Content-Collaborative Disentanglement Representation Learning for Enhanced Recommendation

Yin Zhang, Ziwei Zhu, Yun He, James Caverlee
zhan13679@tamu.edu,zhuziwei@tamu.edu,yunhe@tamu.edu,caverlee@cse.tamu.edu
Texas A&M University
College Station, Texas

ABSTRACT

Modern recommenders usually consider both collaborative features from user behavior data (e.g., clicks) and content information about the users and items (e.g., user ages or item images) for improved recommendations. While encouraging, the uncovered user preference representations derived from these collaborative and content-based perspectives can be entangled by intermixing the influence from each other, leading to sub-optimal performance and unstable recommendations. Hence, we propose to disentangle representations learned from user behavior data and content information. Specifically, we propose a novel two-level disentanglement generative recommendation model (DICER) that supports both content-collaborative disentanglement and feature disentanglement: for the content-collaborative disentanglement, DICER decomposes the features by their marginal distributions based on content and user-item interactions, to ensure the learned features from each type are statistically independent. For feature disentanglement, by decomposing the Kullback-Leibler divergence, we theoretically show that extracted features within each type are disentangled at a granular level. Furthermore, DICER utilizes a co-decoder that simultaneously decodes the content and user-item interactions to ensure the high-quality of learned features. Through extensive experiments on three real-world datasets, results show that DICER significantly outperforms other state-of-the-art methods by 13.5% in NDCG and 14.4% in hit ratio on average.

CCS CONCEPTS

• **Information systems** → *Recommender systems.*

KEYWORDS

disentangled representation learning, content-aware recommendation, collaborative filtering

ACM Reference Format:
Yin Zhang, Ziwei Zhu, Yun He, James Caverlee. 2020. Content-Collaborative Disentanglement Representation Learning for Enhanced Recommendation. In *Fourteenth ACM Conference on Recommender Systems (RecSys '20), September 22–26, 2020, Virtual Event, Brazil.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3383313.3412239

RecSys '20, September 22–26, 2020, Virtual Event, Brazil

ACM ISBN 978-1-4503-7583-2/20/09…$15.00
https://doi.org/10.1145/3383313.3412239

1 INTRODUCTION

One of the fundamental challenges for recommender systems is to learn user and item representations that can uncover user preference towards items. Many recent efforts adopt deep approaches [31, 32] over both *collaborative features* based on user-item interactions (e.g., clicks or likes) and *content information* about the users and items (e.g., user ages or item images) [7, 20, 26, 27, 36] to build user and item representations, as shown in Figure 1(a). While encouraging, the learned user and item features derived from these collaborative and content-based perspectives can be *entangled* by intermixing the influence from each, harming recommendation quality.

For example, a user, and also many similar users, may prefer a dress because of its visual appearance, price, and high quality. The known content information is the dress images. If the user-item interactions and the dress image are considered separately to learn the features that influence user preference towards items, the collaborative features and content features could be highly correlated. In essence, both the collaborative and content features could redundantly encode the visual characteristics of the dress, meaning there is less capacity to focus on learning other features (like price) that could influence user preference towards items. Hence, learning user preferences based on both content-based features and collaborative features could lead to *feature duplication and high feature correlation*, limiting the representation capability for modeling collections of users with diverse interests. Furthermore, the feature duplication and high correlation among features can also result in overweighting these correlated features in learning user preference, leading to sub-optimal performance and unstable recommendations.

Therefore, we propose to *disentangle representations learned from user behavior data and content information* to improve the integrated user and item representation quality for improved recommendation. Such disentanglement representation learning – which aims to learn disentangled features such that any one feature is relatively not influenced by changes in other features – has recently demonstrated powerful and robust performance in many areas, especially in computer vision [6, 12, 18, 19]. However, there is little work on disentangled representation learning in recommendation [22], and none on recommendation when both user behavior data and content information are available. Disentanglement representation learning poses unique challenges in this context:

- First, most existing disentanglement problems target areas where the features are explicitly known (e.g., shape or color features of an image). However, user-item interactions do not necessarily map to specific apriori collaborative features. Hence, this heterogeneity between implicit features in user-item interactions and explicit features in content information

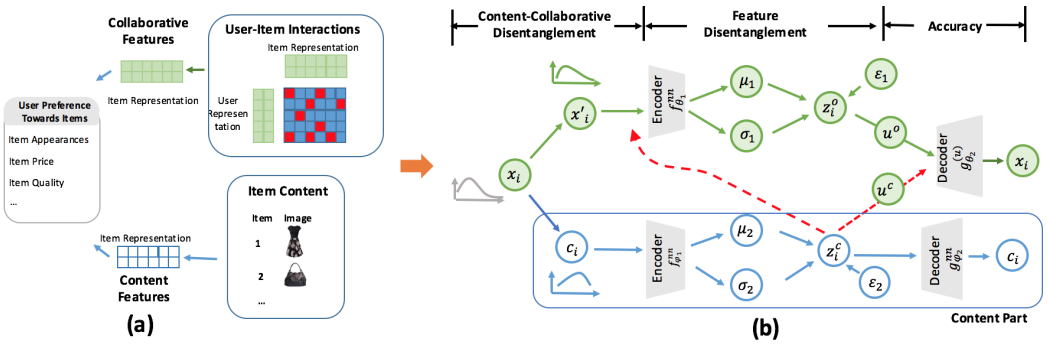

Figure 1: (a) Both collaborative features learned from user-item interactions and content features learned from content information (like images) can be used to discover user preference. (b) DICER first disentangles the user implicit feedback x_i to content information c_i and content disentangled collaborative information x'_i. Then within each type (either collaborative or content), we further disentangle learned features at a granular level (feature disentanglement) to improve the capacity of user and item representations.

makes it extremely difficult to capture the disentanglement *across* content and collaborative features.

- Second, disentanglement across content and collaborative features only ensures that the learned representations from each type are different. However, the specific features *within* the collaborative features could still be highly entangled with each other, and similarly for the content-based features. Hence, we also face the challenge of granular-level disentanglement, to simultaneously learn both disentangled and high-quality features within collaborative and content features for improved recommendation.

With these challenges in mind, we propose a novel two-level disentanglement approach called DICER – DIsentangling Content-aware collaborative filtering for Enhanced Recommendation – that supports both content-collaborative disentanglement and feature disentanglement based on the structure of a variational auto-encoder. For content-collaborative disentanglement, DICER decomposes the features that influence user preference into content features and content disentangled collaborative features, then utilizes their marginal distributions to learn disentangled features for each type. For feature disentanglement, we theoretically show that each extracted feature in DICER is disentangled with other extracted features by decomposing the Kullback-Leibler divergence via statistical independence properties. Furthermore, DICER is characterized by a co-decoder that simultaneously decodes the content and user-item interactions to ensure the high-quality of both learned content and collaborative features. Through extensive experiments on three real-world datasets, results show that DICER significantly outperforms other state-of-the-art methods by 13.5% in NDCG and 14.4% in hit ratio on average. We also find that DICER captures relatively independent features through disentanglement measurement and visualization.

2 RELATED WORK

Latent Factor-based Recommendation Models with Content Information. Latent factor models (e.g., matrix factorization and recent neural approaches) [5, 11, 14, 16, 25, 33] typically map both users and items to a latent factor space with a low latent dimension. However, since only user-item interactions are considered, these models usually suffer from sparsity and cold-start problems. Thus, latent factor models for recommendation have been augmented to incorporate additional content information of items and users [7, 20, 26, 27, 36]. For instance, Li et.al. [15] combined item content information and user-item feedback into a variational autoencoder to learn user preference towards items. Lv et al. [21] proposed a multimodal item similarity-based framework that learned visual and textual features for recommendations. However, most of these existing content-based latent factor models extract the content and collaborative features independently and then simply concatenate them to learn representations for users and items, without fully considering the correlation between user-item feedback and content information. For example, VBPR [7] directly concatenates the visual features learned from item images with collaborative features as the joint item representation and multiplies it with user representations to predict the user preference. This learned representation may contain duplicated/highly correlated features between images and collaborative features, leading to less robust models of user preference.

