8-2022

Aligning dual disentangled user representations from ratings and textual content

Nhu Thuat TRAN
Singapore Management University, nttran.2020@phdcs.smu.edu.sg

Hady Wirawan LAUW
Singapore Management University, hadywlauw@smu.edu.sg

Aligning Dual Disentangled User Representations from Ratings and Textual Content

Nhu-Thuat Tran
Singapore Management University
Singapore
nttran.2020@phdcs.smu.edu.sg

Hady W. Lauw
Singapore Management University
Singapore
hadywlauw@smu.edu.sg

ABSTRACT

Classical recommendation methods typically render user representation as a single vector in latent space. Oftentimes, a user's interactions with items are influenced by several hidden factors. To better uncover these hidden factors, we seek disentangled representations. Existing disentanglement methods for recommendations are mainly concerned with user-item interactions alone. To further improve not only the effectiveness of recommendations but also the interpretability of the representations, we propose to learn a second set of disentangled user representations from textual content and to align the two sets of representations with one another. The purpose of this coupling is two-fold. For one benefit, we leverage textual content to resolve sparsity of user-item interactions, leading to higher recommendation accuracy. For another benefit, by regularizing factors learned from user-item interactions with factors learned from textual content, we map uninterpretable dimensions from user representation into words. An attention-based alignment is introduced to align and enrich hidden factors representations. A series of experiments conducted on four real-world datasets show the efficacy of our methods in improving recommendation quality.

CCS CONCEPTS

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

KEYWORDS

Disentangled Representation; Textual Content-Aware Recommender Systems; User Preferences Interpretation

ACM Reference Format:
Nhu-Thuat Tran and Hady W. Lauw. 2022. Aligning Dual Disentangled User Representations from Ratings and Textual Content. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3534678.3539474

1 INTRODUCTION

Recommender systems play an indispensable role in delivering relevant content to users, alleviating the information overload. The core of an effective personalized recommender system lies at learning expressive representations [21] to capture user's preferences from adoptions and interactions with various items. Ideally this user representation that we seek is not only amenable to prediction accuracy, but is also *interpretable*, so we can make sense of the user's underlying preferences, as well as *controllable*, so we could potentially let users have some control in terms of adaptively retrieving their preferred items according to specific preferences.

Classical methods enmeshing a user's multi-faceted preferences into a monolithic representation would not allow us to appreciate the respective facets meaningfully. Disentangled learning could address these by separating the distinct, informative factors of variations in data [3]. Various methods have been proposed to learn disentangled representations, primarily in the field of computer vision [5, 7, 11]. Its application to recommender systems in discovering disentangled factors underlying user's interactions with a set of items is pioneered by MacridVAE [26], followed by other recent pursuits for disentangled representations in recommendation systems research [27, 29, 30, 44, 50, 52].

However, a couple of shortcomings remain. For one, it is difficult to interpret a factor without further associated interpretable information [23]. Unlike in computer vision where disentangled factors could yield visualizable artefacts, in recommender systems these factors capture 'interactions' abstractly. For another, existing disentangled representations for recommendations primarily rely on interaction data alone, which is known to have sparsity issues with each user interacting with merely a minuscule fraction of all items in the catalogue of interest. To address these shortcomings, we propose to incorporate side information in the form of textual content associated with each user into disentangled representation learning for recommendation. For instance, this user-associated content could be derived from the collection of reviews the user has written or product descriptions the user has adopted.

Our approach is to derive a second set of disentangled representation for the user based on text content, in addition to that based on item interactions, and to align the hidden factors of these dual representations. One expected benefit of the alignment is that text-based factors would provide semantic interpretability of the corresponding interaction-based factors via the former's associations with words. Another benefit is the enhanced accuracy that we gain from supplementing the interaction data with textual data. While textual content has been incorporated into recommendation models with good results [15, 20, 24, 41, 43], they were primarily studied in the classical non-disentangled representational sense. Our work, in contrast, builds a disentangled textual modeling network, uncovering hidden factors in textual content and align these factors with ones capturing user's adoption. Hypothetically, this

KDD '22, August 14–18, 2022, Washington, DC, USA

ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539474

allows knowledge transfer between hidden factors learned from textual content and user-item interactions.

