
	InDGRM	**0.6232**	**0.0089**	**0.6574**	**0.0092**	**0.6734**	**0.0095**	**0.6802**	**0.0096**	**0.6812**	**0.0098**
AliShop-7C	EMF	0.5053	0.0698	0.5315	0.0713	0.5401	0.0739	0.5423	0.0742	0.5432	0.0752
	MacridVAE	0.5231	0.0783	0.5351	0.0803	0.5415	0.0822	0.5514	0.0832	0.5531	0.0847
	InDGRM	**0.5389**	**0.0834**	**0.5571**	**0.0856**	**0.5634**	**0.0884**	**0.5643**	**0.0898**	**0.5652**	**0.0911**
Yelp	EMF	0.3521	0.0357	0.3751	0.0363	0.3901	0.0369	0.3925	0.0371	0.3911	0.0374
	MacridVAE	0.3656	0.0353	0.3765	0.0359	0.3901	0.0365	0.3914	0.0368	0.3931	0.0373
	InDGRM	**0.3733**	**0.0362**	**0.3856**	**0.0371**	**0.3953**	**0.0376**	**0.3998**	**0.0379**	**0.3998**	**0.0384**

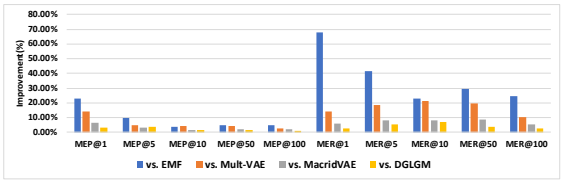
Table 7: Comparisons of explainable recommendation methods (EMF, MacriVAE and InDGRM) on all items with different interaction degrees in terms of MEP@10 and MER@10.



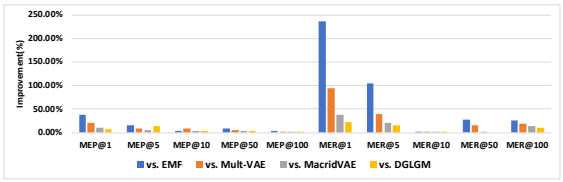
(a) *ML 20M / All Users*



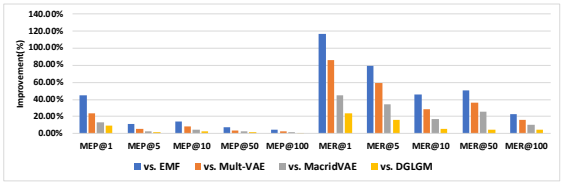
(b) *Netflix / All Users*



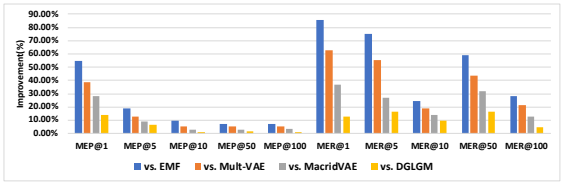
(c) *AliShop-7C / All Users*



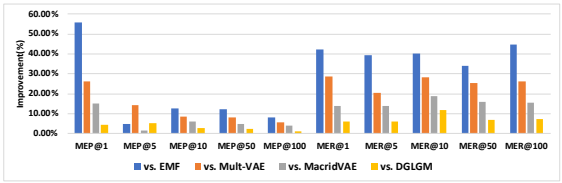
(d) *Yelp / All Users*



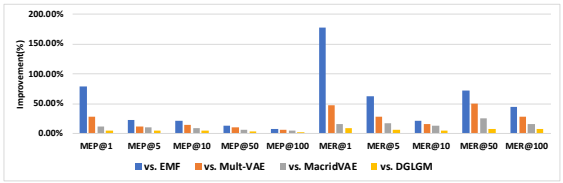
(e) *ML 20M / Near-cold-start Users*



(f) *Netflix / Near-cold-start Users*



(g) *AliShop-7C / Near-cold-start Users*



(h) *Yelp / Near-cold-start Users*

Figure 4: The relative improvements of InDGRM vs. four baselines in terms of interpretability (MEP and MER) on *All Users* and *Near-cold-start Users* view.

than baselines in terms of both *All Users* view and *Near-cold-start Users* view. These results demonstrate that the proposed model has ability to improve the interpretability of recommendation system. At least, the recommended list includes much more explainable recommended items for each user. Note that whether a recommended item is explainable or not depends on whether it is also interacted by similar users. The good performance benefits from two points. Firstly, InDGRM sufficiently determines the user-cluster structure, which is helpful to determine similar users according to their similar preference. Secondly, InDGRM bridges the semantic gap between the latent factors and user-item interaction behavior groups, which is able to disentangle the user latent representation and further improve the model's interpretability. We also report the improvements between InDGRM between four baselines in terms of interpretability metrics, as shown in Figure 4. We can see that the improvements on MEP are more obvious, that is, InDGRM can recall more interpretable items than the baselines. The possible reason is that prototype learning in InDGRM can effectively mine the relationship between items and achieve better preference modeling with the aid of sparse mapping mechanism.

To further demonstrate the interpretability, we split items into five groups (0-5, 6-20, 21-50, 51-100, >100) according to their number of interactions from all users in training data. Three interpretable methods (EMF, MacridVAE and InDGRM) are used for comparison. The interpretability performance on each group is shown in Table 7 (in terms of MEP and MER). The proposed InDGRM and baselines have the similar trends with respect to different groups, i.e., MEP and MER becomes better and better with the increasing of interactions, which indicates that item interaction frequency plays an important role in interpretability performance. To be exciting, the proposed InDGRM significantly outperforms the baselines for all groups. This result further demonstrates that InDGRM has the ability to effectively handle data with various items, especially for *Near-cold-start Items*.

6.2.3 Ablation Test

The proposed InDGRM can be regraded as an improved deep generative model equipped with three modules, mixture of *Gaussian* prior for latent user representation, item group mining and factor-to-group sparse mapping. In this ablation experiment, we try to demonstrate the effect of each module. Two simplified versions of InDGRM (denoted as InDGRM-a and InDGRM-b) are mainly constructed. InDGRM-a considers item group mining and factor-to-group sparse mapping, and removes the mixture of *Gaussian* prior. InDGRM-b only considers mixture of *Gaussian* prior.

Table 8 lists the results (including recommendation performance and interpretability) on Mult-VAE, InDGRM-a, InDGRM-b and InDGRM in terms of *All Users* view and *Near-cold-start Users* view. Here Mult-VAE is used as a basic baseline, which is deep generative model but does not consider three modules (Kingma and Welling, 2013). InDGRM-a and InDGRM-b are superior to Mult-VAE, which show the effectiveness of the corresponding proposed modules. InDGRM-b is superior to InDGRM-a in terms of interpretability evaluation metrics (MEP and MER). The possible reason is that mixture of *Gaussian* prior is able to grouping users with similar preference. Like the neighborhood-based interpretation, this scheme is helpful to provide clues for interpretation. As expected, InDGRM outperforms Mult-VAE, InDGRM-a and InDGRM-b. This confirms that modeling both inter-user