Table 1: Notation.

Notation	Explanation
$\mathbf{z}_i^o$	representation of learned content disentangled collaborative features, $\mathbf{z}_i^o \in R^{K_1}$
$\mathbf{x}_i$	item i interactions with users
$\theta = \{\theta_1, \theta_2\}$	trainable variables for user-item feedback
	θ_1/θ_2 is the encoder/decode trainable variables
$\mathbf{z}_i^c$	representation of learned content features, $\mathbf{z}_i^c \in R^{K_2}$
$\mathbf{c}_i$	item i content vector
$\phi = \{\phi_1, \phi_2\}$	trainable variables for item content
	ϕ_1/ϕ_2 is the encoder/decode trainable variables

Disentangled Representation Learning. Recently, disentanglement representation learning has attracted increasing attention due

to its robust performance and interpretability [6, 12, 18, 19]. Disentanglement learning aims to identify each feature that is relatively not influenced by other feature changes. For example, disentanglement learning on a visual dataset might learn the shape, the color, and the position features of the object [2], where each feature is not easily influenced by other feature changes. One popular method to capture disentangled features is based on *statistical independence* [4], which has demonstrated good performance in many applications [13, 29]. For example, β-VAE shows that disentanglement can be achieved if the KL term in the evidence lower bound (ELBO) is highly penalized. Based on that, Chen *et al.* [4] introduced β-TCVAE which provides a further decomposition of the ELBO to explain the penalty of KL for feature disentanglement. For recommendation, Ma *et al.* [22] learned disentangled latent representations for users and items based on user-item feedback and showed great improvement in recommendation. However, few if any methods have considered the disentanglement problem in the context of recommendation with both user behavior data and content information, which is especially important for building robust and high-quality user-item joint representations for content-aware recommendation in practice. In this work, we aim to address this gap.

3 METHOD

Problem Statement. Suppose we have a set of users $u \in \{1, 2, ..., U\}$ and items $i \in \{1, 2, , ...I\}$. To learn user preference, we have two types of information: (1) User-item interactions capturing the implicit feedback $x_{u,i}$ from user u towards item i, e.g., based on clicks, likes, or purchases. $x_{u,i} = 1$ indicates positive feedback, whereas $x_{u,i} = 0$ means the corresponding feedback is missing; and (2) Item content information $\mathbf{c}_i$ for each item i. This content could correspond to item images, reviews, descriptive text, or other item-specific information. Our task is to consider both user-item interactions and item content information to learn their integrated disentangled representations of users and items for improved recommendation.[1]

Approach. As we have argued, evidence from both user-item interactions and content information can lead to entangled representations. Hence, we propose a two-level approach called DICER that first disentangles features *between* content and collaborative features – called content-collaborative disentanglement, and then disentangles each feature *within* the collaborative features (and each feature within the content features) – called feature disentanglement. Specifically, as shown in Figure 1(b), the proposed DICER approach contains two key variables: *content features* $\mathbf{z}_i^c$ extracted from item content $\mathbf{c}_i$, and *content disentangled collaborative features* $\mathbf{z}_i^o$ extracted from user-item interactions $\mathbf{x}_i$.

3.1 Content-Collaborative Disentanglement

We begin by disentangling the information that is learned from content and user-item interactions, to ensure $\mathbf{z}_i^c$ and $\mathbf{z}_i^o$ capture different aspects of user preference. For example, suppose we have item images as the content information, and $x_{ui} = 1$ (that user u likes item i) due to the item's visual appearance and price. The content-collaborative disentanglement aims to learn the visual aspect from item images, leaving price-oriented features (that may be hard to precisely learn from images) to user-item interactions. In this way, $\mathbf{z}_i^o$ can *discover* other useful features that are not captured by $\mathbf{z}_i^c$.

To model this content-collaborative disentanglement, we propose to start by modeling the *joint distribution* of all the item influence features $\mathbf{z}_i \in R^K$ (the features that influence the user preference towards items), in order to connect user-item interactions and item content information. Then we decompose the joint distribution to extract disentangled features from content and user-item interactions.

Concretely, we first model the user feedback towards items (e.g., click history) $\mathbf{x}_i$ that is generated from all the features $\mathbf{z}_i$ with the following distribution:

$$p_\theta(\mathbf{x}_i) = \mathbb{E}_{p(\mathbf{z}_i)} p_\theta(\mathbf{x}_i|\mathbf{z}_i), \tag{1}$$

where θ is the set of model parameters for modeling user-item interactions.

Then, to utilize both the item content and user-item interaction information, and also disentangle features derived from these two types, we propose to decompose $\mathbf{z}_i$ to be the content features $\mathbf{z}_i^c$ derived from item content $\mathbf{c}_i$ and the content disentangled collaborative features $\mathbf{z}_i^o$ derived from user-item interactions $\mathbf{x}_i$. Therefore, the distribution of $\mathbf{z}_i$ can be expressed as the joint distribution of $\mathbf{z}_i^c$ and $\mathbf{z}_i^o$, i.e. $p(\mathbf{z}_i) = p(\mathbf{z}_i^c, \mathbf{z}_i^o)$. More importantly, to capture the disentangled features from the two types, similar as many disentanglement approaches [4, 22], we set the extracted features from the content and user-item interactions to be statistically independent:

$$p(\mathbf{z}_i) = p(\mathbf{z}_i^c, \mathbf{z}_i^o) = p(\mathbf{z}_i^c)p(\mathbf{z}_i^o). \tag{2}$$

That is, based on the disentanglement between $\mathbf{z}_i^c$ and $\mathbf{z}_i^o$, we can further decompose the joint distribution of $\mathbf{z}_i^c$ and $\mathbf{z}_i^o$ to be their marginal distributions. Then, with the disentanglement, our model Equation (1) can be rewritten as:

$$\begin{aligned} p_\theta(\mathbf{x}_i) = \mathbb{E}_{p(\mathbf{z}_i)} p_\theta(\mathbf{x}_i|\mathbf{z}_i) &= \int p_\theta(\mathbf{x}_i|\mathbf{z}_i^c, \mathbf{z}_i^o) p(\mathbf{z}_i^c, \mathbf{z}_i^o) d(\mathbf{z}_i^c, \mathbf{z}_i^o) \\ &= \int p(\mathbf{z}_i^c) \int p_\theta(\mathbf{x}_i|\mathbf{z}_i^c, \mathbf{z}_i^o) p(\mathbf{z}_i^o) d\mathbf{z}_i^o d\mathbf{z}_i^c \\ &= \mathbb{E}_{p(\mathbf{z}_i^c)} \mathbb{E}_{p(\mathbf{z}_i^o)} p_\theta(\mathbf{x}_i|\mathbf{z}_i^c, \mathbf{z}_i^o), \end{aligned} \tag{3}$$

where $p(\mathbf{z}_i^o) = p(\mathbf{z}_i^o|\mathbf{z}_i^c)$ is based on their statistical independence. With Equation (3), to predict user preference with the known $x_{u,i}$ and $\mathbf{c}_i$, we discuss the two main components: $p(\mathbf{z}_i^c)$ and $p_\theta(\mathbf{x}_i|\mathbf{z}_i^c, \mathbf{z}_i^o)$ in Equation (3), respectively.

For content feature distribution $p(\mathbf{z}_i^c)$, it is modeled based on the item content information $\mathbf{c}_i$:

$$p_\phi(\mathbf{c}_i) = \mathbb{E}_{p(\mathbf{z}_i^c)} p_\phi(\mathbf{c}_i|\mathbf{z}_i^c), \tag{4}$$

where ϕ is the model parameters for modeling the content information. Thus, we can use this content information to get the $\mathbf{z}_i^c$ distribution, then capture the $\mathbf{z}_i^o$ distribution based on both $\mathbf{x}_i$ and $\mathbf{z}_i^c$.

