Disentangled Representation for Diversified Recommendations

Xiaoying Zhang
AI Lab, Bytedance
China
zhangxiaoying.xy@bytedance.com

Hongning Wang
Department of Computer Science
University of Virginia, USA
hw5x@virginia.edu

Hang Li
AI Lab, Bytedance
China
lihang.lh@bytedance.com

ABSTRACT

Accuracy and diversity have long been considered to be two conflicting goals for recommendations. We point out, however, that as the diversity is typically measured by certain pre-selected item attributes, e.g., category as the most popularly employed one, improved diversity can be achieved without sacrificing recommendation accuracy, as long as the diversification respects the user's preference about the pre-selected attributes. This calls for a fine-grained understanding of a user's preferences over items, where one needs to recognize the user's choice is driven by the quality of the item itself, or the pre-selected attributes of the item.

In this work, we focus on diversity defined on item categories. We propose a general diversification framework agnostic to the choice of recommendation algorithm. Our solution disentangles the learnt user representation in the recommendation module into category-independent and category-dependent components to differentiate a user's preference over items from two orthogonal perspectives. Experimental results on three benchmark datasets and online A/B test demonstrate the effectiveness of our solution in improving both recommendation accuracy and diversity. In-depth analysis suggests that the improvement is due to our improved modeling of users' categorical preferences and refined ranking within item categories.

CCS CONCEPTS

• **Information systems** → **Information retrieval diversity**; *Collaborative filtering*.

KEYWORDS

Recommender system, recommendation diversity, disentangled user representation

ACM Reference Format:
Xiaoying Zhang, Hongning Wang, and Hang Li. 2023. Disentangled Representation for Diversified Recommendations. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (WSDM '23), February 27-March 3, 2023, Singapore, Singapore.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3539597.3570389

1 INTRODUCTION

Recommender systems learn users' interests from historical observations (e.g., their clicks, bookmarked or purchased items, etc.) so as to identify the items that best suit users' preferences. The success of recommender system in enhancing user experience and boosting platform utility has been witnessed in a number of scenarios including e-commerce [17, 42], online news recommendation [33] and streaming services [9].

Recommendation accuracy, which measures whether a recommendation model can recommend items that users will like, serves as the dominant target or even the only target in most previous work [9, 16, 17, 31, 42]. Various complicated models [9, 16, 42] have been proposed for higher accuracy. While recommendation accuracy has been shown to be closely related to user satisfaction, it is never the only rule of thumb. Recent work found the recommendation diversity, which measures the dissimilarity among recommended items regarding certain pre-selected item attributes (e.g., item category) also plays an important role in the overall user experience [18, 32, 43]. For example, even if a user is a fan of basketball, he/she can still get bored with recommendations only about basketball videos or news, which increases the risk of user attrition.

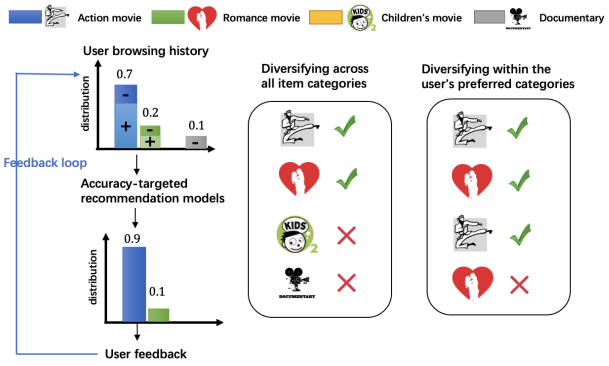

Figure 1: Illustration of recommendation accuracy and diversity optimization in different recommendation models.

Following previous work [30, 31, 40], we focus on diversity defined on item categories in this paper and aim to address the so-called accuracy-diversity dilemma [40]. On one hand, recommendation models with accuracy as their primary target often lose diversity to some extent, due to overly emphasizing items in the dominant categories in a user's interaction history [30, 31]. Figure 1(a) illustrates this issue with an example in movie recommendation, where 70% of the movies watched by a user are action movies, which leads 90% of the system's recommendations to fall in the action movie category. Worse still, because of the feedback loop [4], the emphasis on the dominant categories in the system's recommendations will be further intensified when the user follows the recommendations, causing further decreased recommendation diversity and issues like filter bubbles [25] and echo chambers [14]. On the other hand, simply diversifying recommendations over all item categories without considering the user's categorical preference

WSDM '23, February 27-March 3, 2023, Singapore, Singapore

ACM ISBN 978-1-4503-9407-9/23/02…$15.00
https://doi.org/10.1145/3539597.3570389

hurts the accuracy of generated recommendations [27, 32, 40, 44]. As shown in Figure 1(b), although the recommendation list is diverse by covering all four categories, negative feedback is more likely on the categories where the user interacted less often or negative feedback already prevailed, e.g., children's movies and documentary movies respectively in this example.

Clearly one should not recklessly increase diversity. For categories the user is less likely to be interested in, the risk of making a bad recommendation overweights the benefit of increased diversity. Thus, this paper focuses on conducting diversification only among item categories that the user prefers, suggesting the possibility to improve recommendation diversity without sacrificing recommendation accuracy. Figure 1(c) gives an example recommendation list following such a strategy, where the recommended items mainly fall in action and romance movies, the two preferred categories inferred from the user's interaction history. This strategy requires the recommendation model to clearly distinguish whether the user's positive/negative feedback is due to the item's category or other category-independent features of the item (e.g., the item's own quality), which was ignored by previous recommendation models.

In this paper, we propose a general and model-agnostic framework to disentangle a user's category-dependent and category-independent preferences for an accurate and diversified recommender system (DCRS). Specifically, DCRS takes a user's preference over an item as a product of: (1) the user's preference over the item's category; and (2) the user's preference over category-independent features of the item, e.g., the item's quality. Such disentanglement suggests a hierarchical decision making process by the user: If a user has a strong preference over a particular category of items, he/she may still enjoy items of this category, even though their qualities are not perfect. However, if the probability that a user likes a category is low, only items of high quality in this category could have a chance to be considered. The disentanglement ensures items of the same quality, but in different categories that a user prefers similarly, have equal probabilities to be recommended. It naturally avoids overly recommending items from the dominant categories in the user's interaction history. The main challenge therefore lies in how to disentangle a user's preference regarding the aforementioned two orthogonal perspectives, given his/her preference over the item categories is not observable. This makes naive solutions like using different supervision signals to separately train users' representations [41], or separating items' feature vectors into category dependent and independent segments, ineffective.

DCRS is agnostic to the choice of recommendation module, which is supposed to learn informative representations of users and items. In particular, DCRS adopts a discriminator to disentangle the learnt representation into category-independent and category-dependent segments respectively. The recommendation module and discriminator are learnt simultaneously to ensure the effectiveness of disentangled representation learning for accurate and diverse recommendations. To evaluate the proposed DCRS solution, we conduct both offline experiments on three benchmark datasets and online A/B test on Toutiao app, one of the largest news recommendation platforms in China. Experiment results demonstrate that DCRS can successfully recommend diverse items that users prefer, and thus improve both recommendation accuracy and diversity. In-depth analysis and case studies suggest strong evidence showing: (1) the disentangled category-independent representation from DCRS can distinguish the user's preference within category more accurately; and (2) DCRS can capture a user's diverse preferences in historical interactions more thoroughly. All codes and data can be found in https://github.com/Xiaoyinggit/DCRS.git.