Datasets	Methods	*All Users*				*Near-cold-start Users*			
		Recall@10	NDCG@10	MEP@10	MER@10	Recall@10	NDCG@10	MEP@10	MER@10
ML 20M	Mult-VAE	0.3320	0.3185	0.5001	0.0062	0.2390	0.1976	0.2251	0.0042
	InDGRM-a	0.3368	0.3365	0.5194	0.0070	0.2415	0.1996	0.2402	0.0048
	InDGRM-b	0.3397	0.3396	0.5203	0.0071	0.2433	0.2004	0.2413	0.0050
	InDGRM	**0.3458**	**0.3426**	**0.5234**	**0.0073**	**0.2498**	**0.2033**	**0.2451**	**0.0054**
Netflix	Mult-VAE	0.3208	0.2449	0.6325	0.0089	0.1945	0.1938	0.3453	0.0047
	InDGRM-a	0.3453	0.2623	0.6725	0.0092	0.2078	0.2103	0.3597	0.0052
	InDGRM-b	0.3496	0.2649	0.6743	0.0093	0.2093	0.2112	0.3605	0.0053
	InDGRM	**0.3585**	**0.2695**	**0.6801**	**0.0096**	**0.2136**	**0.2144**	**0.3645**	**0.0056**
AliShop-7C	Mult-VAE	0.1343	0.1632	0.5401	0.0753	0.0952	0.1023	0.2443	0.0451
	InDGRM-a	0.1573	0.1804	0.5591	0.8985	0.1095	0.1183	0.1603	0.0521
	InDGRM-b	0.1589	0.1821	0.5604	0.0900	0.1101	0.1191	0.2617	0.0543
	InDGRM	**0.1622**	**0.1853**	**0.5642**	**0.0913**	**0.1126**	**0.1214**	**0.2651**	**0.0578**
Yelp	Mult-VAE	0.5322	0.7328	0.3778	0.0353	0.0981	0.1352	0.2042	0.0226
	InDGRM-a	0.5421	0.7356	0.3984	0.0359	0.1589	0.1600	0.2287	0.0245
	InDGRM-b	0.5443	0.7390	0.4025	0.0360	0.1611	0.1621	0.2307	0.0251
	InDGRM	**0.5496**	**0.7542**	**0.4078**	**0.0361**	**0.1289**	**0.1672**	**0.2358**	**0.0262**

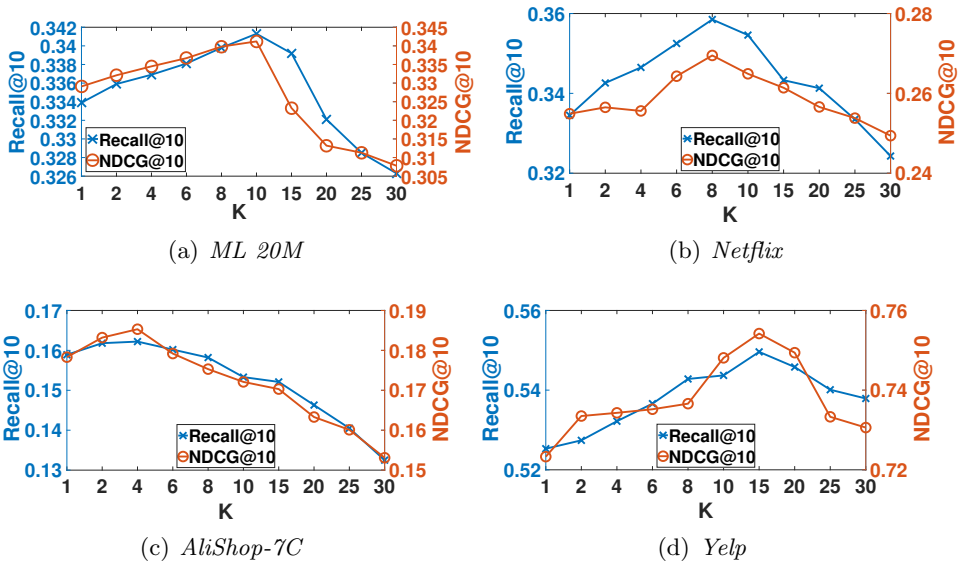
Table 8: Ablation analysis of InDGRM in terms of different metrics.

Datasets	Methods	*All Users*				*Near-cold-start Users*			
		Recall@10	NDCG@10	MEP@10	MER@10	Recall@10	NDCG@10	MEP@10	MER@10
ML 20M	InDGRM-p	**0.3498**	**0.3545**	**0.5361**	**0.0077**	**0.2590**	**0.2136**	**0.2563**	**0.0057**
	InDGRM	0.3458	0.3426	0.5234	0.0073	0.2498	0.2033	0.2451	0.0054
Netflix	InDGRM-p	**0.3658**	**0.2789**	**0.6942**	**0.0098**	**0.2251**	**0.2261**	**0.3763**	**0.0059**
	InDGRM	0.3585	0.2695	0.6801	0.0096	0.2136	0.2144	0.3645	0.0056
AliShop-7C	InDGRM-p	**0.1733**	**0.1894**	**0.5731**	**0.0963**	**0.1262**	**0.1331**	**0.2721**	**0.0615**
	InDGRM	0.1622	0.1853	0.5642	0.0913	0.1126	0.1214	0.2651	0.0578
Yelp	InDGRM-p	**0.5562**	**0.5631**	**0.4151**	**0.0363**	**0.1341**	**0.1721**	**0.2425**	**0.0273**
	InDGRM	0.5496	0.7542	0.4078	0.0361	0.1289	0.1672	0.2358	0.0262

Table 9: Comparison between InDGRM and InDGRM-p in terms of different metrics.

preference similarity and intra-user preference diversity has ability to significantly improve model performance. Furthermore, InDGRM with prototype enhancing by considering privilege information (InDGRM-p) is also investigated. Table 9 lists the comparison results between InDGRM-p and InDGRM. We can see that prototype enhancing with privilege information is helpful for better preference modeling and further improving performance in terms of both recommendation accuracy and interpretability.

Additionally, the number of mixture components K is a key parameter in the proposed InDGRM, which affects the ability of capturing the inter-user preference similarity. Thus, we conduct experiment to evaluate the effect of K on the recommendation performance of InDGRM. Among it, K is selected from {1, 2, 4, 6, 8, 10, 15, 20, 25, 30}. Figure 5 shows the recommendation performance (in terms of Recall@10 and NDCG@10) by varying K. It can be seen that the performance is improved with the increasing of K. This is due to the fact that different mixture components may capture different user preferences and further improve recommendation performance. After reaching the optimal value K (The optimal values are $6, 8, 4$ and 15 on *ML 20M*, *Netflix*, *AliShop-7C* and *Yelp*, respectively), the performance decreases when K further increases. This phenomenon implies that large K may overfit the training data. Meanwhile, an interesting observation is that the optimal K is larger on the large dataset than that on the small dataset. This is reasonable because



(a) *ML 20M* (b) *Netflix*

(c) *AliShop-7C* (d) *Yelp*

Figure 5: Effect of K (the number of mixture components (i.e., user clusters)) on InDGRM.

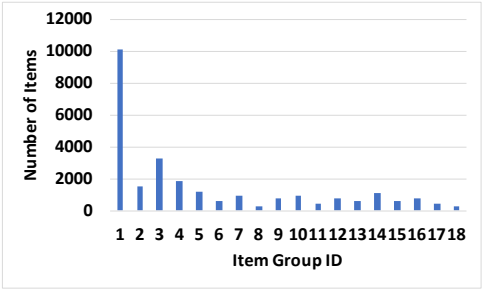
large dataset contains the larger number of users and items, which is much more complicated and requires more components to capture their characteristics.