To this end, we propose a novel textual-aware disentangled recommendation model, called AddVAE. Our model consists of two sub-networks, one for user-item interaction modeling and the other for user's textual content. Each sub-network is responsible for discovering the hidden factors underlying their corresponding input data. Individually, each is based on MacridVAE [26]. To realize the alignment of hidden factors from these two sub-networks, we employ compositional de-attention mechanism [36] and then regularize by Mutual Information (MI) maximization. By means of MI maximization, we explicitly embed collaborative filtering signals into text-based factors and vice versa so as to achieve both goals of resolving sparsity and interpretability simultaneously.

Contributions We make the following contributions. *First*, to the best of our knowledge, this is the first time a content-aware recommendation model is studied in disentangled fashion by deriving k factors underlying user-item interactions and user's adopted textual content. By equipping disentangled textual content modeling, our method not only improves recommendation performance but also is able to interpret user's preferences at the fine-grained level of factors. *Second*, we propose a novel user-oriented content-aware recommendation model, called AddVAE, that incorporates textual content into recommendation model by coupling disentangled factors from two MacridVAE networks via a mutual information maximization strategy. We also propose to leverage compositional de-attention network to effectively align disentangled factors. *Third*, we conduct extensive experiments on four real-world datasets. Experimental results demonstrate that AddVAE not only enjoys better recommendation quality compared to several strong baselines but also provides interpretation of disentangled user's preferences.

2 RELATED WORK

Textual Modeling in Recommender Systems Classical frameworks for recommender systems include matrix factorization [19], algebraic method [32], autoencoder [21, 34], neural network [10], graph-based model [9], etc. To address sparsity, various techniques have been proposed to incorporate textual content. CTR [41] leverages Latent Dirichlet Allocation (LDA) [4]. Stacked Denoising Autoencoder (SDAE) [43] is leveraged in CDL [43], CKE [48]. Variational Autoencoder (VAE) replaces SDAE in CVAE [20]. These models ignore order of words in the input. ConvMF [15], CRAE [42] models order of words in textual content using Convolutional Neural Network and Recurrent Neural Network, respectively. GATE [24] explores attention mechanism to incorporate textual content into autoencoder-based recommedation. There are two key distinctions to our work. First, in contrast to how these models focus on item information, we model text information on the user-side. Second, these models represent textual content as a single vector, whereas we model it in a disentangled fashion.

Disentangled Representation Learning Early successful applications are mostly in computer vision domain such as [5, 7, 11], in which VAE [17] is exploited as backbone network. Recent work has explored disentangled representations beyond image data, e.g., graph data [25, 45], knowledge graph embedding[46, 49]. In recommender systems, MacridVAE [26] learns macro- and micro-disentangled level underlying user adoptions. A suite of following works learn disentangled representations from user-item interactions in various forms such as graph [44], sequential recommendation [27], critiquing recommendation [30], causal embedding [52], generative model [22]. DICER [50] focuses on disentangling content and collaborative signals. Besides recommendation performance, researchers are also interested in the scalability [6, 47]. KDR [29] incorporates side information from knowledge graph on item side. Our key distinction is to focus not only on one but two sets of disentangled representations by incorporating *textual* content on *user* side. We also propose an expressive *alignment* between the dual representations.

Mutual Information Maximization Mutual information (MI) measures how well a random variable knows another random variables. MI is often used as objective to maximize in order to learn expressive representations. Deep InfoMax [12] is the initial work that populates the use of MI maximization. Deep InfoMax maximizes MI between a local image patch and its global context in a contrastive learning task. MI maximization then becomes popular in various applications, including computer vision [1], natural language processing [8, 18], neural topic model [31], graph representation learning [35, 39], recommender systems [29, 53]. MI usually is intractable in the context of neural network. Hence, researcher proposed methods to estimate lower bound of MI [2, 33, 37]. Since a comprehensive comparison between MI estimators is out of scope of this work, we follow [33] to estimate MI lower bound and leave the exploration of other MI estimators for future work.