Overall, our contribution of this work is as follows:

- We demonstrate that accuracy and diversity are not conflicting goals for recommendation, as long as the diversification respects the user's categorical preference.
- To capture a user's latent preferences on item categories more accurately, our proposed DCRS disentangles the user's preference into category-dependent and category-independent components.
- Experiments on three benchmark datasets and online A/B test demonstrate the effectiveness of DCRS in improving both recommendation accuracy and diversity. In-depth analysis further demonstrates the improvement comes from more accurate modeling of the user's preference both over and within categories.

2 FRAMEWORK

In this section, we describe how the proposed DCRS solution disentangles a user's category dependent and independent preferences to simultaneously improve recommendation accuracy and diversity. For the ease of illustration, we first briefly describe a general architecture which covers almost all popularly used recommendation models. We then depict how to smoothly integrate DCRS into such a general architecture to diversify its recommendations.

2.1 Preliminary: A General Recommendation Architecture

In a recommendation task, we are given a user behavior dataset $\boldsymbol{\mathcal{X}}$ that contains interactions between N users and M items. The interaction between user u and item i is represented as a tuple $(u, i, y_{u,i}) \in \boldsymbol{\mathcal{X}}$. Here $y_{u,i} \in \{0, 1\}$ denotes user u's feedback to item i, where $y_{u,i} = 1$ denotes positive feedback (e,g., a click or a positive rating), and $y_{u,i} = 0$ denotes negative feedback. Generally speaking, a recommendation model will first learn a user-item representation to capture the user's preference over the item:

$$\boldsymbol{h}_{u,i} = f(u, i, \boldsymbol{\theta}) \in R^d, \tag{1}$$

where $\boldsymbol{\theta}$ denotes a set of trainable parameters in the recommendation model. Various architectures [16, 17, 31, 42] have been proposed to implement $f(u, i, \boldsymbol{\theta})$, ranging from the simple matrix factorization algorithm [23] that directly takes the element-wise product of user and item embeddings to form the representation, to complex architectures such as the bi-interaction layer in NFM [17]. Let $\hat{p}_{u,i}$ denote the probability that user u gives positive feedback to item i. The goal of the recommendation model is to use the learnt user-item representation to estimate $\hat{p}_{u,i}$, either by directly summing up elements in $\boldsymbol{h}_{u,i}$ as in matrix factorization, or through a learnable projection layer as follows:

$$\hat{p}_{u,i} = P\left(Y_{u,i} = 1 | u, i\right) = \sigma\left(\boldsymbol{W}^\top \boldsymbol{h}_{u,i}\right), \tag{2}$$

where $Y_{u,i}$ is a random variable representing the feedback from user u on item i; $\boldsymbol{W} \in R^{d \times 1}$ is the learnable weight vector of the projection layer, and $\sigma(\cdot)$ is the sigmoid function. The parameters of the recommendation model are then optimized by minimizing

the following loss:

$$\mathcal{L}(\boldsymbol{X}, \boldsymbol{\theta}, \boldsymbol{W}) = \frac{1}{|\boldsymbol{X}|} \sum_{(u,i,y_{u,i}) \in \boldsymbol{X}} \mathcal{L}_{rec}(y_{u,i}, \hat{p}_{u,i}), \tag{3}$$

where $\mathcal{L}_{rec}(\cdot,\cdot)$ represents the chosen loss function. Various loss functions have been explored in literature, inlcuding cross entropy loss, Mean Squared Error (MSE) and BPR loss [28]. In this work, we will use the cross entropy loss by default.

2.2 Disentangle Category Dependent and Independent Representations

We consider a user's feedback on an item as a mixture reflecting his/her preference over the item's category and category-independent properties, e.g., the item's intrinsic quality. As shown in Figure 2, the first action movie that receives positive feedback can very likely be caused by the user's strong preference over the category of action movies, while his/her positive feedback on the second romance movie is more likely to be caused by its high quality that makes up the low probability that the user likes romance movies. In order to diversify the recommendations with respect to a user's preferred categories, the recommendation model needs to clearly distinguish the effect of item category and other category-independent properties on a user's decision making. To make our method description general enough to cover situations where an item can associate with multiple categories, we take item i's category as the set that contains all categories that the item relates to, and denote it as t_i. For example, assume there are three categories $\{c_1, c_2, c_3\}$ in a dataset. If item i is related to the first category, then $t_i = \{c_1\}$. And if item i is associated with the first two categories, then $t_i = \{c_1, c_2\}$.

We propose to disentangle a user's preference over an item into two parts :

- **Category-dependent preference**: it captures the user's preference over the item's category;
- **Category-independent preference**: it depicts how category-independent features affect the user's preference about the item.

Such a disentanglement can be explained through a probabilistic view about the generation of user u's feedback on item i. Let $Y^C_{u,i}$ denote the binary random variable indicating user u's feedback on item i's category. We have the following,

$$P(Y_{u,i} = 1|u,i) = P(Y_{u,i} = 1, Y^C_{u,i} = 1|u,i) \tag{4a}$$

$$= P(Y_{u,i} = 1|u,i, Y^C_{u,i} = 1)P(Y^C_{u,i} = 1|u,i) \tag{4b}$$

$$= P(Y_{u,i} = 1|u,i, Y^C_{u,i} = 1)P(Y^C_{u,i} = 1|u,t_i) \tag{4c}$$

In particular, Eq.(4a) is due to the assumption that user u gives positive feedback to item i only if user u likes item i's category, i.e., $P(Y_{u,i} = 1, Y^C_{u,i} = 1|u,i) = 1$ and $P(Y_{u,i} = 1, Y^C_{u,i} = 0|u,i) = 0$. Eq.(4b) follows the chain rule. And Eq.(4c) is because $Y^C_{u,i}$ only depends on the item's category, instead of specific items.

The first term in Eq.(4c) depicts how likely user u will give positive feedback to item i when he/she is interested in item i's category; and the second term models how likely user u is interested in item i's category. Given that user u likes the category of item i, the probability in the first term only depends on the category-independent features of item i, such as item i's quality, price, etc. Thus, under the disentangled user-item representations, we can compute the first term by the probability $P(Y^{\perp C}_{u,i} = 1|u,i)$, which depicts user u's preference over item i driven by the *category-independent features*. Thus, Eq.(4c) can be rewritten as :

$$P(Y_{u,i} = 1|u,i) = P(Y^{\perp C}_{u,i} = 1|u,i)P(Y^C_{u,i} = 1|u,t_i). \tag{5}$$

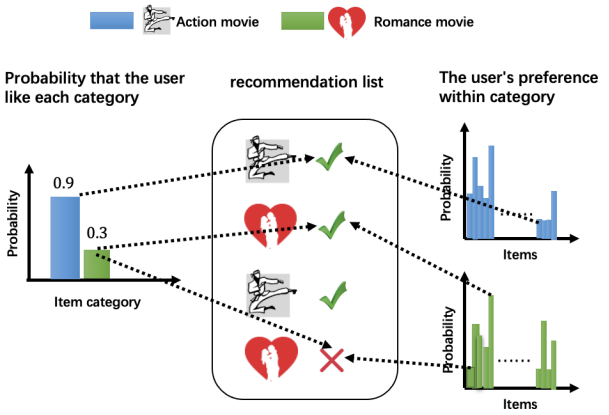

Figure 2: Hierarchical decision making process of DCRS framework. Each feedback is determined by: (1) the user's preference over the item's category; and (2) the user's preference over category-independent features of the item.

Eq.(5) depicts a hierarchical decision making process illustrated in Figure 2. If user u likes item i's category with a higher probability $P(Y^C_{u,i} = 1|u,t_i)$, he/she may still enjoy item i even though item i's quality is not perfect, indicated by a lower $P(Y^{\perp C}_{u,i} = 1|u,i)$. For example, the positive feedback of the first action movie in Figure 2 is generated under such a scenario. Meanwhile, if there is only a small probability that user u would be interested in item i's category (i.e., low $P(Y^C_{u,i} = 1|u,t_i)$), item i must be of high quality to get positive feedback, i.e., high $P(Y^{\perp C}_{u,i} = 1|u,i)$. The positive feedback on the second romance movie in Figure 2 is a good example of this case.