In Section 4.2, the cosine similarity, instead of inner product similarity, is used to calculate the correlations between each pair of item and propertype. In this experiment, a new model with inner product similarity is trained to compare that with cosine similarity. Figure 6 shows the learned item groups with two models on *Ml 20M* and *AliShop*. It can be seen that, inner product promotes most items belonging to one prototype (see Figure 6 (a) and (c)). As expected, each prototype output by cosine-based model can contain a certain number of items (see Figure 6 (b) and (d)). This finding confirms our claim that selecting a proper metric, such as cosine similarity, is important for model training.

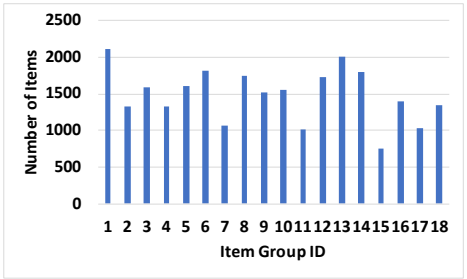
6.2.4 Observed-level Disentanglement Analysis

To capture the user-cluster structure reflecting inter-user preference similarity, the mixture prior is introduced to model user's preference so that users with similar tastes can be assigned to the same group. To confirm this, taking *ML 20M* dataset as an example, we try to investigate the corresponding semantical information of the mixture components with the aid of side information. In *ML 20M*, each movie is marked by one or more genres, and all movies belong to 18 genres, including '*Action*', '*Adventure*', '*Animation*', '*Children*', '*Comedy*', '*Crime*', '*Documentary*', '*Drama*', '*Fantasy*', '*Film-Noir*', '*Horror*', '*Musical*', '*Mystery*', '*Romance*', '*Sci-Fi*', '*Thriller*', '*War*' and '*Western*'.

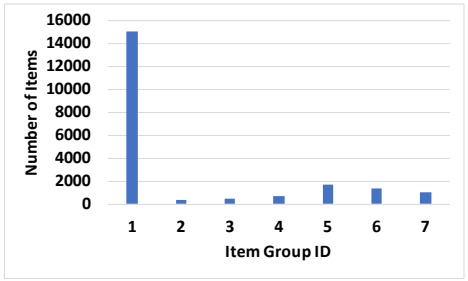
For the k-th user cluster obtained by InDGRM, let $M^{(k)}$ be the number of items related by the users belonging to this cluster, where the number of items belonging to the q-th genre is denoted as $M_q^{(k)}$ and $\sum_{q=1}^{N_l} M_q^{(k)} = M^{(k)}$ ($N_l = 18$ (the number of genres) in *ML 20M*). $\{M_q^{(k)}/M^{(k)}\}_{q=1}^{N_l}$ indicates the item proportion along different genres for k-th user



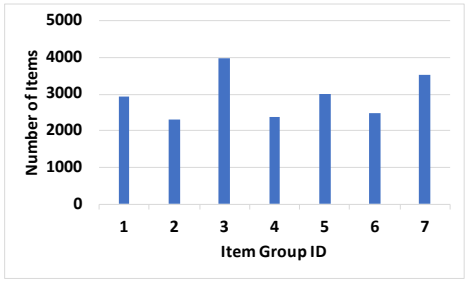
(a) Item groups with inner product similarity on *ML 20M*



(b) Item groups with cosine similarity on *ML 20M*



(c) Item groups with inner product similarity on *AliShop-7C*



(d) Item groups with cosine similarity on *AliShop-7C*

Figure 6: The number of item in each item group obtained via InDGRM with cosine similarity and inner product similarity.

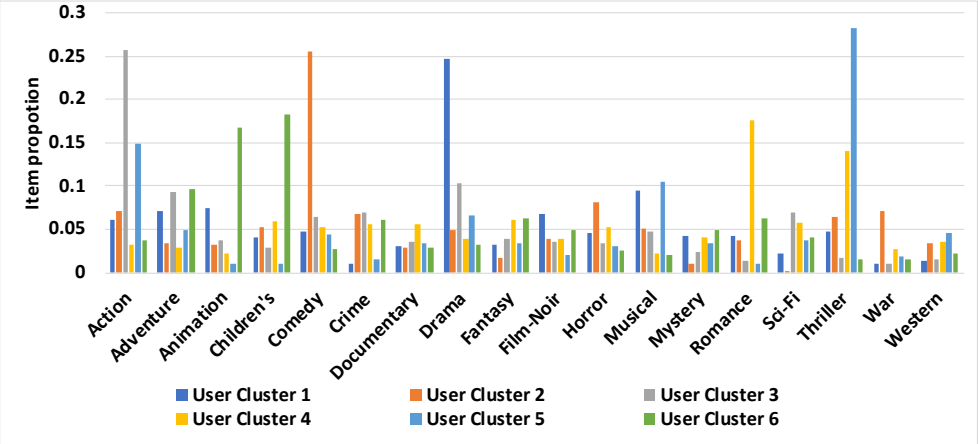


Figure 7: The item proportion with genre in each user clusters obtained by InDGRM on *ML 20M*.

Dataset	K^*	$d^{(k)}$ in each component
ML 20M	6	{159, 126, 147, 152, 144, 156}
Netflix	8	{144, 78, 134, 153, 126, 159, 145, 114}
AliShop-7C	4	{136, 184, 135, 90}
Yelp	15	{137, 126, 154, 142, 137, 165, 121, 95, 78, 115, 125, 147, 89, 120, 156}

Table 10: The optimal latent dimensionality d^k in each component determined by InDGRM on four datasets. K^* is the optimal number of mixture components. d^k is the optimal dimensionality in the k-th component.

cluster. The item proportion with specific genre is shown for six clusters in Figure 7. It can be seen that most items ($> 70\%$) in one component are always from the same genre, such as {*Drama*}, {*Comedy*}, {*Action*}, {*Romance*}, {*Thriller*}, {*Animation*, *Children's*} for user cluster 1 to 6. This result indicates each user cluster definitely captures one main user preference. Thus, we can say InDGRM has the ability to determine the user-cluster structure and achieve observed-level disentanglement on users.

Thanks for *ARD* technique with thresh, the proposed InDGRM model can adaptively determine the optimal latent dimensionality for each mixture component. Table 10 demonstrates the optimal latent space size for $\mathbf{z}_i$ in each user cluster on different datasets. It can be seen that the latent dimensionalities have different variances from different datasets. For instance, the minimal and maximal sizes are {126, 159}, {78, 159}, {90, 184} and {78, 165} on *ML 20M*, *Netflix*, *AliShop-7C* and *Yelp*, respectively. An interesting thing is that the optimal size is almost proportional to the group density. This result confirms that user clusters with few interactions should be represented in a lower dimensional space, other be of higher dimensional space, which is consistent with the conclusion given in Li et al. (2017) and Liu et al. (2019a).