3 METHODOLOGY

In this section, we describe our proposed model called AddVAE, which stands for *Aligned Dual Disentangled VAE*. Specifically, our proposed architecture design aims at finding an effective method to combine disentangled factors discovered by two networks learned from user-item interactions (ratings[1]) signals and textual content signals respectively. Figure 1 gives an illustration of our proposed model. Table 1 lists the main notations in this paper.

3.1 Disentangled Representation Module

The key design is to couple factors from two disentangled representation modules from user's rating as well as user's associated textual content. We base each component module on Macro-Micro Disentangled Variational Autoencoder [26], abbreviated as MacridVAE. It is a generative model based on Variational Autoencoder [17]. The goal is to derive K hidden factors, $\{\mathbf{z}_k\}_{k=1}^{K}, \mathbf{z}_k \in \mathbb{R}^d$, underlying input data. In the case of product recommendation, these represent K latent factors (which could be categories or genres) that a user is interested in. Learning K latent factors for a given input is aimed at achieving *macro* disentanglement. On the other hand, *micro* disentanglement requires each dimension of disentangled factor $\mathbf{z}_k$ to capture fine-grained aspects of items belonging to the latent category represented by $\mathbf{z}_k$. Like any VAE, MacridVAE consists of two parts, encoder and decoder.

[1]We call user-item interactions and user's ratings interchangeably

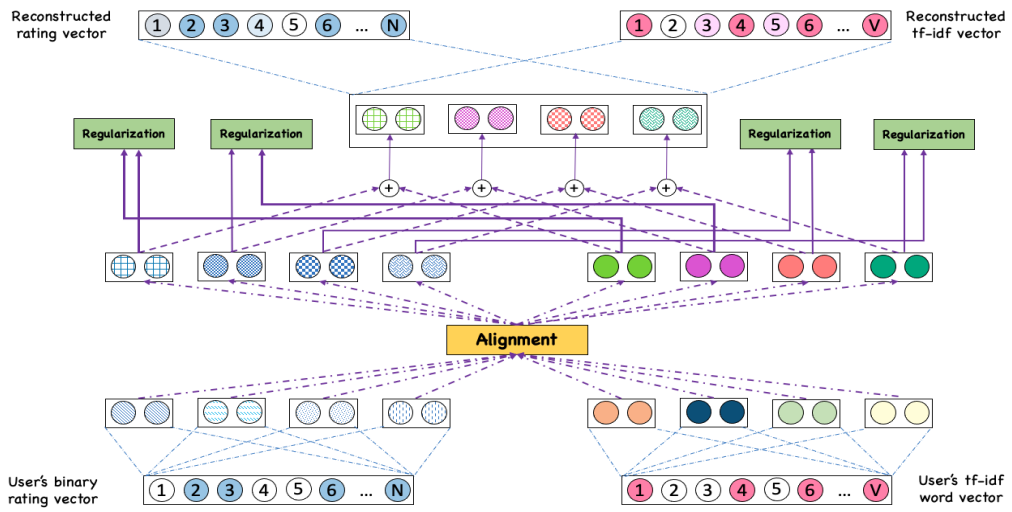

Figure 1: Illustration of our model AddVAE. Best viewed in color. Shaded circles in user's binary rating vector are observed interactions. Box with two circles represents a factor. Factors with patterns represent rating factors while colored factors represent textual factors. Different patterns (colors) indicate different factors. Combination of pattern and color means addition.