Eq.(5) also suggests why disentanglement makes recommendations diversified within a user's preferred categories. Assume there are two categories c_1 and c_2 on which the user has similar preference. Instead of recommending more items from the dominant category (either c_1 or c_2), via the disentanglement in Eq.(5), items of the same quality within c_1 and c_2 will have an equal chance to be recommended, thus diversifying the recommendations.

Unfortunately, both terms in Eq.(5) cannot be learnt via direct supervision signals, since user u's feedback driven solely by item i's category or its category-independent features cannot be observed. Classical solutions would appeal to Expectation Maximization type algorithms [10] to estimate the two terms in an iterative manner. However, given modern recommendation algorithms are usually realized via complex deep neural networks, posterior inference becomes cumbersome and also leads to slow convergence. Instead, DCRS implements Eq.(5) by simultaneously learning two disentangled representations for estimating the two terms separately. Specifically, DCRS learns two disentangled representations by:

$$\left[\left(\boldsymbol{h}^{\perp C}_{u,i}\right)^\top, \left(\boldsymbol{h}^{C}_{u,i}\right)^\top\right]^\top = f(u,i,\boldsymbol{\theta}) \in R^{2d}, \tag{6}$$

where $\boldsymbol{h}^{\perp C}_{u,i} \in R^d$ aims to capture user u's preference over category-independent features to estimate $P(Y^{\perp C}_{u,i} = 1|u,i)$, and $\boldsymbol{h}^{C}_{u,i} \in R^d$

depicts user u's preference over item i's category t_i, aiming to estimate $P(Y^C_{u,i} = 1|u, t_i)$.

Simply splitting item i's feature vector into two parts, even with separate networks, cannot ensure complete disentanglement. Instead, in addition to requiring the learnt the representations to best capture the user's preference, we employ an adversarial discriminator that enforces the learnt $\boldsymbol{h}^{\perp C}_{u,i}$ and $\boldsymbol{h}^{C}_{u,i}$ to be category-independent and category-dependent respectively.

Discriminator Module. The discriminator $D(\cdot)$ acts as a category classifier, which takes one segment of disentangled representation, such as $\boldsymbol{h}^{C}_{u,i}$ or $\boldsymbol{h}^{\perp C}_{u,i}$, as input, and aims to predict the category of item i (i.e., t_i). However, it is hard for the discriminator to directly predict t_i, since t_i can take 2^K-1 values, where K is the number of unique categories available in the dataset. For ease of learning, we represent t_i by a vector over K unique categories, denoted as $\tilde{\boldsymbol{t}}_i$. Again, assume there are three categories $\{c_1, c_2, c_3\}$, if $t_1 = \{c_1\}$, then $\tilde{\boldsymbol{t}}_i = [1, 0, 0]^\top$. And if $t_1 = \{c_1, c_2\}$, then $\tilde{\boldsymbol{t}}_i = [0.5, 0.5, 0]^\top$. Specifically, when relevance between item i and each associated category can be measured [26], a more accurate $\tilde{\boldsymbol{t}}_i$ can be achieved by making the j-th element of $\tilde{\boldsymbol{t}}_i$ proportional to the relevance between item i and the j-th category. Otherwise, $\tilde{\boldsymbol{t}}_i$ can be simply assumed to be evenly distributed among related categories, which is also the default setting in our experiments. The discriminator then takes $\boldsymbol{h}^{C}_{u,i}$ or $\boldsymbol{h}^{\perp C}_{u,i}$ as input to predict $\tilde{\boldsymbol{t}}_i$. In our experiments, the discriminator $D(\cdot)$ is implemented via a fully connected layer, and it should enforce the following:

- Given $\boldsymbol{h}^{C}_{u,i}$ is closely related to item i's category, the discriminator should predict $\tilde{\boldsymbol{t}}_i$ accurately based on $\boldsymbol{h}^{C}_{u,i}$, i.e., the following loss should be minimized:
$$\min \mathcal{L}^{C}_{D}(u, i) = \mathcal{L}_{CE}\left(D(\boldsymbol{h}^{C}_{u,i}), \tilde{\boldsymbol{t}}_i\right),$$
where $\mathcal{L}_{CE}$ is the cross entropy loss.
- Given $\boldsymbol{h}^{\perp C}_{u,i}$ is independent from item category, $\boldsymbol{h}^{\perp C}_{u,i}$ should *fool* the discriminator by maximizing the classification loss:
$$\max \mathcal{L}^{\perp C}_{D}(u, i) = \mathcal{L}_{CE}\left(D(\boldsymbol{h}^{\perp C}_{u,i}), \tilde{\boldsymbol{t}}_i\right).$$

We leverage a Gradient Reverse Layer (GRL) [13] to implement above requirements due to its simplicity. More specifically, we insert a Gradient Reverse Layer between $\boldsymbol{h}^{\perp C}_{u,i}$ and the discriminator, as shown in Figure 3. During back propagation, the gradients for minimizing the discriminator loss $\frac{\partial \mathcal{L}^{\perp C}_{D}(u,i)}{\partial \boldsymbol{h}^{\perp C}_{u,i}}$ flow backward through the discriminator. After the GRL, the gradients will be reversed, i.e., becoming $-\frac{\partial \mathcal{L}^{\perp C}_{D}(u,i)}{\partial \boldsymbol{h}^{\perp C}_{u,i}}$. Thus, we perform gradient descent on parameters of the discriminator for accurately predicting item i's category, while performing gradient ascent on $\boldsymbol{h}^{\perp C}_{u,i}$, so that $\boldsymbol{h}^{\perp C}_{u,i}$ cannot predict item i's category.

Learning category-independent representation. $\boldsymbol{h}^{\perp C}_{u,i}$ should be optimized under two objectives: (1) it can accurately estimate the first term $P(Y^{\perp C}_{u,i} = 1|u, i)$ in Eq.(5) by:
$$\hat{p}^{\perp C}_{u,i} = P\left(Y^{\perp C}_{u,i} = 1|u, i\right) = \sigma\left(\boldsymbol{W}_1{}^\top \boldsymbol{h}^{\perp C}_{u,i}\right); \tag{7}$$
and (2) it needs to be independent from item categories. Thus we minimize the following loss for its learning:
$$\mathcal{L}_{rec}\left(\hat{p}^{\perp C}_{u,i}, y_{u,i}\right) - \lambda \mathcal{L}^{\perp C}_{D}(u, i) \tag{8}$$

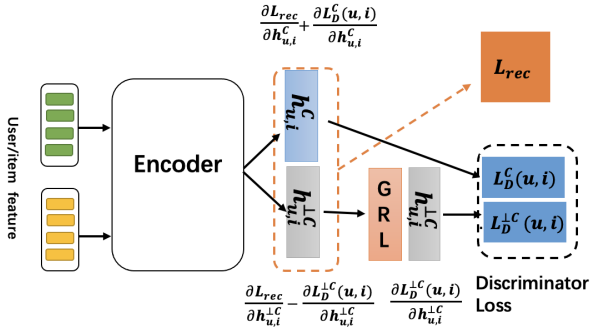

Figure 3: The architecture of DCRS, which disentangles the user u's preference on item i into category-dependent segment $\boldsymbol{h}^{C}_{u,i}$ and category-independent segment $\boldsymbol{h}^{\perp C}_{u,i}$ for diverse and accurate recommendations.

where the two terms optimize two distinct objectives respectively, and λ is a hyper-parameter that controls the strength of category-independent constraint on $\boldsymbol{h}^{\perp C}_{u,i}$.