Datasets	*ML 20M*		*Netflix*		*AliShop-7C*		*Yelp*	
Metrics	Recall@10	NDCG@10	Recall@10	NDCG@10	Recall@10	NDCG@10	Recall@10	NDCG@10
InDGRM(d=10)	0.3334	0.3323	0.3401	0.2489	0.1537	0.1822	0.5403	0.7343
InDGRM(d=20)	0.3338	0.3357	0.3419	0.2504	0.1558	0.1824	0.5405	0.7369
InDGRM(d=50)	0.3314	0.3283	0.3405	0.2545	0.1543	0.1784	0.5393	0.7354
InDGRM(d=80)	0.3321	0.3314	0.3414	0.2506	0.1545	0.1805	0.5398	0.7365
InDGRM(d=90)	0.3345	0.3355	0.3422	0.2555	0.1546	0.1815	0.5421	0.7388
InDGRM(d=100)	0.3326	0.3311	0.3431	0.2553	0.1589	0.1835	0.5431	0.7391
InDGRM(d=110)	0.3354	0.3326	0.3457	0.2589	0.1598	0.1846	0.5415	0.7387
InDGRM(d=120)	0.3389	0.3331	0.3478	0.2656	0.1577	0.1836	0.5396	0.7385
InDGRM(d=130)	0.3413	0.3358	0.3532	0.2675	0.1568	0.1827	0.5382	0.7378
InDGRM(d=140)	0.3424	0.3379	0.3492	0.2631	0.1555	0.1815	0.5390	0.7381
InDGRM(d=150)	0.3373	0.3369	0.3455	0.2575	0.1532	0.1789	0.5395	0.7383
InDGRM(d=160)	0.3367	0.3358	0.3432	0.2557	0.1576	0.1825	0.5374	0.7376
InDGRM(d=170)	0.3354	0.3328	0.3446	0.2565	0.1554	0.1813	0.5353	0.7368
InDGRM(d=180)	0.3321	0.3310	0.3421	0.2545	0.1533	0.1804	0.5365	0.7372
InDGRM(d=190)	0.3348	0.3356	0.3468	0.2595	0.1565	0.1819	0.5345	0.7369
InDGRM(d=200)	0.3346	0.3357	0.3446	0.2564	0.1542	0.1816	0.5327	0.7361
DGLGM	0.3428	0.3391	0.3513	0.2669	0.1585	0.1792	0.5458	0.7404
InDGRM	**0.3458**	**0.3426**	**0.3585**	**0.2695**	**0.1622**	**0.1853**	**0.5496**	**0.7542**

Table 11: Recommendation performance of InDGRM against InDGRM with fixed latent dimensionality for $\mathbf{z}_i$ in terms of Recall@10 and NDCG@10.

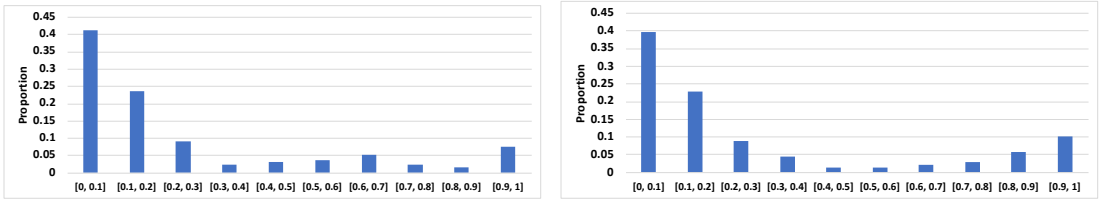
Additionally, we compare the recommendation performance of InDGRM against InDGRM with fixed latent dimensionalities in all components {10, 20, 50, 80, 90, 100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200}. The results are list in Table 11. As we can see, the performance of InDGRM with fixed latent dimensionality is unstable when the dimensionality increases from 10 to 200. The competitive performance can be obtained by InDGRM with fixed dimensions 140, 130, 110 and 100 on *ML 20M*, *Netflix*, *AliShop-7C* and *Yelp*, respectively. In the case of fixed dimensions, the optimal value is better than DGLGM, but still worse than InDGRM. The reason is that mapping users with preference diversity into a fixed latent space size cannot model all users well, so that some users are either under-fitted or over-fitted. Furthermore, InDGRM is superior to DGLGM which adopts *ARD* technique as well, the main reason is that item group structure can model user preference in a more fine-grained manner.

6.2.5 Disentangled User Representations Analysis

a). Factor-to-group Mapping

By telling us the item groups "responsible" for a given prediction, factor-to-group mapping matrices reveal insights about how models rely on and extrapolate from the user behavior. In this subsection, we focus on show how InDGRM can understand model behavior with the aid of factor-to-group mapping mechanism.

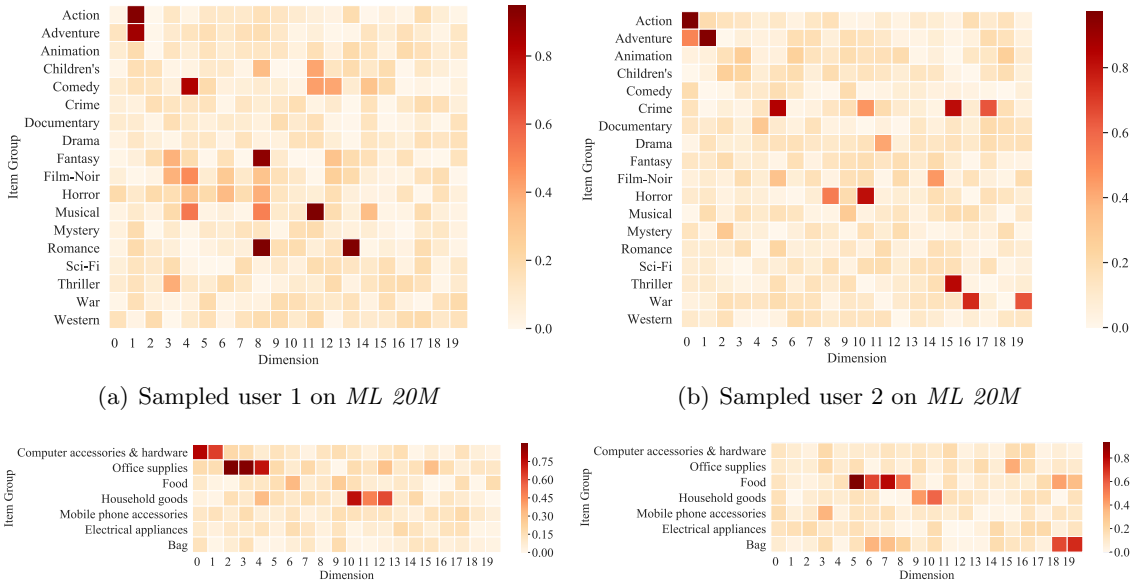
Group structure among items can effectively reflect the intra-user preference diversity. Firstly, we investigate the properties of the learned factor-to-group mapping matrix by splitting its values ($o_{i,l}^{(g)}$) into ten groups at interval of 0.1 according. Figure 8 shows the proportion of mapping value in difference intervals. Obviously, they approximately follow *Laplace-like* distribution. In other words, most values belong to first three ranges, which



(a) *ML 20M* (b) *Netflix*

(c) *AliShop-7C* (d) *Yelp*

Figure 8: Factor-to-group mapping value learned by InDGRM.



(a) Sampled user 1 on *ML 20M* (b) Sampled user 2 on *ML 20M*

(c) Sampled user 1 on *AliShop-7C* (d) Sampled user 2 on *AliShop-7C*

Figure 9: Demonstrating the learned factor-to-group mapping matrices $\mathbf{O}_i^\top$ related to 2 sampled users on *ML 20M* and *AliShop-7C*.

indicates that there is few relations between a large number of dimensions and item groups. This result is obtained by penalizing the interactions via group lasso, which has ability to encourage each factor to capture the distinct modes of variation, so that the correlations among factors of $\mathbf{z}_i$ can be minimized as much as possible. Therefore, InDGRM can achieve the disentanglement of latent item representations.