For $p_\theta(\mathbf{x}_i|\mathbf{z}_i^c, \mathbf{z}_i^o)$, we model it as:

$$p_\theta(\mathbf{x}_i|\mathbf{z}_i^c, \mathbf{z}_i^o) = \prod_{x_{i,u} \in \mathbf{x}_i} p_\theta(x_{i,u}|\mathbf{z}_i^c, \mathbf{z}_i^o).$$

[1] We could also consider user content information alone (instead of item content information), or both user and item information together in addition to user-item interactions. We discuss both of these scenarios in Section 3.4.

which is similar to the VAE-based method [13, 17]. Then the probability of user u's preference to item i is $p_\theta(x_{i,u}|\mathbf{z}_i^c, \mathbf{z}_i^o)$, which can be derived by a user-aware non-linear transformation, such as a feed-forward neural network. We discuss the details in Section 3.3. Thus, we can first learn $\mathbf{z}_i^c$ and $\mathbf{z}_i^o$, and use the decoder to capture the $p_\theta(\mathbf{x}_i|\mathbf{z}_i^c, \mathbf{z}_i^o)$.

Variational Inference. To estimate the model parameters θ in Equation (3), we follow the VAE-based paradigm. Note here, different from traditional VAE based methods [17] that can directly estimate the posterior distribution of $p(\mathbf{z}_i|\mathbf{x}_i)$ for each data point, or the conditional VAE [28] that knows part of the data information (i.e., labels of the data), our case is more complex. That is, $\mathbf{x}_i$ is related to the joint distribution of latent factors $\mathbf{z}_i^o$ and $\mathbf{z}_i^c$, where $\mathbf{z}_i^c$ is extracted from item content information. Thus, we need to calculate the evidence lower bound (ELBO) based on both $\mathbf{z}_i^o$ and $\mathbf{z}_i^c$, their independent relations, and their relations to $\mathbf{x}_i$ and $\mathbf{c}_i$.

Concretely, $\mathbf{x}_i$ is generated based on the joint distribution of $\mathbf{z}_i^c$ and $\mathbf{z}_i^o$, where $\mathbf{z}_i^c$ can be estimated by item content (such as images or text-based descriptions). The ELBO of $lnp(\mathbf{x}_i)$ can be written as:

$$\begin{aligned} lnp_\theta(\mathbf{x}_i) \geq \mathbb{E}_{p(\mathbf{z}_i^c)}[\mathbb{E}_{q_\theta(\mathbf{z}_i^o|\mathbf{x}_i,\mathbf{z}_i^c)}(lnp_\theta(\mathbf{x_i}|\mathbf{z}_i^c, \mathbf{z}_i^o)) \\ - D_{KL}(q_\theta(\mathbf{z}_i^o|\mathbf{x}_i, \mathbf{z}_i^c)||p(\mathbf{z}_i^o))] \equiv L(\mathbf{x}_i; \theta). \end{aligned} \tag{5}$$

Since $\mathbb{E}_{q(\mathbf{z}_i^c)}[\cdot]$ and $\mathbb{E}_{q(\mathbf{z}_i^o|\mathbf{x}_i,\mathbf{z}_i^c)}[\cdot]$ are intractable, we utilize the variational inference and reparametrization trick [13]. Details are discussed in Section 3.3.

To estimate ϕ in Equation (4), for consistency, we also use the VAE-based paradigm. The corresponding ELBO is:

$$\begin{aligned} lnp_\phi(\mathbf{c}_i) \geq \mathbb{E}_{q_\phi(\mathbf{z}_i^c|\mathbf{c}_i)}(lnp_\phi(\mathbf{c}_i|\mathbf{z}_i^c)) \\ - D_{KL}(q_\phi(\mathbf{z}_i^c|\mathbf{c}_i)||p(\mathbf{z}_i^c)) \equiv L(\mathbf{c}_i; \phi). \end{aligned} \tag{6}$$

Through Equation (6), we can get an estimation of the item content features $\mathbf{z}_i^c$. Then we transform it into the user-item space to help learn user preference towards this item through Equation (5).

3.2 Feature Disentanglement

The content-collaborative disentanglement ensures $\mathbf{z}_i^c$ and $\mathbf{z}_i^o$ learn different information from content and user-item interactions. However, the extracted features in $\mathbf{z}_i^c$ (and in $\mathbf{z}_i^o$) at a granular level could also be entangled and confound with each other, making the recommendation unstable and difficult to generalize. Thus, to ensure the quality of learned representations, we also aim to disentangle each extracted feature in $\mathbf{z}_i^c$ and $\mathbf{z}_i^o$ at a granular level. For example, we might want to learn features like color and shape from an item image, where changes to each feature (e.g., the color) do not strongly depend on other feature changes (e.g., the shape). To do so, considering each latent dimension represents a single item feature, we disentangle each dimension of the item representation to extract independent features. That is, we make a feature disentanglement that forces $z_{i,k}^c, \forall k/\ z_{i,k}^o, \forall k$ to be statistically independent. $z_{i,k}^c$ is the k_{th} value in the vector $\mathbf{z}_i^c$. Similar reasoning holds for $z_{i,k}^o$. The user feature disentanglement is jointly modeled with item disentanglement through a decoder of user-item interactions (for details see Section 3.3).

More importantly, in DICER, the content-collaborative disentanglement ensures the correlation $Corr_i(\mathbf{z}_i^c, \mathbf{z}_i^o) = \mathbf{0}$ (based on the Equation (2)). *Thus, the feature disentanglement inside $\mathbf{z}_i^c$ and $\mathbf{z}_i^o$ can further ensure all features in $\mathbf{z}_i$ that are learned in DICER are independent with each other: $Corr_i(z_{i,k}, z_{i,j}) = 0,\ \forall k \neq j$.*

Concretely, to enforce feature independence inside $\mathbf{z}_i^c$ and $\mathbf{z}_i^o$, we have:

$$q_\theta(\mathbf{z}_i^o|\mathbf{z}_i^c) \approx \prod_{k=1}^{K_1} q_\theta(z_{i,k}^o|\mathbf{z}_i^c), \quad q_\phi(\mathbf{z}_i^c) \approx \prod_{k=1}^{K_2} q_\phi(z_{c,k}^c). \tag{7}$$

Let's look at each equation separately.

Feature Disentanglement in $\mathbf{z}_i^o$. For the first equation in Equation (7), the aggregated posterior distribution of $q_\theta(\mathbf{z}_i^o|\mathbf{z}_i^c) = \int_{x_i} q_\theta(\mathbf{z}_i^o|\mathbf{x}_i, \mathbf{z}_i^c)p_{fdata}(\mathbf{x}_i)d\mathbf{x}_i$, where $p_{fdata}(\mathbf{x}_i)$ is the user feedback distribution. The posterior distribution captures the aggregated structure of the latent variables based on the user feedback distribution [23]. Therefore, the KL term in Equation (5) can be decomposed as:

$$\begin{aligned} &\mathbb{E}_{p_{fdata}(\mathbf{x}_i)}[D_{KL}(q_\theta(\mathbf{z}_i^o|\mathbf{x}_i, \mathbf{z}_i^c)||p(\mathbf{z}_i^o))] \\ &= I_q(\mathbf{x_i}; \mathbf{z_i^o}) + D_{KL}(q_\theta(\mathbf{z}_i^o|\mathbf{z}_i^c)||\Pi_k q_\theta(\mathbf{z}_{i,k}^o|\mathbf{z}_i^c)) \\ &+ \sum_k D_{KL}(q_\theta(\mathbf{z}_{i,k}^o|\mathbf{z}_i^c)||p(\mathbf{z}_{i,k}^o)), \end{aligned} \tag{8}$$

where $I_q(\mathbf{x_i}; \mathbf{z_i^o})$ stands for the mutual information (MI) [3]. A similar decomposition can be found in [3, 4]. For each term in Equation (8): (1) the index-code MI $I_q(\mathbf{x_i}; \mathbf{z_i^o}) = D_{KL}(q_\theta(\mathbf{z}_i^o, \mathbf{x}_i|\mathbf{z}_i^c)||q_\theta(\mathbf{z}_i^o|\mathbf{z}_i^c))$ is the mutual information between $\mathbf{x}_i$ and $\mathbf{z}_i^o$ based on the empirical user-item feedback distribution $q_\theta(\mathbf{z}_i^o|\mathbf{x}_i, \mathbf{z}_i^c)p_{fdata}(\mathbf{x}_i)$. As indicated by many recent studies [1, 4, 12], penalizing mutual information through the information bottleneck can encourage feature disentanglement; (2) the second term $D_{KL}(q_\theta(\mathbf{z}_i^o|\mathbf{z}_i^c)||\Pi_k q_\theta(\mathbf{z}_{i,k}^o|\mathbf{z}_i^c))$ is the total correlation. This penalty can encourage statistical independence of the learned latent representation in each dimension of $\mathbf{z}_i^o$ under the condition of $\mathbf{z}_i^c$; (3) the third term $\sum_k D_{KL}(q_\theta(\mathbf{z}_{i,k}^o|\mathbf{z}_i^c)||p(\mathbf{z}_{i,k}^o))$ ensures the learned latent representations in each dimension are close to their corresponding priors, which is known as dimension-wise KL [4].