Learning category-dependent representation. While user u's preference on item i's category is unobservable, $P(Y^{C}_{u,i} = 1|u, t_i)$ can be estimated by fixing the learnt category-independent representation $\boldsymbol{h}^{\perp C}_{u,i}$ and estimating the overall probability that user u gives positive feedback to item i:
$$\hat{p}_{u,i} = P(Y_{u,i} = 1|u, i) = \sigma\left(\boldsymbol{W}_2{}^\top \begin{bmatrix} \text{stop\_gradient}(\boldsymbol{h}^{\perp C}_{u,i}) \\ \boldsymbol{h}^{C}_{u,i} \end{bmatrix}\right), \boldsymbol{W}_2 \in R^{2d \times 1} \tag{9}$$
where $\text{stop\_gradient}\left(\boldsymbol{h}^{\perp C}_{u,i}\right)$ implies that $\boldsymbol{h}^{\perp C}_{u,i}$ will not be updated by this prediction. In other words, given the learnt user u's preference over category-independent features of item i, only user u's preference over item i's category is optimized to accurately predict the overall feedback of user u to item i, by minimizing the loss:
$$\mathcal{L}_{rec}\left(\hat{p}_{u,i}, y_{u,i}\right) + \lambda \mathcal{L}^{C}_{D}(u, i), \tag{10}$$
where the second loss forces $\boldsymbol{h}^{C}_{u,i}$ to predict item i's category accurately with λ representing the strength of the constraint.

Overall, combining Eq.(8) and Eq.(10), given a user behavior dataset $\boldsymbol{X}$, DCRS learns a disentangled recommendation model as in Eq.(5) by minimizing the following loss:
$$\begin{aligned}\mathcal{L}(\boldsymbol{X}, \boldsymbol{\theta}, \boldsymbol{W}_1, \boldsymbol{W}_2) = &\frac{1}{|X|} \sum_{(u,i,y_{u,i}) \in \boldsymbol{X}} \mathcal{L}_{rec}\left(\hat{p}_{u,i}, y_{u,i}\right) \\ &+ \mathcal{L}_{rec}\left(\hat{p}^{\perp C}_{u,i}, y_{u,i}\right) - \lambda \mathcal{L}^{\perp C}_{D}(u, i) + \lambda \mathcal{L}^{C}_{D}(u, i).\end{aligned}$$

Inference. At the inference stage, we leverage $\hat{p}_{u,i}$ in Eq.(9) as the predicted preference of user u over item i to rank items. We adopt $\hat{p}_{u,i}$ in Eq.(9) since it considers both the category dependent and independent preference of the user, while $\hat{p}^{\perp C}_{u,i}$ in Eq.(7) only captures the user's preference over category-independent features.

3 OFFLINE EXPERIMENTS

In this section, we conduct experiments on several public offline datasets to demonstrate the effectiveness of DCRS. We mainly investigate from two perspectives:

- How does the proposed DCRS perform in terms of recommendation accuracy and diversity?

Table 1: Statistics of Three Datasets

Dataset	#Users	#Items	#Interactions	#Group
ML-1M	6040	3883	1000209	18
ML-10M	69878	10680	10000047	19
Amazon-Books	22929	33130	1178117	141

- Can the disentangled category-independent representation accurately distinguish a user's preference within item categories?

A case study is also conducted to illustrate the effectiveness of the proposed DCRS more explicitly.

3.1 Experimental Settings

Dataset. We use three widely-used datasets under different recommendation scenarios for evaluation.

- **ML-1M**[1]**:** This dataset contains 1 million ratings from 6040 users on 3883 movies from the online movie recommendation service MovieLens. It also contains rich user features (e.g., age, gender, etc.) and movie features (e.g., titles). We encode user and movie features following previous work [31, 42]. We take $y_{u,i} = 1$, if user u gives item i a rating greater than 3, otherwise $y_{u,i} = 0$.
- **ML-10M**[2]**:** This dataset is also from MovieLens. It contains 10 million ratings from 69878 users on 10680 movies. Similarly, we take $y_{u,i} = 1$, if user u gives item i a rating greater than 3, otherwise $y_{u,i} = 0$.
- **Amazon-Books**[3]**:** This dataset contains reviews and metadata of books from Amazon. To ensure data quality, we only keep categories that link to more than 20 books with 141 categories, and adopt the 20-core settings [31], i.e., discarding users and books with less than 20 interactions. To make the number of positive and negative samples balanced, we take $y_{u,i} = 1$, if user u gives item i a rating greater than 4, otherwise $y_{u,i} = 0$.

The statistics of the three datasets are summarized in Table 1. On each dataset, we also randomly sampled items that the user did not interact with as negative instances. We then sorted the user-item interactions by timestamps, and split them into training, validation, and testing datasets with the ratio of 80%, 10%, and 10%.

Baselines. The proposed DCRS is a general and model-agnostic framework to disentangle category dependent and independent representations for accurate and diverse recommendations. In this paper, we instantiated it with Neural Factorization Machine (NFM) [17], one representative recommendation model that has been widely used. NFM was also taken as the backbone model in several closely related work for diversified recommendations [15, 31]. We compared DCRS with the following algorithms that have different focuses on recommendation diversity and accuracy.

- **NFM [17]**: The state-of-the-art recommendation model serving as the backbone model of DCRS.
- **Unawareness [15]**: It also takes NFM as the backbone model and tries to improve diversity by directly removing categorical features of items from model input.
- **IPS [29]**: It is a state-of-the-art technique of improving diversity by boosting item categories that a user interacted with less often, while suppressing the dominant categories in the user's interaction history. Specifically, it takes the category distribution in a user's historical interactions as propensity scores to reweigh items of this category during training. Propensity clipping [29] is also employed to reduce the variance with clipping threshold searched in {0.001, 0.005, 0.01, 0.05, 0.1}.
- **MMR [3]**: One of the state-of-art post-processing methods for diversified recommendations. It re-ranks the recommended items generated by NFM by a greedy strategy to reduce redundancy.
- **DPP [5]**: An effective post-processing method for diversified recommendations. It selects a diverse set of items from the recommended items generated by NFM by balancing the relevance of items and their similarities.
- **PD_GAN [36]**: A recent work that leverages the generative adversarial networks (GAN) framework to generate diverse and relevant recommendations. Its discriminator aims to distinguish the generated diverse set of items by its generator from the ground-truth sets randomly sampled from the observed data of the user.
- **DGCN [40]**: A recent work that leverages rebalanced neighbor discovering, category-boosted negative sampling and adversarial learning on top of Graph Convolutional Networks (GCN) for diversified recommendations.
- **DecRS [31]**: A recent work for alleviating the bias that previous recommendation models over-recommend items of the dominant categories in a user's interaction history from a causal view. It aims at improving both recommendation accuracy and diversity.
- **DCRS_CI**: A variant of DCRS that leverages $\hat{p}_{u,i}^{\perp C}$ in Eq.(7) for item ranking without considering the user's preference over categories. Its comparison with DCRS_CI can reveal the importance of modeling users' categorical preference.

Implementation Details. Following previous work [17, 31], we set the embedding size of user/item features to 64 (i.e., $d = 64$), and used AdaGrad [11] for optimization. We used grid search to select the hyperparameters based on the model's performance on validation dataset: the learning rate was searched in {0.005, 0.01, 0.05}; the normalization coefficient was searched in {0, 0.1, 0.2}; the dropout ratio was searched in {0.2, 0.3, ..., 0.5}; λ for controlling strength of category independent and dependent constraints was searched in {0.01, 0.05, 0.1, 0.5, 1}. For baseline algorithms, when evaluating on the dataset the algorithms were also evaluated in their original papers, we adopted the recommended hyperparameters from the original paper; otherwise we performed a similar grid search as above with the search range following the original paper.