Furthermore, we plot the learned factor-to-group matrix $\mathbf{O}_i$ for two sampled users in *ML 20M* (Figure 9(a) - (b)) and *AliShop-7C* (Figure 9(c) - (d)). InDGRM successfully disentangles the dimensions of $\mathbf{z}_i$ because each dimension only corresponds to one group with higher probability. For instance, from Figure 9(a), we know that sampled user 1 likes movies with genre *'Action', 'Adventure', 'Comedy', 'Fantasy', 'Musical'*, and *'Romance'*. Sampled user 2 in Figure 9(b) tends to watch movies about *'Action', 'Adventure', 'Crime', 'Horror'* genres, which indicates this user's taste is crime and violence. Category *'Action'* is liked by most of users. The reason is that lots of movies contain genre *'Action'*. Furthermore, we can see some dimensions are associated with multiple genres since some genres are usually co-existed, e.g., an *'Action'* movie is usually an *'Adventure'* movie.

AliShop-7C contains goods with 7 categories, including *'Computer accessories & hardware'*, *'Office supplies'*, *'Food'*, *'Household goods'*, *'Mobile phone accessories'*, *'Electrical appliances'*, and *'Bag'*. Figure 9(c) -(d) show two sample users with diverse interests on *AliShop-7C*. For example, sample user 1 tends to buy *'Computer accessories & hardware'*, *'Office supplies'* and *'Household goods'*, while sample user 2 like to buy *'Food'*, *'Household goods'* and *'Bag'*. Like *ML 20M*, each dimensions is related to its own group. However, unlike in *ML 20M* (where each dimension is related to one group), in *AliShop-7C* each group is usually associated with more than one dimensions. The main reason is that category information in *AliShop-7C* is high-level primary category and contains more categories that can be subdivided. For example, *'Food'* includes *sea food*, *fruit*, *meat* and etc. It can be seen that the factor-to-group mapping matrix is able to demonstrate user's preference explicitly. This interpretable result benefits from the factor-to-group sparse mapping so that each latent factor is associated with distinct mode.

b). Disentangled Representation

Inspired by Kim and Mnih (2018), the disentanglement ability of the latent representations can be quantitatively assessed. Specifically, by fixing a particular latent feature u ($1 \leq u \leq d$), we can generate the user-item interaction information ($\mathbf{x}_i$) with randomly varying other features. Then the empirical variance of each feature is computed according to the normalized generated data. The feature with lowest variance and target feature u will be taken as two samples and fed into a majority-vote classifier to check the *disentanglement* scores of u. The evaluation metric *Independence* (Ma et al., 2019) is adopted to investigate the relationship (independence) between any pair of latent features, which is defined as

$$Independence = 1 - \frac{2}{d(d-1)} \sum_{1 \leq i < j \leq d} |\text{corr}(i, j)|.$$

Here d is the number of latent features. $\text{corr}(i, j)$ is the correlation between dimension i and j, which is computed via cosine similarity.

Table 12 lists the *disentanglement* score (mean $\pm$ one standard deviation) and *independence* score on all latent features for each dataset. It can be seen that InDGRM is

Metrics	Datasets	Mult-VAE	DGLGM	MacridVAE	InDGRM-a	InDGRM-b	InDGRM
Disentanglement	*ML 20M*	0.4122±0.005	0.4132±0.005	0.4218±0.005	0.4253±0.005	0.4144±0.005	**0.4346±0.005**
	Netflix	0.4628±0.005	0.4611±0.006	0.4654±0.006	0.4674±0.005	0.4608±0.005	**0.4844±0.007**
	AliShop-7C	0.3521±0.005	0.3513±0.004	0.3589±0.005	0.3664±0.005	0.3532±0.005	**0.3732±0.006**
	Yelp	0.8341±0.005	0.8315±0.006	0.8536±0.005	0.8594±0.005	0.8321±0.005	**0.8732±0.006**
Independence	*ML 20M*	0.8242	0.8233	0.8344	0.8854	0.8232	**0.9056**
	Netflix	0.5322	0.5325	0.5457	0.5648	0.5344	**0.5677**
	AliShop-7C	0.6042	0.6115	0.6235	0.6389	0.6105	**0.6437**
	Yelp	0.7402	0.7531	0.7783	0.7843	0.7520	**0.7931**

Table 12: Disentanglement and independence evaluation on learned user representations.

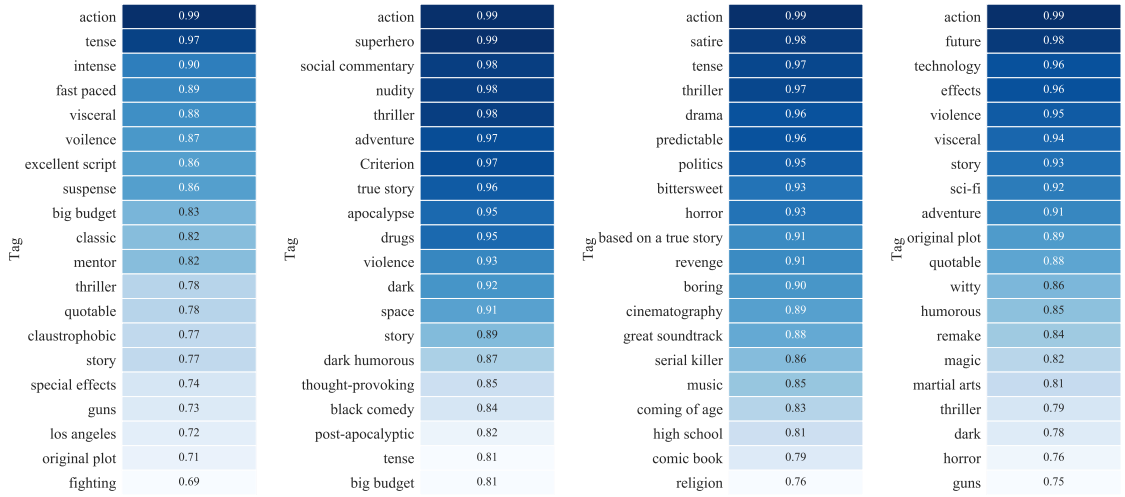
competitive to Mult-VAE, DGLGM and MacridVAE in disentangled and independent ability. To be exciting, InDGRM and InDGRM-a obviously improve the independence among latent features (e.g., improvement gain is larger than 7% on *ML 20M*). The main reason is that each latent factor is associated with only few item groups via the column-wise factor-to-group sparse mapping matrix, which is able to identify disentangled representations. These results further confirm that our approach is able to discover and disentangle the latent structure underlying the user behavior in an interpretable way.

6.2.6 Disentangled Item Representations Analysis

In this experiment, Movielens Tag Genome (Vig et al., 2012) and *ML 20M* dataset are used to evaluate the disentanglement of item representations. In Movielens Tag Genome, each "genome" describes a given movie via 1128 tags with relevance scores between $[0, 1]$. After removing the non-informative tags (e.g., 'good', 'bad') and merging the similar tags (their relevance scores are combined via a weighted mean), each movie is represented by 450 tags.

Here, we aim to evaluate the interpretability of the dimensions in item representation, thus proposing new items to create. Similar to the previous disentanglement experiment on users, the target dimension is gradually altered to demonstrate the disentanglement of item representations. Here we selected four movies with similar tags, and listed their representations in Figure 10. Among it, Figure 10(a)-(d) list four generated movies (GM-1 to GM-4) related to action, and Figure (e)-(h) are about movies about dramas (GM-5 to GM-8). Although these generated movies (by modifying the value of a particular dimension) belong to the same category, they all have their own characteristics. Specificially, GM-1 is an action movie with a tense plot and violence; GM-2 is an adventure action comedy with superhero; GM-3 is an action thriller with a tense plot and satire; GM-4 is an action sci-fi movie with an emphasis on the future and technology; GM-5 is a sentimental drama with mentor and love; GM-6 is a sentimental drama with intense plot and dark humor; GM-7 is a sentimental drama with crime and war; GM-8 a realistic and narrated drama with great ending and friendship. This experiment not only evaluates the disentangled ability of learned item representations, but also determines the important features. For example, the tags shared by the generated movies ("sentimental","mentor", and "story") suggest that these movies have common properties, thus it is necessary to perform item grouping.