Thus, based on Equation (8), if we use an independent prior distribution for each latent dimension of $\mathbf{z}_i^o$, i.e., $p_\theta(\mathbf{z_i^o}) = \Pi_k p_\theta(\mathbf{z}_{i,k}^o)$, the KL penalty term can encourage the disentanglement of the learned features in $\mathbf{z}_i^o$ from the mutual information and total correlation aspects. At the same time, it also ensures the close distribution between the posterior distribution and the priors by $\sum_k D_{KL}(q_\theta(\mathbf{z}_{i,k}^o|\mathbf{z}_i^c)||p(\mathbf{z}_{i,k}^o))$. Hence, the loss function of Equation (5) can be refined by adding a KL penalized parameter $\beta_1 > 1$:

$$\begin{aligned} L_{\beta_1}(\mathbf{x}_i; \theta) \equiv \mathbb{E}_{q_\theta(\mathbf{z}_i^c)}[\mathbb{E}_{q_\theta(\mathbf{z}_i^o|\mathbf{x}_i,\mathbf{z}_i^c)}(lnp_\theta(\mathbf{x_i}|\mathbf{z}_i^c, \mathbf{z}_i^o)) \\ - \beta_1 D_{KL}(q_\theta(\mathbf{z}_i^o|\mathbf{x}_i, \mathbf{z}_i^c)||p(\mathbf{z}_i^o))], \end{aligned} \tag{9}$$

to encourage each feature disentanglement of $\mathbf{z}_i^o$.

Feature Disentanglement in $\mathbf{z}_i^c$. For the second equation in Equation (7), the aggregated posterior distribution of $q_\phi(\mathbf{z}_i^c) = \int_{c_i} q_\phi(\mathbf{z}_i^c|\mathbf{c}_i)p_{cdata}(\mathbf{c}_i)d\mathbf{c}_i$. The $p_{cdata}(\mathbf{c}_i)$ is the item content data distribution. For the disentanglement of the content based representation $\mathbf{z}_i^c$, similarly, we decompose the KL term in Equation (6)

as:

$$\mathbb{E}_{p_{cdata}(\mathbf{c}_i)}[D_{KL}(q_\phi(\mathbf{z}_i^c|\mathbf{c}_i)||p(\mathbf{z}_i^c))] = I_q(\mathbf{c_i};\mathbf{z_i^c}) + D_{KL}(q_\phi(\mathbf{z}_i^c)||\Pi_k q_\phi(\mathbf{z}_{i,k}^c)) + \sum_k D_{KL}(q_\phi(\mathbf{z}_{i,k}^c)||p(\mathbf{z}_{i,k}^c)). \quad (10)$$

Thus, by using an independent prior distribution for each latent dimension of $\mathbf{z}_i^c$ and penalizing the KL term in Equation (6), we can encourage the disentanglement of content features. Similarly, the loss function of Equation (6) can be refined by adding a KL penalized parameter $\beta_2 > 1$:

$$L_{\beta_2}(\mathbf{c}_i;\phi) \equiv \mathbb{E}_{q_\phi(\mathbf{z}_i^c|\mathbf{c}_i)}(lnp_\phi(\mathbf{c}_i|\mathbf{z}_i^c)) - \beta_2 D_{KL}(q_\phi(\mathbf{z}_i^c|\mathbf{c}_i)||p(\mathbf{z}_i^c)), \quad (11)$$

to ensure the feature disentanglement of $\mathbf{z}_i^c$.

In sum, with Equations (9) and (11), the final loss function for DICER is: $L = L_{\beta_1}(\mathbf{x}_i;\theta) + \lambda L_{\beta_2}(\mathbf{c}_i;\phi)$, where $\beta = \{\beta_1, \beta_2\}$ are parameters used for feature disentanglement. λ is used to balance the two types of information.

3.3 Implementation

In this section, we provide details of the implementation of DICER as shown in the Figure 1(b): $p(\mathbf{z}_i^c)$, $p(\mathbf{z}_i^o)$ (the prior), $q_{\theta_1}(\mathbf{z}_i^o|\mathbf{x}_i, \mathbf{z}_i^c)$ and $q_{\phi_1}(\mathbf{z}_i^c|\mathbf{c}_i)$ (the encoder), $p_{\theta_2}(\mathbf{x}_i|\mathbf{z}_i^o, \mathbf{z}_i^c)$ and $p_{\phi_2}(\mathbf{x}_i|\mathbf{z}_i^o, \mathbf{z}_i^c)$ (the decoder), where the parameters $\theta = \{\theta_1, \theta_2\}$ and $\phi = \{\phi_1, \phi_2\}$. Specifically, the θ_1 and ϕ_1 are parameters for encoders, and the θ_2 and ϕ_2 are parameters for decoders.

Prior. To encourage feature disentanglement, as illustrated in Section 3.2, we set the prior of $\mathbf{z}_i^o$ and $\mathbf{z}_i^c$ to be the centered isotropic multivariate Gaussian distribution:

$$\mathbf{z}_i^o \sim N(\mathbf{0}, \mathbf{I}_{K_1}), \mathbf{z}_i^c \sim N(\mathbf{0}, \mathbf{I}_{K_2}).$$

The priors ensure the extracted features from different types of information (be they item content or the user-item interactions) are statistically independent in each dimension.

Encoder. To extract the content features $\mathbf{z}_i^c$ and the content disentangled collaborative features $\mathbf{z}_i^o$, there are two parts in DICER: the encoder for the user-item feedback and the encoder for the item content information. Since both the true posterior distribution $p_\theta(\mathbf{z}_i^o|\mathbf{x}_i, \mathbf{z}_i^c)$ and $p_\phi(\mathbf{z}_i^c|\mathbf{c}_i)$ are intractable, here we utilize the variational inference and reparametrization trick [13]. That is, we use a Gaussian distribution form with a diagonal covariance $q_\theta(\mathbf{z}_i^o|\mathbf{x}_i, \mathbf{z}_i^c)$ and $q_\phi(\mathbf{z}_i^c|\mathbf{c}_i)$ to approximate the true intractable posterior distributions.