3.2 Performance on Recommendation Accuracy & Diversity

We first evaluate all algorithms in terms of recommendation accuracy and diversity.

Evaluation Metrics. We evaluate the accuracy of a recommendation model from two perspectives: (1) Whether the model can rank positively interacted items of a user before those negatively interacted ones accurately in the testing dataset; (2) Whether the model can accurately retrieve those positively interacted items in the testing dataset from the item pool, which includes all items that the user did not interact with in the training dataset. For MMR and DPP, because they only re-rank the recommended items generated by NFM, a specifically created item pool that contains top-200 items of NFM is used. We adopted AUC [12] and UAUC [42] as metrics to evaluate the first perspective. Basically, UAUC is a micro-average

[1]https://grouplens.org/datasets/movielens/1m/
[2]https://grouplens.org/datasets/movielens/10m/
[3]https://jmcauley.ucsd.edu/data/amazon/

Dataset	Method	AUC	UAUC	RelaImpr	R@10	NDCG@10	CE@10	CC@10	R@20	NDCG@20	CE@20	CC@20
ML_1M	NFM	0.8461	0.8224	0.00%	0.0522	0.0572	1.8056	0.4741	0.0908	0.0681	1.9764	0.6185
	UnAwareness	0.8414^-	0.8134^-	-2.79%	0.0512^-	0.0568^-	1.8919^+	0.4998^+	0.0880^-	0.0669^-	2.0513^+	0.6419^+
	IPS	0.8446^-	0.8210^-	-0.43%	0.0513^-	0.0572	1.7929^-	0.4713^-	0.0890^-	0.0681	1.9759^-	0.6225^+
	MMR	–	0.8194^-	-0.93%	0.0501^-	0.0545^-	2.1279^+	0.5886^+	0.0902^-	0.0670^-	2.2224^+	0.7244^+
	DPP	–	0.6021^-	-68.3%	0.0454^-	0.0518^-	2.4119^+	0.7315^+	0.0770^-	0.0601^-	2.5974^+	0.9586^+
	PD_GAN	–	–	–	0.0326^-	0.0347^-	2.5495^+	0.8347^+	0.0503^-	0.0386^-	2.6650^+	0.9393^+
	DGCN	0.7949^-	0.7759^-	-14.4%	0.0365^-	0.0402^-	1.9133^+	0.5088^+	0.0640^-	0.0482^-	2.0748^+	0.6466^+
	DecRS	0.8462^+	0.8202^-	-0.07%	0.0537^+	0.0588^+	1.8560^+	0.4876^+	0.0919^+	0.0694^+	2.0378^+	0.6365^+
	DCRS_CI	0.8332^-	0.8096^-	-3.90%	0.0530^+	0.0581^+	1.7606^-	0.4468^-	0.0936^+	0.0699^+	1.9108^-	0.5766^-
	DCRS	$\mathbf{0.8483^+}$	$\mathbf{0.8237^+}$	**0.40%**	$\mathbf{0.0551^+}$	$\mathbf{0.0602^+}$	$\mathbf{1.8877^+}$	$\mathbf{0.4909^+}$	$\mathbf{0.096^+}$	$\mathbf{0.0722^+}$	$\mathbf{2.0620^+}$	$\mathbf{0.6368^+}$
ML_10M	NFM	0.8346	0.8193	0.00%	0.0474	0.0448	1.9351	0.5127	0.0797	0.0547	2.0877	0.6504
	UnAwareness	0.8274^-	0.8078^-	-3.60%	0.0394^-	0.0363^-	2.0308^+	0.5410^+	0.0659^-	0.0446^-	2.2036^+	0.6891^+
	IPS	0.8378^+	0.8218^+	0.78%	0.0469^-	0.0441^-	1.9280^-	0.5070^-	0.0783^-	0.0538^-	2.0913^+	0.6491^-
	MMR	–	0.8084^-	-3.41%	0.0436^-	0.0418^-	2.2941^+	0.6629^+	0.0762^-	0.0521^-	2.3451^+	0.7639^+
	DPP	–	0.6459^-	-54.3%	0.0390^-	0.0392^-	2.5014^+	0.7740^+	0.0629^-	0.0459^-	2.6248+	0.9376^+
	PD_GAN	–	–	–	0.0108^-	0.0119^-	2.3134^+	0.7164^+	0.0176^-	0.0136^-	2.4446^+	0.8606^+
	DGCN	0.8069^-	0.8081^-	-3.51%	0.0425^-	0.0380^-	2.0459^+	0.5530^+	0.0740^-	0.0482^-	2.1925^+	0.6934^+
	DecRS	0.8417^+	0.8261^+	2.12%	0.0477^+	0.0445^-	1.9401^+	0.5048^-	0.0814^+	0.0551^+	2.1181^+	0.6480^-
	DCRS_CI	0.8357^+	0.8197^+	0.12%	0.0478^+	0.0448	1.9810^+	0.5269^+	0.0813^+	0.0554^+	2.1322^+	0.6635^+
	DCRS	$\mathbf{0.8447^+}$	$\mathbf{0.8301^+}$	**3.38%**	$\mathbf{0.0499^+}$	$\mathbf{0.0465^+}$	$\mathbf{2.0050^+}$	$\mathbf{0.5327^+}$	$\mathbf{0.0838^+}$	$\mathbf{0.0572^+}$	$\mathbf{2.1655^+}$	$\mathbf{0.6733^+}$
Amazon-Books	NFM	0.6667	0.6289	0.00%	0.0076	0.0052	1.6722	0.0495	0.0118	0.0066	1.9551	0.0740
	UnAwareness	0.6267^-	0.5687^-	-46.7%	0.0064^-	0.0043^-	1.6660^-	0.0524^+	0.0097^-	0.0054^-	1.8762^-	0.0721^-
	IPS	0.6650^-	0.6269^-	-1.55%	0.0078^+	0.0053^+	1.5969^-	0.0453^-	0.0115^-	0.0066	1.9148^-	0.0704^-
	MMR	–	0.6096^-	-15.0%	0.0067^-	0.0045^-	2.2899^+	0.0864^+	0.0109^-	0.0060^-	2.5119^+	0.1278^+
	DPP	–	0.5300^-	-76.7%	0.0054^-	0.0040^-	2.5184^+	0.1005^+	0.0081^-	0.0049^-	2.8741+	0.1645^+
	PD_GAN	–	–	–	0.0004^-	0.0003^-	2.3179^+	0.0920^+	0.0016^-	0.0007^-	2.7648^+	0.1545^+
	DGCN	0.6747^+	0.6404^+	8.92%	0.0071^-	0.0044^-	2.0003^+	0.0698^+	0.0122^+	0.0061^-	2.2842^+	0.1074^+
	DecRS	0.6964^+	0.6558^+	20.8%	0.0074^-	0.0051^-	1.8207^+	0.0601^+	0.0111^-	0.0063^-	2.0973^+	0.0918^+
	DCRS_CI	0.6893^+	0.6546^+	19.9%	0.0057^-	0.0036^-	2.0172^+	0.0704^+	0.0095^-	0.0049^-	2.2799^+	0.1068^+
	DCRS	$\mathbf{0.6974^+}$	$\mathbf{0.6573^+}$	**22.0%**	$\mathbf{0.0079^+}$	**0.0052**	$\mathbf{1.8639^+}$	$\mathbf{0.0622^+}$	$\mathbf{0.0123^+}$	$\mathbf{0.0067^+}$	$\mathbf{2.1415^+}$	$\mathbf{0.0953^+}$

Table 2: Experimental results regarding to recommendation accuracy and diversity. Improved (or dropped) performance over the base NFM model under the same setting is marked as + (or −).