Furthermore, we demonstrate the good's images generated by gradually altering the value of target dimension on *AliShop-7C*. Since we can not guarantee the generated items exist in the dataset, only the generated items similar to the existed items are selected. Here, inner-product is used to measure the similarity between two items. For convenient

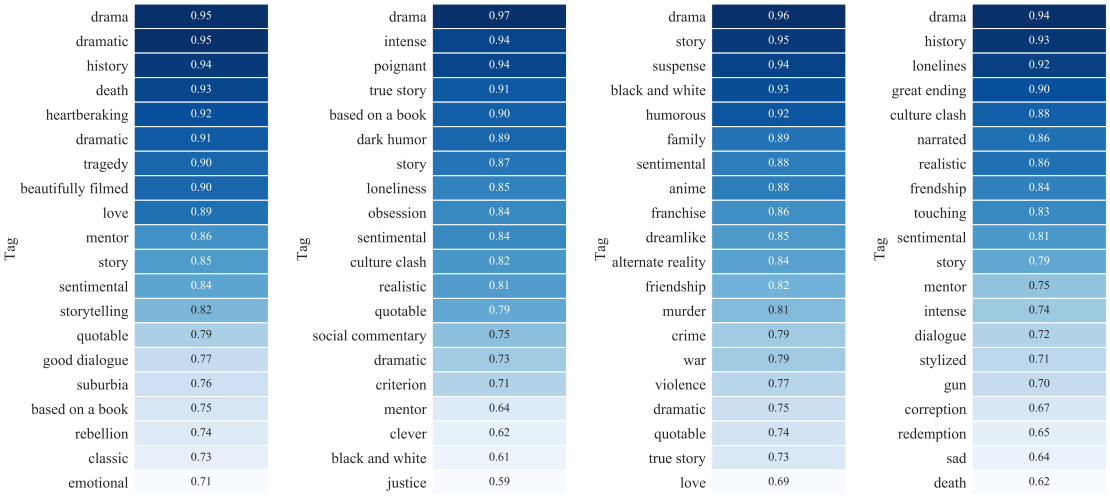


(a) GM-1: An action movie with tense plot and violence

(b) GM-2: An adventure action comedy with superhero

(c) GM-3: An action thriller with a tense plot and satire

(d) GM-4: An action sci-fi movie with an emphasis on the future and technology



(e) GM-5: A sentimental drama with mentor and love

(f) GM-6: A sentimental drama with intense plot and dark humor

(g) GM-7: A sentimental drama with crime and war

(h) GM-8: A realistic and narrated drama with great ending and friendship

Figure 10: Demonstrating top 20 tags for the generated movie tag genomes by varying the value of a target dimension.

(a) The color of fruit: Yellow ⟶ Green ⟶ Red

(b) The color of bag: Pink ⟶ Brown ⟶ Black

(c) The type of flesh: Prawn ⟶ Crab ⟶ Meat

Figure 11: Disentangled representations with human-understandable factor. Starting from an item representation, we gradually alter the value of a target dimension, and list the items that have representations similar to the altered representations.

demonstration, three representative dimensions with human-understandable semantics are used, the color of fruit, the color of bag and the type of flesh. As shown in Figure 11, the target dimensions control the specific aspects, which has ability to provide fine-grained interpretability on the recommendation lists. However, we note that not all dimensions are human-understandable, as shown in Figure 12. Fortunately, the generated items belong to the same category, i.e., fruit, bag and flesh, which demonstrates a high-level disentanglement.

6.2.7 Generalization ability and convergence analysis

In order to evaluate the generalization ability of the proposed model, we choose different training sets with different sizes {20%, 40%, 60%, 80%} and fix the testing set. We guarantee that small size training set is included in large size training set. Since Mult-VAE and aWAE obtain competitive performance among all baselines, the following comparisons and analysis focus on Mult-VAE, MacridVAE, DGLGM and InDGRM. Figure 13 shows the comparison results in terms of NDCG@10 on different training sizes for four datasets. Obviously, the recommendation performance becomes better and better with the increasing of training data. As expected, InDGRM is superior to baselines in all cases. Note that *Yelp* is extremely sparse, Figure 13(d) shows that InDGRM significantly outperforms Mult-VAE, especially for small size training sets (20%, 40%). This result demonstrates that InDGRM has a strong generalization ability.

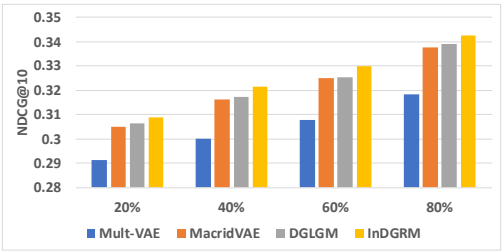
Furthermore, we investigate the convergence property of the proposed local variational optimization (LVO) method. As mentioned before, the proposed LVO algorithm has ability to theoretically obtain tighter upper bound for InDGRM model than the original stochastic

(a) Fruit

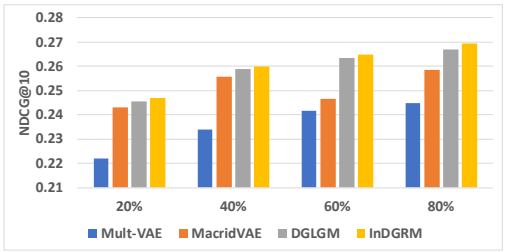
(b) Bag

(c) Flesh

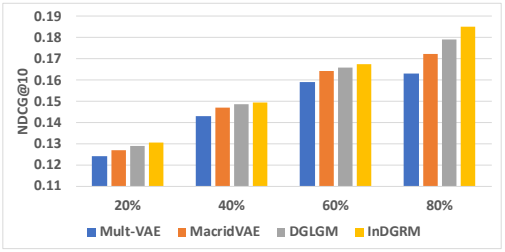
Figure 12: Entangled representations with human-misunderstood factor. Starting from an item representation, we gradually alter the value of a target dimension, and list the items that have representations similar to the altered representations.



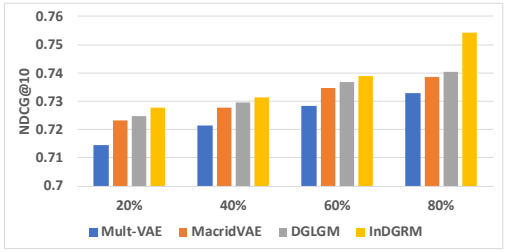
(a) *ML 20M*



(b) *Netflix*



(c) *AliShop-7C*

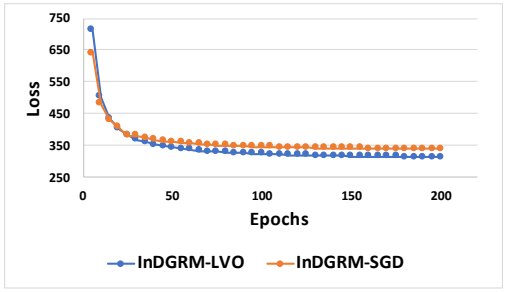


(d) *Yelp*

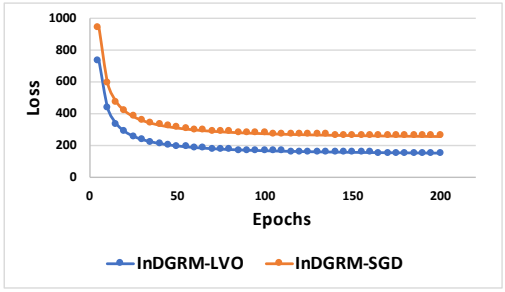
Figure 13: NDCG@10 comparison of Mult-VAE, MacriVAE, DGLGM, and InDGRM in terms of different training sets.