Concretely, for $\mathbf{z}_i^o$, we assume:

$$lnq_{\theta_1}(\mathbf{z}_i^o|\mathbf{x}_i, \mathbf{z}_i^c) = lnN(\mu_u(\mathbf{x}_i, \mathbf{z}_i^c), diag\{\sigma_u^2(\mathbf{x}_i, \mathbf{z}_i^c)\}),$$

where the mean and standard deviation are parameterized by a neural network $f_{\theta_1}^{nn}$:

$$(\boldsymbol{\mu}^o, \boldsymbol{\sigma}^o) = f_{\theta_1}^{nn}(\mathbf{x}_i - \mathbf{U}^c \cdot \frac{\mathbf{z}_i^c}{||\mathbf{z}_i^c||}). \quad (12)$$

The $\mathbf{U}^c \in R^{|U|\times K_2}$ here is the user embedding matrix that represents the user preference towards item content features $\mathbf{z}^c$. Each row $\mathbf{u}_u^c \in R^{K_2}$ in $\mathbf{U}^c$ represents user u embedding. Since $\mathbf{x}_i$ is related to the joint contribution of item representation $\mathbf{z}_i^o$ and $\mathbf{z}_i^c$, we extract information in $\mathbf{x}_i$ that are not learned from $\mathbf{z}_i^c$ by $\mathbf{x}_i' = \mathbf{x}_i - \mathbf{U}^c \cdot \frac{\mathbf{z}_i^c}{||\mathbf{z}_i^c||}$. The $\cdot$ is the dot product. Here we first normalize the item representation and project it to the same space as $\mathbf{x}_i$ by multiplying with each user embedding $\mathbf{u}_u^c$. Thus, the learned representation $\mathbf{z}_i^o$ can capture the remaining factors in the user-item feedback. The neural network $f_{\theta_1}^{nn}$ is leveraged here to model the complex and non-linear relationship between $\mathbf{x}_i'$ and $\mathbf{z}_i^o$.

For $\mathbf{z}_i^c$, the encoder of $q_{\phi_1}(\mathbf{z}_i^c|\mathbf{c}_i)$ is similar to other VAE-based methods by using variational inference and $f_{\phi_1}^{nn}(\mathbf{c}_i)$, as shown in Figure 1(b).

Decoder. In DICER, we use a co-decoder to ensure the learned latent features in each type of information are appropriately encoded: one decoder $p_{\theta_2}(\mathbf{x}_{u,i}|\mathbf{z}_i^o, \mathbf{z}_i^c)$ is used to predict the user preference towards item i given both $\mathbf{z}_i^o$ and $\mathbf{z}_i^c$; the other one is the content-based decoder $p_{\phi_2}(\mathbf{c}_i|\mathbf{z}_i^c)$ to ensure the quality of the encoded content feature representation $\mathbf{z}_i^c$.

For $p_{\theta_2}(\mathbf{x}_{u,i}|\mathbf{z}_i^o, \mathbf{z}_i^c)$, we assume the distribution is proportional to the nonlinear transformation of both $\mathbf{z}_i^o$ and $\mathbf{z}_i^c$:

$$ln(p_{\theta_2}(\mathbf{x}_{u,i}|\mathbf{z}_i^o, \mathbf{z}_i^c)) \propto ln(g_{\theta_2}^{(u)}(\mathbf{z}_i^o) + g_{\theta_2}^{(u)}(\mathbf{z}_i^c)), \quad (13)$$

where $g_{\theta_2}^{(u)}(\mathbf{z}_i^o) = exp(cosine(\mathbf{z}_i^o, \mathbf{u}_u^o))$ and $g_{\theta_2}^{(u)}(\mathbf{z}_i^c) = exp(cosine(\mathbf{z}_i^c, \mathbf{u}_u^c))$. Here, $\mathbf{u}_u^o \in R^{K_1}$ is the user u embedding, which represents the user u preference towards features in z_i^o. The cosine similarity is used to connect user and item embeddings rather than the inner product similarity, since it can prevent mode collapse [3, 22]. Note here since item features are disentangled, the cosine also ensures the disentanglement of user features in each dimension [22].

For $p_{\phi_2}(\mathbf{c}_i|\mathbf{z}_i^c)$, similar to other VAE-based methods, we use the normal distribution with a single hidden fully-connected neural network and we found it has good performance. That is: $lnp_{\phi_2}(\mathbf{c}_i|\mathbf{z}_i^c) = lnN(\boldsymbol{\mu}_i', \boldsymbol{\sigma}_i'^2\mathbf{I})$, where $\boldsymbol{\mu}_i'$ and $\boldsymbol{\sigma}_i'$ are calculated based on $\mathbf{z}_i^c$ by using neural network $g_{\phi_2}^{nn}$ [13].

3.4 Variations of DICER

Our presentation so far has focused on a scenario in which we have user-item interactions and content information associated with items. In practice, DICER can also be used when only *user* content information is available (in place of item information) or when *both user and item* information are available. When only user content information is known, we can simply change DICER from item-based to user-based by reversing users and items. That is, instead of encoding each item feedback $\mathbf{x}_i$ across users, we encode each user feedback $\mathbf{x}_u$ towards different items. When both user and item content information are known, we can do both user- and item-based DICER, and combine the results together, similar to many integrated auto-encoder based methods.

4 EXPERIMENTS

In this section, we investigate the following key research questions: (i) What is the recommendation performance of DICER compared with state-of-the-art methods that do not disentangle content-collaborative information? How does this performance vary for different numbers of latent dimensions? (ii) What impact do the

Table 2: Summary of the three Amazon datasets.

Dataset	# Users	# Items	# Feedback	Sparsity
Clothing	2,872	28,586	43,418	0.053%
Beauty	2,513	39,746	91,044	0.091%
Toys&Games	2,771	58,306	106,131	0.066%

disentanglement design choices have on DICER? and (iii) Are the learned features in DICER really disentangled? Finally, we visualize the disentanglement learning representation to illustrate how it facilitates recommendation.

4.1 Datasets

To evaluate the importance of disentangling content-collaborative information, we require datasets containing interaction data (e.g., clicks, purchases) as well as content information. Hence, we adopt three real-world publicly accessible Amazon datasets [8, 24] which cover rich and commonly used content information for items – visual images and text descriptions. Other kinds of content information can be easily incorporated into the proposed model as discussed in Section 3.4. For items, we choose three domains that are widely used: *Clothing*, *Toys&Games*, and *Beauty*. We select users with more than 10 reviews in *Clothing*, 20 reviews in *Beauty* and 25 reviews in *Toys&Games* for different levels of feedback sparsity, as shown in Table 2. We extract UserID, ItemID, and the rating scores to indicate whether the user purchased the item (1 represents a user purchase, 0 otherwise). For item-based content information, we consider the images associated with each item (which were collected in [24]) and the text-based descriptions provided by the seller. Note here all the items are considered. Thus the datasets have a long-tail distribution, which is challenging in recommendation since many items have very little feedback [34]. We randomly partition the implicit feedback of each user into 80% for training, and the remaining as testing, reserving 20% of the training data as validation.

4.2 Setup

Modeling Item Content. For the content information $\mathbf{c}_i$, here we consider the widely used item images and text-based descriptions [7, 15, 35]. For fairness, all content-aware baselines use the same item content as input. Other sources of item content could also be incorporated into DICER, and other pre-processing steps can also be adapted here. Concretely, for item images, we apply the same method as [8, 24, 35] to get the high-level visual feature vector $\mathbf{m}_i \in R^{4096}$. For item descriptions, we use the same word2vec-based method [35] which turns the description paragraph into a fixed-length feature vector $\mathbf{t}_i$. Based on [35], here we set $\mathbf{t}_i \in R^{1000}$. The $\mathbf{c}_i \in R^{5096}$ is the concatenation of $\mathbf{m}_i$ and $\mathbf{t}_i$.

Evaluation Metrics. Following many previous works [10], we adopt NDCG at top-k (N@k) and hit ratio at top-k (HR@k) for evaluating personalized ranking. The HR@k measures the fraction of purchased items that appear in top-k recommendation lists across all users. The N@k takes the position of correctly recommended items into account by assigning higher scores to the top hits.

Baselines. We compare DICER with the following competitive baselines, with particular emphasis on VAE-based methods for comparison.[2] The same content information is used for all content-based methods.