		ML-1M				**ML-10M**				**Amazon-Books**			
Category	Method	AUC	UAUC	R@20	NDGG@20	AUC	UAUC	R@20	NDGG@20	AUC	UAUC	R@20	NDGG@20
1^{st} ranked cat	NFM	0.8547	0.8180	0.3034	0.1678	0.8498	0.8229	0.2814	0.1453	0.6474	0.5976	0.0679	0.0343
	DecRS	0.8563	0.8135	0.3079	0.1718	0.8545	0.8273	0.3031	0.1572	0.6724	0.6113	0.0698	0.0331
	DCRS_CI	**0.8606**	**0.8241**	**0.3230**	**0.1783**	**0.8608**	**0.8340**	**0.3210**	**0.1659**	**0.6730**	**0.6178**	**0.0730**	**0.0345**
2^{nd} ranked cat	NFM	0.8403	0.8009	0.3820	0.1962	0.8372	0.8078	0.3434	0.1655	0.6637	0.5489	0.0536	0.0302
	DecRS	0.8407	0.8014	0.3817	0.1982	0.8407	0.8088	0.3558	0.1718	0.6978	0.5661	0.0576	0.0310
	DCRS_CI	**0.8449**	**0.8056**	**0.3960**	**0.2078**	**0.8485**	**0.8170**	**0.3861**	**0.1875**	**0.7413**	**0.5709**	**0.0595**	**0.0312**
3^{rd} ranked cat	NFM	0.8344	0.8046	0.6665	0.3350	0.8172	0.7926	0.4156	0.1969	0.6931	0.5910	0.0554	0.0241
	DecRS	0.8381	**0.8062**	0.6743	0.3423	0.8231	0.7968	0.4458	0.2137	0.7044	0.5920	0.0548	0.0233
	DCRS_CI	**0.8419**	0.8055	**0.6873**	**0.3463**	**0.8264**	**0.8014**	**0.4794**	**0.2291**	**0.7176**	**0.6162**	**0.0591**	**0.0245**

Table 3: Recommendation accuracy of disentangled category-independent representation on category-specifc testing data.

version of AUC, measuring the goodness of intra-user recommendation by averaging AUC over users. Besides, we followed previous work [39, 42] to use the RelaImpr metric to measure the relative improvement over the base NFM model on UAUC. For a random guesser, the value of AUC is 0.5, and thus RelaImpr is defined as:

$$\text{RelaImpr} = \left(\frac{\text{UAUC(mesured model)} - 0.5}{\text{UAUC(base NFM model)} - 0.5} - 1 \right) \times 100\%.$$

To evaluate the second perspective of recommendation accuracy, we adopted Recall@K and NDCG@K for the purpose. Regarding recommendation diversity, we used two widely-adopted metrics: (1) Category coverage (CC@K), which is the ratio between number of categories covered by top-K recommendations and the total number of categories in dataset; (2) Category entropy (CE@K), which is the entropy of category distribution in top-K recommendations. Higher CC@K and CE@K suggest more diverse top-K recommendations.

Table 2 shows the experiment results of all algorithms. We cannot report AUC, UAUC and RelaImpr for PD_GAN, since it directly recommends a set of items. For MMR and DPP, we can only report UAUC and RelaImpr since it is hard to find an appropriate way to merge the re-ranked list of different users to calculate AUC. Based on the results, we can observe that:

- Although Unawareness, MMR, DPP, PD_GAN and DGCN promoted more diverse recommendations with higher CE@K and CC@K, their recommendation accuracy dropped a lot, indicating their failure to handle accuracy-diversity dilemma.
- IPS did not consistently outperform the base NFM model in recommendation diversity or accuracy, due to the inaccurate estimation and high variance of propensity scores.
- At most time, especially on ML_1M and ML_10M100K dataset, DecRS improved both recommendation accuracy and diversity, since it could avoid many less-relevant or low-quality items from the dominant categories being recommended. However, its improvement was not larger than our proposed DCRS.

- Our proposed DCRS effectively improved both recommendation accuracy and diversity on all three datasets compared to the base NFM model. One can observe on all datasets, DCRS achieved the highest recommendation accuracy in all metrics, and generated more diverse recommendations than the base NFM model with higher CC@K and CE@K. This implies that DCRS tends to generate diverse recommendations the users will prefer, rather than solely pursuing diversity regardless of recommendation accuracy. Moreover, compared to DCRS, the recommendation accuracy of DCRS_CI dropped on all three datasets, confirming the importance of modeling users' categorical preference.

3.3 Predicting Users' In-Category Preferences

We dive deeper to investigate why DCRS can make accurate and diversified recommendations. Based on our design, the disentanglement shields the users' preference on item categories from their preference on items within the category when learning the user-item representations. As a result, the user-item representations learnt by DCRS should better predict a user's interest within item category, compared to those did not consider this aspect. Thus we inspect whether the disentangled category-independent representation (i.e., $\{\boldsymbol{h}_{u,i}^{\perp C}\}$) can distinguish less relevant (or low-quality) items from relevant (or high-quality) items more accurately within a given category of items.

We split the testing dataset according to item categories, and evaluated all algorithms on each category-specific testing dataset separately. On all three of our evaluation datasets, an item may relate to multiple categories. For example, the movie "Toy Story (1995)" in ML-1M dataset is related to three categories: "Animation", "Children's", and "Comedy". Here, we split the testing dataset according to each unique combination of related categories. Then given one unique combination of categories, we traversed the testing dataset and only kept user-item interactions where the interacted item is associated with the same category combination. To ensure the reliability of the evaluation results, on each dataset, we only evaluated the algorithms on the top-3 most popular categories.

In this experimental setting, we only need to evaluate DCRS_CI, as all items are from the same category. Table 3 demonstrates the experiment results. Due to space limit, we only report results on AUC, UAUC, Recall@20 and NDCG@20, and omit baselines that perform worse than the base NFM model. From Table 3, we can observe that both DecRS and DCRS_CI performed better than the base NFM model, as aligned with the results in Section 3.2. Moreover, DCRS_CI achieved the best performance most time, implying that disentangled representations contribute to more accurate preference modeling within categories.

3.4 Case Study

We also use a case study to qualitatively illustrate the behavior of the proposed DCRS model. Figure 4 shows the distribution of categories in the interacted items in training and testing data of a user from ML-1M dataset, as well as the top-10 recommended items generated by NFM, DecRS and DCRS. One can observe from Figure 4 that: the top-10 recommendations of NFM and DecRS model mainly fell in the "Thriller" category, which is the most popular in the training data of this user. Our proposed DCRS could capture the user's preference over categories more thoroughly. As shown in Figure 4,

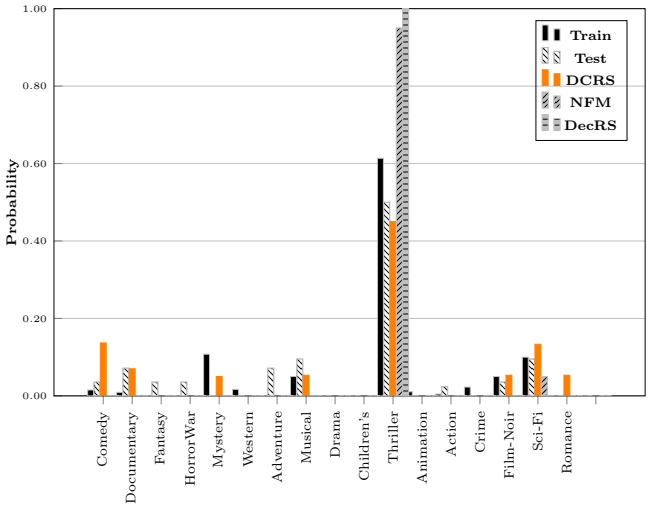

Figure 4: Categorical distributions of training data, testing data and top-10 recommended items of a sampled user.

the recommended items from DCRS did not simply concentrate to the dominant category "Thriller" as in other recommendation algorithms, but they successfully covered six out of ten categories that have a non-zero support in the user's testing data. Moreover, DCRS could also identify the user's preference on categories that the user seldom interacted with before, for example, the category of "Documentary". This explains its improved recommendation diversity without losing recommendation accuracy.