Metrics	Methods	*ML 20M*	*Netflix*	*AliShop-7C*	*Yelp*
InDGRM-SGD	Recall@10	0.3438	0.3536	0.1601	0.5471
	NDCG@10	0.3403	0.2679	0.1821	0.7498
InDGRM-LVO	Recall@10	0.3458	0.3589	0.1622	0.5496
	NDCG@10	0.3426	0.2695	0.1853	0.7542

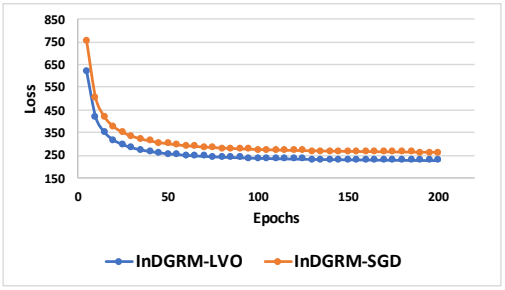
Table 13: Comparisons of InDGRM-SGD and InDGRM-LVO.



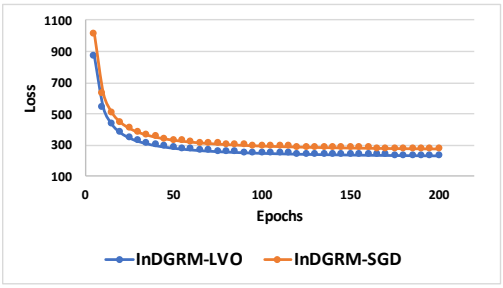
(a) *ML 20M*



(b) *Netflix*



(c) *AliShop-7C*



(d) *Yelp*

Figure 14: Comparison of reconstruction loss versus epochs between InDGRM-SGD and InDGRM-LVO.

gradient descent (SGD) optimization method (see Eq. (17)). The objective loss (reflecting the upper bound) on training set is visualized as a function of epochs in Figure 14. As expected, LVO is much faster to converge and obtain tighter upper bound for our proposed model InDGRM. Additionally, the recommendation performance obtained by InDGRM with SGD and LVO are listed in Table 13. The results indicate that LVO can estimate more precise model parameters and further improve the recommendation performance.

7. Conclusion and Future Work

In this paper, we propose an interpretable deep generative method (InDGRM) for recommendation by modeling both inter-user preference similarity and intra-user preference diversity to achieve disentanglement from both observed-level and latent-level. A mixture distribution is introduced to model the latent representation to capture the user correlations

the aid of non-parametric Bayesian prior. Meanwhile, the items are grouped via a prototype learning technique. With the aid of user grouping and item grouping, the whole deep generative model leads to a meaningful factor-to-group interaction and effectively disentangle the latent factors. The experiments have shown that InDGRM has the ability to output the high-quality recommended items and provide interpretable results. It will be an interesting topic to extend this interpretable model with the aid of multi-resource recommendation data, such as social relations among users, item contents, user profiles and etc.

Acknowledgments

This work was supported in part by This work was supported in part by the Beijing Natural Science Foundation (Z180006); the National Science Foundation of China (61822601, 61773050 and 61632004); Science and Technology Innovation Planning Foundation of Universities from Ministry of Education; the Open Project Program Foundation of the Key Laboratory of Opto-Electronics Information Processing, Chinese Academy of Sciences (OEIP-O-202004). The research of Michale K. Ng is partially supported by the Research Grants Council of Hong Kong, General Research Fund (12300218, 12300519, 17201020 and 17300021). We are also grateful for the anonymous reviewers and the editor for their helpful comments.

References

Behnoush Abdollahi and Olfa Nasraoui. Explainable matrix factorization for collaborative filtering. In *Proceedings of WWW*, pages 5–6, 2016.

Behnoush Abdollahi and Olfa Nasraoui. Using explainability for constrained matrix factorization. In *Proceedings of ACM RecSys*, pages 79–83. ACM, 2017.

G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transaction on Knowledge and Data Engineering*, 17(6):734–749, 2005.

Bhadra Anindya, Datta Jyotishka, Nicholas G. Polson, and Willard Brandon. Default bayesian analysis with global-local shrinkage priors. *Biometrika, 2015, 103(4).*, 103(4): 955–969, 2015.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proceedings of ICML*, pages 214–223, 2017.

Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (gans). In *Proceeding of ICML*, pages 224–232. JMLR. org, 2017.

Krisztian Balog, Filip Radlinski, and Shushan Arakelyan. Transparent, scrutable and explainable user models for personalized recommendation. In *Proceedings of ACM SIGIR*, pages 265–274, 2019.

James S. Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Proceedings of NIPS*, pages 2546–2554. Curran Associates, Inc., 2011.

Alex Beutel, Amr Ahmed, and Alex Smola. Accams: Additive co-clustering to approximate matrices succinctly. In *Proceedings of WWW*, pages 119–129, 2015.

D.M. Blei and M.I. Jordan. Variational inference for dirichlet process mixtures. *Bayesian Anal.*, 42(1):121–144, 2006.

Emmanuel J Candes and Yaniv Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2011.

Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717, 2009.

Chao Chen, Dongsheng Li, Yingying Zhao, Qin Lv, and Li Shang. Wemarec: accurate and scalable recommendation through weighted and ensemble matrix approximation. In *Proceedings of ACM SIGIR*, pages 303–312. ACM, 2015.

Chao Chen, Dongsheng Li, Qin Lv, Junchi Yan, Stephen M Chu, and Li Shang. Mpma: mixture probabilistic matrix approximation for collaborative filtering. In *Proceedings of the IJCAI*, pages 1382–1388, 2016.

Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. Sequential recommendation with user memory networks. In *Proceedings of WSDM*, pages 108–116. ACM, 2018a.

Xu Chen, Yongfeng Zhang, Hongteng Xu, Yixin Cao, Zheng Qin, and Hongyuan Zha. Visually explainable recommendation. *arXiv preprint arXiv:1801.10288*, 2018b.

Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. Sequential user-based recurrent neural network recommendations. In *Proceedings of ACM RecSys*, pages 152–160, 2017.

Zoubin Ghahramani and Thomas L Griffiths. Infinite latent feature models and the indian buffet process. In *Proceedings of NIPS*, pages 475–482, 2006.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of NIPS*, pages 2672–2680, 2014.

James Hampton. Prototype models of concept representation. *Categories and concepts: Theoretical views and inductive data analysis*, 1993.

F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: history and context. *ACM Transaction on Interactive Intelligent System*, 5(4), December 2015.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of WWW*, pages 173–182, 2017.

J. R. Hershey and P. A. Olsen. Approximating the kullback leibler divergence between gaussian mixture models. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, volume 4, pages IV–317–IV–320, 2007.

Hemant Ishwaran and Lancelot F James. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96(453):161–173, 2001.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. On using very large target vocabulary for neural machine translation. In *Proceedings of the ACL*, pages 1–10. Association for Computational Linguistics, July 2015.