- *POP*. Items are ranked by their popularity based on user's interactions with items.
- *CDAE* [32]. Collaborative Denoising Auto-Encoder uses auto-encoders to find the relationship between users and items based on implicit feedback. The number of negative samples is set to $q = 100$ which is in line with the other negative sampling methods.
- *Multi-VAE* [17]. This is a classic VAE-based latent factor models for recommendation. The implicit feedback from users is treated as being generated from a multinomial likelihood for recommendation.
- *NGCF* [31]. This is a state-of-the-art collaborative filtering approach based on graph neural networks. It treats use-item interactions as a bipartite graph and propagates the user and item embeddings on the graph to explore the high-order connectivity between users and items.
- *CBPR* [7]. This is a classic content-based Bayesian Personalized Ranking [7] method, which is widely used for its robust and strong performance.
- *CVAE* [15]. CVAE is a content-based VAE model that uses a Bayesian generative model to first learn the item content information and then user-item implicit feedback through an inference network. It adds the item latent content variable and collaborative latent variable as the joint representation of each item.
- *CNGCF*. This is an augmented version of NGCF that incorporates item content information [31]. Specifically, we concatenate item content information with latent factors as the joint representation, and then we propagate three layer's embeddings in the user-item bipartite graph for the final recommendation.

Parameter settings. The dimension of latent factors and hidden dimensions for each method is empirically set to be 40 (for the impact of such choices, see Section 4.3.2). Regularization terms are determined by grid search in the range of {0.1, 0.01, 0.001, 0.0001, 0.00001}. β in VAE-based methods are determined from the range {1.0, 0.9, ...,0.1, 0.01, 0.001, 0.0001, 0.00001}. The drop out rate is also chosen by grid search in the range of {0.1, 0.3, 0.5, 0.7, 0.9} and the learning rate is in the range of {0.0001, 0.001, 0.01, 0.1}. Model parameters are first randomly initialized according to truncated normal distributions with mean 0 and standard deviation 0.001. All experimental settings, code, data, and the DICER algorithms will be released publicly at http://people.tamu.edu/~zhan13679/.

4.3 Top-K Recommendation

We first compare the Top-K recommendation performance of all methods, and then vary the key hyperparameter – the number of latent factor dimensions – to further investigate its effect on recommendation.

[2]We also experimented with neural machine factorization [9] but the training process is time consuming and its performance is not good here.

Table 3: N@K and HR@K of DICER and baselines. Δ is the difference between DICER and the next-best alternative (marked underline).

Dataset	Measure%	User-Item Interaction				User-Item Interaction + Content				
		POP	CDAE	Multi-VAE	NGCF	CBPR	CVAE	CNGCF	DICER	Δ
Clothing	N@5	0.423	0.833	0.693	1.311	0.812	1.228	1.380	**1.661**	20.4%
	N@10	0.616	1.111	1.031	1.735	1.109	1.682	1.809	**2.177**	20.3%
	HR@5	0.731	1.462	1.253	1.950	1.288	2.089	2.089	**2.646**	26.7%
	HR@10	1.358	2.228	2.159	3.203	2.089	3.412	3.377	**4.039**	18.4%
Toys	N@5	0.997	3.139	3.350	3.522	2.612	3.502	4.560	**4.984**	9.3%
	N@10	1.454	4.273	4.490	4.835	3.728	4.991	6.074	**7.017**	15.5 %
	HR@5	1.877	5.485	5.413	5.702	4.186	5.521	7.326	**8.156**	11.3%
	HR@10	3.284	8.697	8.336	9.347	7.434	9.780	11.332	**13.389**	18.2%
Beauty	N@5	2.354	6.338	6.739	6.556	5.922	7.230	7.054	**7.753**	7.2%
	N@10	3.449	8.567	9.005	9.017	7.755	9.464	9.693	**10.477**	8.1 %
	HR@5	4.338	11.421	10.864	10.744	9.391	11.421	11.779	**12.694**	7.8%
	HR@10	7.402	16.514	15.002	16.076	13.490	16.275	17.589	**18.305**	4.1%

4.3.1 Overall Comparison. For fair comparison, we set the latent dimension K to be the same for all methods. Notice that CBPR, CVAE and CNGCF all use the same item content information (images and descriptions) as DICER. We report the N@k and HR@k (for k at 5, 10) for the three datasets in Table 3. Overall, we see that DICER consistently outperforms the next-best performing baseline for all datasets and for all metrics. Concretely, we have the following key observations:

First, methods that incorporate item content information generally achieve better performance comparing with corresponding methods that only rely on user-item interactions. For example, the HR@K and N@k of the content-based latent factor model CNGCF is higher than NGCF for the three datasets. This verifies the importance of incorporating additional item content information for improving recommendation performance.

Second, among methods, DICER consistently achieves the best performance over all datasets, as shown in Table 3 Δ column. The improvement demonstrates that by carefully disentangling content-collaborative features, DICER is able to enhance the learning of diverse features that influence user preference towards items. Particularly, comparing with CVAE which considers the same content information as DICER, the large improvement of DICER further confirms that the disentangling in DICER can effectively deal with the complicated relationship between item content information and user interactions, which strengthens DICER to discover different features among the two types of information for recommendation improvement.

Third, among datasets, DICER gives a relatively larger improvement for the sparsest dataset (Clothing). This may be because DICER not only considers the content information that is helpful for sparse data, but also utilizes disentanglement to discover features that cover a wider space rather than duplicated or related features. This further shows the importance of disentangling content and collaborative information. An interesting finding is that comparing the improvement from NGCF to content-based CNGCF, the improvement of DICER is higher in VAE-based methods. Such high improvement may be attributed to the learning of disentangled features rather than directly concatenating content and collaborative features in CNGCF. This shows the benefit of capturing the disentangled features for recommendation.

4.3.2 Influence of Latent Dimension. We also analyze the effect of the key hyperparameter – the number of latent factors – on DICER and representative baselines. Results for the Toys dataset is shown in Figure 2. Similar results hold for the other two datasets. We observe that DICER consistently outperforms the other methods for different numbers of latent dimensions.

4.4 Ablation Study

Given the good performance of DICER versus baselines, what impact do the design choices have on its performance? Specifically, does the disentanglement approach in DICER effectively incorporate item content and collaborative information to enhance recommendation (encoder in Equation (12) and decoder in Equation (13))? In this section, we explore several variants to incorporate item content information compared with DICER and analyze their effects:

- *Original* $\mathbf{x}$: $\mathbf{z}_i^o$ is encoded based on the original user-item interactions $\mathbf{x}_i$ rather than the content disentangled one. That is, $\mathbf{z}_i^o$ and $\mathbf{z}_i^c$ are separately learned without considering their disentanglement;
- *Nonlinear*: In Equation (12), instead of using $\mathbf{z}_i^c$, here we add a *tanh* layer to project $\mathbf{z}_i^c$ to the same space as $\mathbf{x}_i$, then encode $\mathbf{z}_i^o$. It may influence the disentanglement relation between $\mathbf{z}_i^o$ and $\mathbf{z}_i^c$;
- *LatentConcat*: We directly concatenate $\mathbf{z}_i^o$ and $\mathbf{z}_i^c$ as the joint latent representation to reconstruct $\mathbf{x}_i$. That is, we use $ln(g_{\theta_2}^{(u)}(concate(\mathbf{z}_i^o, \mathbf{z}_i^c))$ in Equation (13) to formulate $ln(p_{\theta_2}(\mathbf{x}_{u,i}|\mathbf{z}_i^o, \mathbf{z}_i^c))$;
- *LatentAdd*: $\mathbf{z}_i^o$ and $\mathbf{z}_i^c$ are directly added as the joint latent representation to reconstruct $\mathbf{x}_i$. That is, we use $ln(g_{\theta_2}^{(u)}(\mathbf{z}_i^o + \mathbf{z}_i^c))$ in Equation (13).

The results of the ablation study for the Toys dataset are shown in Table 4. The *Default* row shows the DICER results. We observe

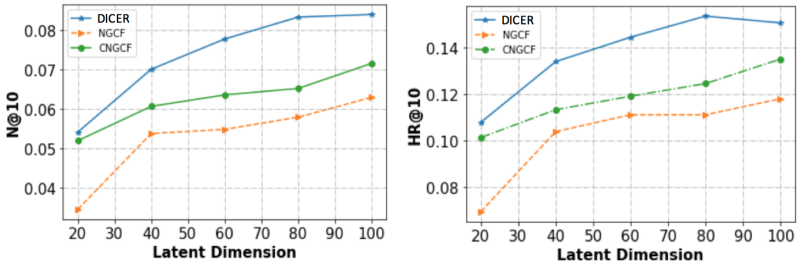

Figure 2: Recommendation performance for different latent dimensions in Toys dataset. Similar results hold for the other two datasets.