4 ONLINE DEPLOYMENT AND A/B TEST

To further verify the effectiveness of DCRS, we deploy it on the recommendation channel of Toutiao app, one of the largest news recommendation platforms in China, for online A/B test.

More specifically, we implemented DCRS based on the main candidate generator of Toutiao. Here, the main candidate generator is one of many candidate generators that produce recommendation candidates, which are later scored and ranked by a separate ranking model before presenting to users [6]. But the recommendation candidates produced by the main candidate generator account for the largest proportion of the recommendations shown to users. We then replaced the main candidate generator by DCRS in the experimental group, and used the prior main candidate generator in the control group. We adopted two key metrics: (1) *Click Through Rate (CTR)*; (2) *StayTime*, to measure users' satisfaction with the resulting recommendations. To accurately evaluate recommendation diversity, we only targeted items with more than 1000 impressions, because for those that appear less frequently could be introduced by some special strategies rather than the compared methods. We then calculated four metrics: (1) E_CN: number of distinct categories of targeted items shown to a user; (2) E_CE: entropy of category distribution of targeted items shown to a user ; (3) R_CN: number of distinct categories of targeted items read by a user ; (4) R_CE: entropy of category distribution of targeted items read by a user. The A/B test was conducted for seventeen consecutive days and the average performance of the above metrics is reported. We report experimental results in Table 4. All reported results are significant with p-value < 0.05. We can observe that DCRS achieved higher *CTR* and *StayTime*, indicating improved users' satisfaction. Moreover, while the improvements in E_CN and E_CE were not that

	Δ*CTR*	Δ*StayTime*	ΔE_CN	ΔE_CE	ΔR_CN	ΔR_CE
DCRS	+0.973%	+0.062%	+0.197%	+0.111%	+2.372%	+2.276%

Table 4: Results of online A/B test on Toutiao app.

large, DCRS gained huge improvements in R_CN and R_CE, implying DCRS is able to generate *diverse recommendations the user will prefer.*

5 RELATE WORK

DCRS is closely related to two lines of existing work: (1) addressing accuracy-diversity dilemma in recommendations; and (2) disentangled user representation learning for general user modeling.

Addressing accuracy-diversity dilemma in recommendations. Besides recommendation accuracy, more and more research suggests other factors of recommendation quality also contribute to the overall user satisfaction about the system. Of these factors, recommendation diversity has been shown as a critically important one [1, 32], which however also leads to the so-called accuracy-diversity dilemma [31, 40]: higher accuracy often means losing diversity to some extent and vice verse. One main reason is that previous solutions with accuracy as the primary goal tend to focus on items in the dominant categories in users' interaction history. In order to guarantee user satisfaction, three different types of solutions are proposed, namely post-processing, learning to rank, and diversified recommendation models.

For the first, and most popular, type of solutions, a re-ranking or post-processing module is appended to a chosen recommendation model. The post-processing module takes recommended items as input and re-orders them to balance recommendation accuracy and diversity. Various post-processing algorithms [2, 5, 19, 27, 44] are proposed. For example, Ziegler et. al. [44] first applied the Maximal Marginal Relevance (MMR) algorithm, which was used for topic diversification in search engines, to minimize redundancy among recommended items. Determinantal Point Process (DPP) has been shown as the most effective one [5] of all post-processing algorithms, which scores an entire list of items rather than every item individually for better modeling of item correlations. However, all these post-processing algorithms are separately constructed from the recommendation models, though their learning highly depends on the performance of the recommendation model. When the recommendation model fails to provide a diverse item list to start with, or gives pretty-low scores to diverse items, the effectiveness of the aforementioned post-processing algorithms will largely deteriorate. Moreover, as shown in our experiment results in Section 3, the aforementioned post-processing algorithms usually seriously sacrifice recommendation accuracy.

Learning To Rank type solutions [8, 21, 34] aim to directly recommend a list of items to users, rather than selecting items one by one according to their prediction scores. However, this line of work often suffers from high time complexity, which limits its application in real world recommendation scenarios.

Recently, several solutions are proposed to directly improve the diversity of recommendation models. Zheng et. al [40] proposed a diversified recommendation model based on Graph Convolutional Networks (GCN), with improving recommendation diversity as its only target. Wu et. al [36] leveraged the GAN framework for diverse recommendations, where a generator tries to recommend diverse sets of items, and a discriminator aims to distinguish the generated recommendations from a set of items randomly sampled from the observed data of the target user. The most related work to ours is [31], where the authors studied the problem of lack of diversity in recommendations from a casual perspective, and proposed DecRS to alleviate the problem. Experiments demonstrate the advantage of our proposed DCRS over these solutions in improving recommendation accuracy and diversity. A recent work [20] also tried to diversify recommendations in relevant recommendation scenario, where the diversification is conducted regarding multiple item aspects such that relevance and diversity are adaptively balanced among different item aspects. However, when only one item aspect is considered, e.g., the item category in this paper, their algorithm degenerates to the MMR algorithm.

Disentangled user representations. Learning disentangled user representations has drawn increasing attention in recent years. A family of solutions are based on Variational Auto-Encoder (VAE) to force each dimension of learnt representations to focus on different latent factors [22, 24, 38]. However, such a disentanglement is implicit and therefore one cannot associate the disentangled representation with the specific attributes of interest. Zheng et. al [41] proposed DICE to learn representations where user interest and conformity are structurally disentangled via direct supervision from cause-specific data. However, in our problem, we cannot access users' preferences over item categories explicitly, thus are not able to get any direct supervision about it. Chen et. al. [7] proposed to disentangle item representations to address popularity bias, by requiring the two disentangled item representations to be orthogonal. In our solution, we disentangle a user's preference over an item into category dependent and independent segments. Both segments relate to the user and thus they do not need to be orthogonal to each other.

6 CONCLUSION

In this paper, we propose a new principle that the diversification of recommendations should be performed within a user's preferred categories, such that improved recommendation diversity can be achieved without sacrificing recommendation accuracy. We realize this principle via a general framework, named DCRS, to disentangle a user's category dependent and independent preference in the learnt representations. We evaluate DCRS through both offline experiments on three widely-used benchmark datasets for recommendation and online A/B test on Toutiao, one of the largest news recommendation platforms in China. We demonstrate DCRS can provide more accurate and diversified recommendations. Via in-depth analysis and case studies, we find that the benefit of DCRS is introduced by: (1) it can capture a user's diverse preference in historical interactions more thoroughly; and (2) it can rank items in the same category more accurately.

In this work, we took a static view of users' preferences over items and item categories. But numerous studies have demonstrated that users' preferences evolve over time [35, 37]. It is interesting to study the problem of recommendation diversification in an interactive manner over time. Moreover, currently we only recommend one item a time to a user. It is interesting to study how to generate a list of diverse recommendations, where diversity should be optimized within and across recommendation lists.