Table 1: List of Notations

Notation	Description
$\mathcal{U}, u$	the set of users and user index, respectively
i	input index, e.g., an item or a word
$\mathbf{x}_u, \mathbf{x}_{u,i}$	vector representation indexed by u and its i^{th} element
$i : \mathbf{x}_{u,i} = 1$	set of indices i where $\mathbf{x}_{u,i} = 1$
N, V	number of items and number of words, respectively
k	factor index
K	number of factors
d	dimensionality of factor, item and word representations
r, t	(without any other specification) rating and text
$\mathbf{z}_k^r, \mathbf{z}_k^t$	k-th disentangled factor from CF-MacridVAE and Text-MacridVAE, respectively
$\mathbf{z}_k^{r\|t}$	k-th aligned rating factor after aligning with texts
$\mathbf{z}_k^{t\|r}$	k-th aligned text factor after aligning with ratings
$\mathbf{A}, \mathbf{A}^T$	matrix and transposed matrix
$\mathbf{a}, \mathbf{a}^T$	vector and transposed vector

Encoder. Let $\mathbf{x}_u \in \mathbb{R}^N$ as N-dimension input vector, e.g., binary input rating vector of user u. A neural network with non-linear activation, $f_{nn}(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^{2d}$, first projects $\mathbf{x}_u$ into two parts:

$$(\mathbf{a}_u^k, \mathbf{b}_u^k) = f_{nn}(\frac{\sum_{i:\mathbf{x}_{u,i}=1} \mathbf{c}_{i,k}\mathbf{t}_i}{\sqrt{\sum_{i:\mathbf{x}_{u,i}=1} \mathbf{c}_{i,k}^2}}) \tag{1}$$

in which $\mathbf{t}_i \in \mathbb{R}^d$ is the context vector, which is the i^{th}-row of context matrix $\mathbf{T} \in \mathbb{R}^{N\times d}$ used by encoder to compute representation of each factor, $\mathbf{c}_{i,k}$ is the probability that i^{th} element in the input, e.g., item when the input is user's rating vector, belongs to the k^{th} prototype. Without any specification, a prototype refers to a factor underlying data. Vector $\mathbf{c}_i \in \mathbb{R}^K$ is an approximated one-hot vector drawn from categorical distribution (CATE) and estimated using Gumbel-Softmax [14, 28] during training.

$$\mathbf{c}_i \sim CATE(SOFTMAX([\mathbf{s}_{i,1}, \mathbf{s}_{i,2}, .., \mathbf{s}_{i,K}])),\ \ \mathbf{s}_{i,k} = \frac{\mathbf{h}_i^T \mathbf{m}_k}{\tau \cdot ||\mathbf{h}_i||_2 \cdot ||\mathbf{m}_k||_2} \tag{2}$$

in which $\mathbf{m}_k \in \mathbb{R}^d$ is the vector representation of k^{th} prototype, $\mathbf{h}_i \in \mathbb{R}^d$ is another vector representation of input elements, e.g., item vector. Note that i^{th} input element has two vector representations, $\mathbf{t}_i$ and $\mathbf{h}_i$. A temperature τ is added when calculating cosine similarity between $\mathbf{h}_i$ and $\mathbf{m}_k$ to obtain more skewed distribution.

With estimated $\mathbf{a}_u^k$ and $\mathbf{b}_u^k$ at hand, parameters of Gaussian variational distribution $q(\mathbf{z}_u^k|\mathbf{x}_u, \mathbf{C})$ for k^{th} prototype is computed as

$$\boldsymbol{\mu}_u^k = \frac{\mathbf{a}_u^k}{||\mathbf{a}_u^k||_2}, \quad \boldsymbol{\sigma}^k \leftarrow \sigma_0 \cdot exp(-\frac{1}{2}\mathbf{b}_u^k) \tag{3}$$

Then the k^{th} prototype representation $\mathbf{z}_u^k$ is drawn from multivarite normal distribution with diagonal variance $\mathcal{N}(\boldsymbol{\mu}_u^k, [diag(\boldsymbol{\sigma}_u^k)]^2)$. σ_0 is a hyper-parameter.