Table 4: Ablation study on Toys dataset. Similar results hold for the other two datasets.

%	N@5	N@10	HR@5	HR@10
Default	4.984	7.017	8.156	13.389
Original $\mathbf{x}$	4.071	5.614	6.712	10.682
Nonlinear	4.659	6.206	7.615	11.620
LatentConcat	4.756	6.504	7.470	12.090
LatentAdd	4.828	6.6520	7.795	12.450

that DICER outperforms the other variations that consider the same item content information, which further shows DICER can more effectively model the relationship between item content and user-item interaction information for recommendation improvement.

Concretely, the performance of DICER is superior to *Original* $\mathbf{x}$ method which uses raw user-item feedback. This demonstrates that disentangling evidence learned from item content to encode $\mathbf{z}_i^o$ can bring a large improvement, compared to directly using the raw $\mathbf{x}_i$. This is reasonable since the disentanglement can encourage the learned features in $\mathbf{z}_i^o$ and $\mathbf{z}_i^c$ to cover more diverse independent features in item representations, and thus enhance the learning of user preference towards these items. An interesting finding is that comparing the nonlinear transform (the *Nonlinear* method), DICER achieves a better performance. This is probably because the nonlinear calculation may have difficulties maintaining disentanglement among features and can be more prone to cause overfitting.

Furthermore, DICER outperforms methods that directly concatenate or add $\mathbf{z}_i^c$ to $\mathbf{z}_i^o$ to jointly reconstruct the user-item feedback, such as the *LatentConcat* and *LatentAdd*. This indicates that explicitly decoding item content separately can more precisely capture item content features to further help model user-item interaction.

4.5 Disentanglement

Since the disentanglement representation learning plays a pivotal role in DICER, we further explore its influence on the recommendation performance from both the content-collaborative disentanglement and feature disentanglement perspective. Following [22], we measure disentanglement based on statistical independence. We vary β and plot the relationship between the disentanglement and recommendation performance, where β is the parameter of disentanglement for corresponding latent factors.

4.5.1 Content-Collaborative Disentanglement. The Content-Collaborative disentanglement is measured by

$$Average_{k,j}(1 - Corr_i(z_{i,k}^c, z_{i,j}^o)),$$

where $z_{i,k}^o$ represents the kth dimension of $\mathbf{z}_i^o$. Similar notation holds for $z_{i,j}^c$. $Corr_i(z_{i,k}^c, z_{i,j}^o)$ is the correlation between each dimension in the corresponding $\mathbf{z}_i^c$ and $\mathbf{z}_i^o$ across items. The results are shown in Figure 3(a). Note here since DICER outperforms the other methods in N@10, the lines for the other methods stop earlier than the line for DICER.

From Figure 3(a), we observe that (1) For the content-collaborative disentanglement, DICER achieves a good disentanglement when N@10 is high. This indicates that DICER can effectively capture the relatively statistically independent features between item content and user-item interaction information; (2) The figure also demonstrates that good recommendation performance is related to a relatively high disentanglement, which is consistent with [22]. This makes sense since the disentangled features are more stable [4, 12] and can cover a variety of user preferences.

4.5.2 Feature Disentanglement. Similarly, the item and user feature disentanglement are measured based on:

$$\begin{aligned}
&Average_{1\le k_1<k_2\le K_1}(1 - Corr_i(z_{i,k_1}^o, z_{i,k_2}^o)) \\
&\quad + Average_{1\le j_1<j_2\le K_2}(1 - Corr_i(z_{i,j_1}^c, z_{i,j_2}^c)) \\
&\quad + Average_{k,j}(1 - Corr_i(z_{i,k}^c, z_{i,j}^o)), \\
&Average_{1\le k_1<k_2\le K_1}(1 - Corr_u(u_{u,k_1}^o, u_{u,k_2}^o)) \\
&\quad + Average_{1\le j_1<j_2\le K_2}(1 - Corr_u(u_{u,j_1}^c, u_{u,j_2}^c)) \\
&\quad + Average_{k,j}(1 - Corr_u(u_{u,k}^c, u_{u,j}^o)),
\end{aligned}$$

respectively. The calculation of $Corr_u(\cdot,\cdot)$ is similar as $Corr_i(\cdot,\cdot)$ but across users instead of items. Figure 3(b)(c) shows the disentanglement results.

We have the following key observations: (1) Similar to the content-collaborative disentanglement finding, when N@10 is high, DICER achieves high feature disentanglement. This indicates DICER can also capture good statistically independent features at a granular level. (2) For CVAE, though it has a good disentanglement for item and content-collaborative features, the disentanglement for user features is much lower than the other two methods. This may be since DICER and LatentAdd jointly model the disentanglement of user and item features, the user representation can capture the

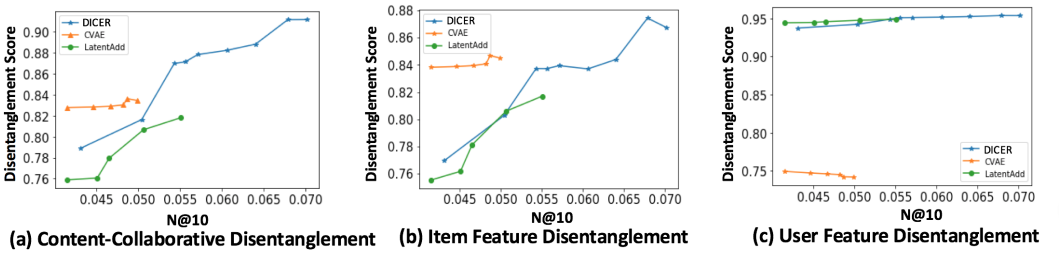

Figure 3: Disentanglement Results for N@10 on Toys dataset. We vary the disentanglement parameter β and plot the relationship between the disentanglement and recommendation performance. The higher the disentanglement score, the more disentanglement between corresponding features.

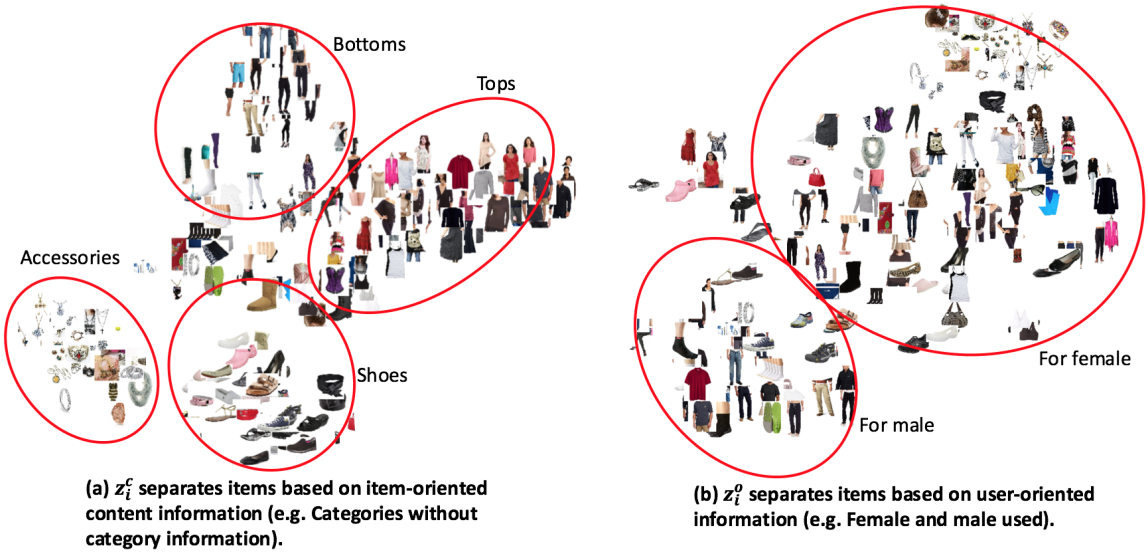

Figure 4: The 2-D visualization (with t-SNE [30]) of items in Clothing datasets by using (a) $\mathbf{z}_i^c$ (b) $\mathbf{z}_i^o$.

disentangled features based on item features, while CVAE considers them separately. (3) Another finding is that the item feature disentanglement is lower than content-collaborative disentanglement. One possible reason is that content and collaborative representations are calculated from different types of information.