Santosh Kabbur, Xia Ning, and George Karypis. Fism: factored item similarity models for top-n recommender systems. In *Proceedings of ACM SIGKDD*, pages 659–667. ACM, 2013.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *Proceedings of ICML*, volume 80, pages 2649–2658, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

Diederik P Kingma and Jimmy Ba. Adam: a method for stochastic optimization. In *Proceedings of ICLR*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Joseph A Konstan, Bradley N Miller, David Maltz, Jonathan L Herlocker, Lee R Gordon, and John Riedl. Grouplens: applying collaborative filtering to usenet news. *Cacm*, 40 (3):77–87, 1997.

Y. Koren, R. Bell, and C. Kolinsky. Factor in the neighbors: scalable and accurate collaborative filtering. *ACM Transaction on Knowledge Discovery from Data*, 4(1):1:1–24, 2010.

Rahul G Krishnan, Dawen Liang, and Matthew Hoffman. On the challenges of learning with inference networks on sparse, high-dimensional data. *Proceedings of AISTATS*, 2017.

Joonseok Lee, Seungyeon Kim, Guy Lebanon, Yoram Singer, and Samy Bengio. Llorma: Local low-rank matrix approximation. *Journal of Machine Learning Research*, 17(15): 1–24, 2016.

Dongsheng Li, Chao Chen, Wei Liu, Tun Lu, Ning Gu, and Stephen Chu. Mixture-rank matrix approximation for collaborative filtering. In *Proceedings of the NIPS*, pages 477–485, 2017.

Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. *Proceedings of WWW*, 2018.

Huafeng Liu, Liping Jing, Yuhua Qian, and Jian Yu. Adaptive local low-rank matrix approximation for recommendation. *ACM Transaction on Information System*, 37(4), October 2019a.

Huafeng Liu, Jingxuan Wen, Liping Jing, and Jian Yu. Deep generative ranking for personalized recommendation. In *Proceedings of RecSys*, pages 34–42. ACM, 2019b.

Huafeng Liu, Jingxuan Wen, Liping Jing, Jian Yu, Xiangliang Zhang, and Min Zhang. In2rec: Influence-based interpretable recommendation. In *Proceedings of the CIKM*, pages 1803–1812, 2019c.

Huafeng Liu, Liping Jing, Jingxuan Wen, Zhicheng Wu, Xiaoyi Sun, Jiaqi Wang, Lin Xiao, and Jian Yu. Deep global and local generative model for recommendation. In *Proceedings of The Web Conference 2020*, page 551–561, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450370233.

Huafeng Liu, Liping Jing, Jingxuan Wen, Pengyu Xu, Jian Yu, and Michael K Ng. Bayesian additive matrix approximation for social recommendation. *ACM Transactions on Knowledge Discovery from Data*, 16(1):1–34, 2021a.

Huafeng Liu, Liping Jing, Jian Yu, and Michael Kwok-Po Ng. Social recommendation with learning personal and social latent factors. *IEEE Transactions on Knowledge and Data Engineering*, 33(7):2956–2970, 2021b.

Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. Learning disentangled representations for recommendation. In *Proceedings of NIPS*, pages 5712–5723, 2019.

Lester W Mackey, Michael I Jordan, and Ameet Talwalkar. Divide-and-conquer matrix factorization. In *Proceedings of NIPS*, pages 1134–1142, 2011.

Julian Mcauley and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of ACM RecSys*, pages 165–172, 2013.

Pascal Mettes, Elise van der Pol, and Cees GM Snoek. Hyperspherical prototype networks. *arXiv preprint arXiv:1901.10514*, 2019.

Andriy Mnih and Ruslan R Salakhutdinov. Probabilistic matrix factorization. In *Proceedings of NIPS*, pages 1257–1264, 2008.

Radford M. Neal. *Bayesian learning for neural networks*. Springer, 1996.

Radford M Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of computational and graphical statistics*, 9(2):249–265, 2000.

Haekyu Park, Hyunsik Jeon, Junghwan Kim, Beunguk Ahn, and U Kang. Uniwalk: explainable and accurate recommendation for rating and network data. In *Proceedings of SIAM CDM*. SIAM, 2018.

Michael J Pazzani and Daniel Billsus. Content-based recommendation systems. In *The Adaptive Web*, pages 325–341. Springer, 2007.

Jim Pitman. Combinatorial stochastic processes. Technical report, Technical Report, 2002.

Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. Restricted boltzmann machines for collaborative filtering. In *Proceedings of ICML*, pages 791–798. ACM, 2007.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of WWW*, pages 285–295. ACM, 2001.

Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *Proceedings of ACM RecSys*, pages 297–305, 2017.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Ruoyu Sun and Zhi-Quan Luo. Guaranteed matrix completion via non-convex factorization. *IEEE Transactions on Information Theory*, 62(11):6535–6579, 2016.

Alex Tank, Ian Covert, Nicholas Foti, Ali Shojaie, and Emily Fox. Neural granger causality for nonlinear time series, 2018.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 58(1):267–288, 1996.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. In *Proceedings of ICLR*, 2018.

Vladimir Vapnik and Rauf Izmailov. Learning using privileged information: similarity control and knowledge transfer. *Journal of Machine Learning Research*, 16(1):2023–2049, 2015.

Jesse Vig, Shilad Sen, and John Riedl. The tag genome: encoding community knowledge to support novel interaction. *ACM Transactions on Interactive Intelligent Systems*, 2(3), September 2012.

Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

Keqiang Wang, Wayne Xin Zhao, Hongwei Peng, and Xiaoling Wang. Bayesian probabilistic multi-topic matrix factorization for rating prediction. In *Proceedings of the IJCAI*, pages 3910–3916, 2016.

David P Wipf and Bhaskar D Rao. An empirical bayesian strategy for solving the simultaneous sparse approximation problem. *IEEE Transactions on Signal Processing*, 55(7): 3704–3716, 2007.

Yao Wu, Xudong Liu, Min Xie, Martin Ester, and Qing Yang. Cccf: Improving collaborative filtering via scalable user-item co-clustering. In *Proceedings of WSDM*, pages 73–82. ACM, 2016.

Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. Deep matrix factorization models for recommender systems. In *Proceedings of IJCAI*, pages 3203–3209, 2017.

Shuai Yu, Yongbo Wang, Min Yang, Baocheng Li, Qiang Qu, and Jialie Shen. Nairs: A neural attentive interpretable recommendation system. In *Proceedings of the ACM WSDM*, pages 790–793. ACM, 2019.

Dagmar Zeithamova. *Prototype Learning Systems*, pages 2715–2718. Springer US, Boston, MA, 2012. ISBN 978-1-4419-1428-6. doi: 10.1007/978-1-4419-1428-6_1628. URL `https://doi.org/10.1007/978-1-4419-1428-6_1628`.

Menghao Zhang, Binbin Hu, Chuan Shi, and Bai Wang. Local low-rank matrix approximation with preference selection of anchor points. In *Proceedings of WWW*, pages 1395–1403, 2017.

Yongfeng Zhang and Xu Chen. Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval*, 14(1):1–101, 2020.

Yongfeng Zhang, Min Zhang, Yiqun Liu, and Shaoping Ma. Improve collaborative filtering through bordered block diagonal form matrices. In *Proceedings of SIGIR*, pages 313–322. ACM, 2013.

Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of ACM SIGIR*, pages 83–92. ACM, 2014.

Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep interest network for click-through rate prediction. In *Proceedings of the SIGKDD*, KDD '18, page 1059–1068. Association for Computing Machinery, 2018.