REFERENCES

[1] Ashton Anderson, Lucas Maystre, Ian Anderson, Rishabh Mehrotra, and Mounia Lalmas. 2020. Algorithmic effects on the diversity of consumption on spotify. In *Proceedings of The Web Conference 2020*. 2155–2165.

[2] Azin Ashkan, Branislav Kveton, Shlomo Berkovsky, and Zheng Wen. 2015. Optimal greedy diversity for recommendation. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

[3] Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. 335–336.

[4] Allison JB Chaney, Brandon M Stewart, and Barbara E Engelhardt. 2018. How algorithmic confounding in recommendation systems increases homogeneity and decreases utility. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 224–232.

[5] Laming Chen, Guoxin Zhang, and Hanning Zhou. 2018. Fast greedy map inference for determinantal point process to improve recommendation diversity. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. 5627–5638.

[6] Minmin Chen, Alex Beutel, Paul Covington, Sagar Jain, Francois Belletti, and Ed H Chi. 2019. Top-k off-policy correction for a REINFORCE recommender system. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 456–464.

[7] Zhihong Chen, Jiawei Wu, Chenliang Li, Jingxu Chen, Rong Xiao, and Binqiang Zhao. 2022. Co-training Disentangled Domain Adaptation Network for Leveraging Popularity Bias in Recommenders. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 60–69.

[8] Peizhe Cheng, Shuaiqiang Wang, Jun Ma, Jiankai Sun, and Hui Xiong. 2017. Learning to recommend accurate and diverse items. In *Proceedings of the 26th international conference on World Wide Web*. 183–192.

[9] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.

[10] Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)* 39, 1 (1977), 1–22.

[11] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research* 12, 7 (2011).

[12] Tom Fawcett. 2006. An introduction to ROC analysis. *Pattern recognition letters* 27, 8 (2006), 861–874.

[13] Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*. PMLR, 1180–1189.

[14] Yingqiang Ge, Shuya Zhao, Honglu Zhou, Changhua Pei, Fei Sun, Wenwu Ou, and Yongfeng Zhang. 2020. Understanding echo chambers in e-commerce recommender systems. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*. 2261–2270.

[15] Nina Grgic-Hlaca, Muhammad Bilal Zafar, Krishna P Gummadi, and Adrian Weller. 2016. The case for process fairness in learning: Feature selection for fair decision making. In *NIPS symposium on machine learning and the law*, Vol. 1. 2.

[16] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).

[17] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. 355–364.

[18] Komal Kapoor, Vikas Kumar, Loren Terveen, Joseph A Konstan, and Paul Schrater. 2015. " I like to explore sometimes" Adapting to Dynamic User Novelty Preferences. In *Proceedings of the 9th ACM Conference on Recommender Systems*. 19–26.

[19] Mesut Kaya and Derek Bridge. 2019. A comparison of calibrated and intent-aware recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 151–159.

[20] Zihan Lin, Hui Wang, Jingshu Mao, Wayne Xin Zhao, Cheng Wang, Peng Jiang, and Ji-Rong Wen. 2022. Feature-aware Diversified Re-ranking with Disentangled Representations for Relevant Recommendation. *arXiv preprint arXiv:2206.05020* (2022).

[21] Yuli Liu, Christian Walder, and Lexing Xie. 2022. Determinantal Point Process Likelihoods for Sequential Recommendation. *arXiv preprint arXiv:2204.11562* (2022).

[22] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learning disentangled representations for recommendation. *arXiv preprint arXiv:1910.14238* (2019).

[23] Andriy Mnih and Russ R Salakhutdinov. 2007. Probabilistic matrix factorization. *Advances in neural information processing systems* 20 (2007).

[24] Preksha Nema, Alexandros Karatzoglou, and Filip Radlinski. 2021. Disentangling Preference Representations for Recommendation Critiquing with ß-VAE. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1356–1365.

[25] Tien T Nguyen, Pik-Mai Hui, F Maxwell Harper, Loren Terveen, and Joseph A Konstan. 2014. Exploring the filter bubble: the effect of using recommender systems on content diversity. In *Proceedings of the 23rd international conference on World wide web*. 677–686.

[26] Shi Pu, Yijiang He, Zheng Li, and Mao Zheng. 2020. Multimodal Topic Learning for Video Recommendation. *arXiv preprint arXiv:2010.13373* (2020).

[27] Lijing Qin and Xiaoyan Zhu. 2013. Promoting diversity in recommendation by entropy regularizer. In *Twenty-Third International Joint Conference on Artificial Intelligence*.

[28] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[29] Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. 2020. Unbiased recommender learning from missing-not-at-random implicit feedback. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 501–509.

[30] Harald Steck. 2018. Calibrated recommendations. In *Proceedings of the 12th ACM conference on recommender systems*. 154–162.

[31] Wenjie Wang, Fuli Feng, Xiangnan He, Xiang Wang, and Tat-Seng Chua. 2021. Deconfounded Recommendation for Alleviating Bias Amplification. *arXiv preprint arXiv:2105.10648* (2021).

[32] Mark Wilhelm, Ajith Ramanathan, Alexander Bonomo, Sagar Jain, Ed H Chi, and Jennifer Gillenwater. 2018. Practical diversified recommendations on youtube with determinantal point processes. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 2165–2173.

[33] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. Npa: Neural news recommendation with personalized attention. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2576–2584.

[34] Liwei Wu, Cho-Jui Hsieh, and James Sharpnack. 2018. Sql-rank: A listwise approach to collaborative ranking. In *International Conference on Machine Learning*. PMLR, 5315–5324.

[35] Qingyun Wu, Naveen Iyer, and Hongning Wang. 2018. Learning contextual bandits in a non-stationary environment. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 495–504.

[36] Qiong Wu, Yong Liu, Chunyan Miao, Binqiang Zhao, Yin Zhao, and Lu Guan. 2019. PD-GAN: Adversarial Learning for Personalized Diversity-Promoting Recommendation.. In *IJCAI*, Vol. 19. 3870–3876.

[37] Qingyun Wu, Huazheng Wang, Yanen Li, and Hongning Wang. 2019. Dynamic ensemble of contextual bandits to satisfy users' changing interests. In *The World Wide Web Conference*. 2080–2090.

[38] Zhe Xie, Chengxuan Liu, Yichi Zhang, Hongtao Lu, Dong Wang, and Yue Ding. 2021. Adversarial and Contrastive Variational Autoencoder for Sequential Recommendation. In *Proceedings of the Web Conference 2021*. 449–459.

[39] Ling Yan, Wu-jun Li, Gui-Rong Xue, and Dingyi Han. 2014. Coupled group lasso for web-scale ctr prediction in display advertising. In *International Conference on Machine Learning*. PMLR, 802–810.

[40] Yu Zheng, Chen Gao, Liang Chen, Depeng Jin, and Yong Li. 2021. DGCN: Diversified Recommendation with Graph Convolutional Networks. In *Proceedings of the Web Conference 2021*. 401–412.

[41] Yu Zheng, Chen Gao, Xiang Li, Xiangnan He, Yong Li, and Depeng Jin. 2021. Disentangling User Interest and Conformity for Recommendation with Causal Embedding. In *Proceedings of the Web Conference 2021*. 2980–2991.

[42] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1059–1068.

[43] Tao Zhou, Zoltán Kuscsik, Jian-Guo Liu, Matúš Medo, Joseph Rushton Wakeling, and Yi-Cheng Zhang. 2010. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences* 107, 10 (2010), 4511–4515.

[44] Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*. 22–32.