4.6 Case Studies of Disentanglement

In this section, we illustrate how the disentangled representation approach facilitates recommendation.

First, we visualize the content-collaborative disentanglement representations $\mathbf{z}_i^c$ and $\mathbf{z}_i^o$, as shown in Figure 4(a)(b). We observe that (1) Comparing Figure 4(a) and (b), $\mathbf{z}_i^c$ can nicely separate different categories of items even without knowledge of the ground-truth categories. That is, items are separated mainly based on item content oriented information. For example, as shown in Figure 4(a), the accessories, shoes, tops and bottoms are in different clusters. Different from $\mathbf{z}_i^c$, in Figure 4(b), $\mathbf{z}_i^o$ separates items by user-oriented information, e.g., items used by male (left bottom) and female (right top). Items that can be bought together (e.g., bottoms and shoes) are also close to each other. This indicates $\mathbf{z}_i^c$ and $\mathbf{z}_i^o$ can capture item features in very different aspects (item content and user aspects) by disentanglement. (2) Furthermore, in each cluster of (a), items with similar content information are relatively close to each other. For example, items in similar colors are close to each other, and items with dark colors are far from the light color items, such as for shoes and tops. This demonstrates that $\mathbf{z}_i^c$ can capture reasonable content features. In sum, by using content-collaborative disentanglement, $\mathbf{z}_i^c$ and $\mathbf{z}_i^o$ are complementary in how they discover item features based on different aspects to enhance the prediction of user preference towards items.

Next, we illustrate the features that DICER learns in each dimension at the granular level. We vary the target dimension and list the items that have similar values in the target dimension. Some representative examples are shown in Figure 5. Each row shows items that have similar values in a target dimension. For each row, we see DICER is able to capture very different features in each dimension without item category information, and some of them may be interpretable. For example, in Figure 5, the first target dimension likely captures the circle-based characteristic of items. The second target dimension captures the shoe features. The clothing items in the third row have similar styles. For the fourth row, all the items have colorful patterns. This highlights how DICER can capture independent and interpretable features in each dimension.

Figure 5: Each row shows items that have similar values in one target dimension of item representations.

5 CONCLUSION

In this paper, we focus on disentangled representation learning from both content information and user-item interactions to enhance recommendation. By disentangling the content-collaborative features and each feature at a granular level, the proposed method DICER learns features that are relatively statistically independent and diverse, leading to a more powerful recommender. Through extensive experiments, DICER outperforms the next-best baseline by 13.5% and 14.4% on average in NDCG and hit ratio. In our continuing work, we are interested in extending DICER to other scenarios, e.g., where a user's social network is available.

ACKNOWLEDGMENTS

This work was supported in part by NSF grant IIS-1841138 and DARPA W911NF-16-1-0565.

REFERENCES

[1] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. 2016. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410* (2016).
[2] Chris Burgess and Hyunjik Kim. 2018. 3D Shapes Dataset. https://github.com/deepmind/3dshapes-dataset/.
[3] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. 2018. Understanding disentangling in β-VAE. *arXiv preprint arXiv:1804.03599* (2018).
[4] Tian Qi Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. 2018. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*. 2610–2620.
[5] Zhiyong Cheng, Ying Ding, Lei Zhu, and Mohan Kankanhalli. 2018. Aspect-aware latent factor model: Rating prediction with ratings and reviews. In *Proceedings of the World Wide Web conference*. 639–648.
[6] Ryuhei Hamaguchi, Ken Sakurada, and Ryosuke Nakamura. 2019. Rare Event Detection using Disentangled Representation Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 9327–9335.
[7] Ruining He and Julian McAuley. 2016. VBPR: visual bayesian personalized ranking from implicit feedback. In *AAAI conference on artificial intelligence*.
[8] Ruining He, Charles Packer, and Julian McAuley. 2016. Learning compatibility across categories for heterogeneous item recommendation. In *IEEE International Conference on Data Mining*. IEEE, 937–942.
[9] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the International ACM SIGIR conference on Research and Development in Information Retrieval*. 355–364.
[10] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the international conference on World Wide Web*. 173–182.
[11] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the International ACM SIGIR conference on Research and Development in Information Retrieval*. 549–558.
[12] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. *Proceedings of the International Conference on Learning Representations (ICLR)* 2, 5 (2017), 6.
[13] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).
[14] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.
[15] Xiaopeng Li and James She. 2017. Collaborative variational autoencoder for recommender systems. In *Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining*. 305–314.
[16] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. 2014. GeoMF: joint geographical modeling and matrix factorization for point-of-interest recommendation. In *Proceedings of the ACM SIGKDD international conference on Knowledge discovery and data mining*. 831–840.
[17] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the World Wide Web Conference*. 689–698.
[18] Francesco Locatello, Gabriele Abbati, Thomas Rainforth, Stefan Bauer, Bernhard Schölkopf, and Olivier Bachem. 2019. On the fairness of disentangled representations. In *Advances in Neural Information Processing Systems*. 14611–14624.
[19] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. 2019. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning (ICML)*. 4114–4124.
[20] Zhongqi Lu, Zhicheng Dou, Jianxun Lian, Xing Xie, and Qiang Yang. 2015. Content-based collaborative filtering for news topic recommendation. In *AAAI conference on artificial intelligence*.
[21] Junmei Lv, Bin Song, Jie Guo, Xiaojiang Du, and Mohsen Guizani. 2019. Interest-related item similarity model based on multimodal data for top-N recommendation. *IEEE Access* 7 (2019), 12809–12821.
[22] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learning disentangled representations for recommendation. In *Advances in Neural Information Processing Systems*. 5711–5722.
[23] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. 2015. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644* (2015).
[24] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM.
[25] Andriy Mnih and Russ R Salakhutdinov. 2008. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*. 1257–1264.
[26] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International Conference on Data Mining*. IEEE, 995–1000.
[27] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the International Conference on World Wide Web*. 811–820.
[28] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*. 3483–3491.
[29] Qingquan Song, Shiyu Chang, and Xia Hu. 2019. Coupled Variational Recurrent Collaborative Filtering. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 335–343.
[30] Laurens Van Der Maaten. 2014. Accelerating t-SNE using tree-based algorithms. *The Journal of Machine Learning Research* 15, 1 (2014), 3221–3245.
[31] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the international ACM SIGIR conference on Research and development in Information Retrieval*. 165–174.
[32] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. 2016. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the ACM International Conference on Web Search and Data Mining*. 153–162.
[33] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. 2017. Deep Matrix Factorization Models for Recommender Systems. In *Proceedings of the International Joint Conference on Artificial Intelligence*, Vol. 17. Melbourne, Australia, 3203–3209.
[34] Xinyang Yi, Ji Yang, Lichan Hong, Derek Zhiyuan Cheng, Lukasz Heldt, Aditee Kumthekar, Zhe Zhao, Li Wei, and Ed Chi. 2019. Sampling-bias-corrected neural modeling for large corpus item recommendations. In *Proceedings of the ACM Conference on Recommender Systems*. 269–277.
[35] Yin Zhang, Haokai Lu, Wei Niu, and James Caverlee. 2018. Quality-aware neural complementary item recommendation. In *Proceedings of the ACM Conference on Recommender Systems*. 77–85.
[36] Tong Zhao, Julian McAuley, and Irwin King. 2014. Leveraging social connections to improve personalized ranking for collaborative filtering. In *Proceedings of the ACM International Conference on Information and Knowledge Management*. 261–270.