Decoder. Given the K factors $\mathbf{z}_u = [\mathbf{z}_u^1, \mathbf{z}_u^2, ..., \mathbf{z}_u^K]$ and the prototype assignment for $\mathbf{C} = \{\mathbf{c}_i\}_{i=1}^N$, MacridVAE predicts the output distribution such that

$$p(\mathbf{x}_{u,i}|\mathbf{z}_u, \mathbf{C}) = \frac{\sum_{k=1}^K \mathbf{c}_{i,k} \cdot f(\mathbf{z}_u^k)}{\sum_{i=1}^N \sum_{k=1}^K \mathbf{c}_{i,k} \cdot f(\mathbf{z}_u^k)} \tag{4}$$

in which

$$f(\mathbf{z}_u^k) = exp(\frac{(\mathbf{z}_u^k)^T \cdot \mathbf{h}_i}{\tau \cdot ||\mathbf{z}_u^k||_2 \cdot ||\mathbf{h}_i||_2}) \tag{5}$$

and $p(\mathbf{x}_{u,i})$ follows a categorical distribution over N elements.

Learning Objective. MacridVAE follows β-VAE [11] to maximize the following objective

$$\mathbb{E}_{p(\mathbf{C})}[\mathbb{E}_{q(\mathbf{z}_u|\mathbf{x}_u,\mathbf{C})}[ln(p(\mathbf{x}_u|\mathbf{z}_u, \mathbf{C}))] - \beta \cdot D_{KL}(q(\mathbf{z}_u|\mathbf{x}_u, \mathbf{C})||p(\mathbf{z}_u))] \tag{6}$$

$p(\mathbf{C})$ is estimated using Equation 2. $q(\mathbf{z}_u|\mathbf{x}_u, \mathbf{C}) = \Pi_{k=1}^{K} q(\mathbf{z}_u^k|\mathbf{x}_u, \mathbf{C})$, where each $q(\mathbf{z}_u^k|\mathbf{x}_u, \mathbf{C})$ is Gaussian distribution with parameters as described in Equation 3. ln is logarithm operation with natural base while Equation 4 describes how to obtain $p(\mathbf{x}_u|\mathbf{z}_u, \mathbf{C})$. Finally, $D_{KL}(\cdot, \cdot)$ is the Kullback-Leibler (KL) divergence that encourages variational distribution $q(\mathbf{z}_u|\mathbf{x}_u, \mathbf{C})$ to be close to prior distribution $p(\mathbf{z}_u) = \mathcal{N}(0, \sigma_0^2\mathbf{I})$. The prior distribution is chosen so as to encourage independence between dimensions. β is used to strengthen disentanglement [11]. We study more about β in Section 4.4.

Our Dual Formulation Let $\mathbf{x}^r \in \{0, 1\}^N$, N is the number of items, as the binary rating of vector of user u. Given $\mathbf{x}^r$ as input of a MacridVAE network named CF-MacridVAE, we obtain $\hat{\mathbf{x}}^r$ as the predicted (or reconstructed) rating vector at the output of CF-MacridVAE as well as K factors $\{\mathbf{z}_k^r\}_{k=1}^K$ learned from rating signals. CF-MacridVAE network parameters, denoted as $\boldsymbol{\theta}_{CF}$, are trained by *minimizing* the negative version of objective as described in Equation 6. We denote this objective as $\mathcal{L}_{CF}$.

Analogously, we have $\mathbf{x}^t \in \mathbb{R}^V$ as the $tf - idf$ vector representation of user u's associated text[2] with V is the number of words in vocabulary. By providing $\mathbf{x}^t$ as input to a second MaridVAE network called Text-MacridVAE, we have reconstructed (predicted) textual content $\hat{\mathbf{x}}^t$ at the output and K disentangled factors $\{\mathbf{z}_k^t\}_{k=1}^K$ learned from textual signals. Let $\boldsymbol{\theta}_{text}$ as the set of parameters of Text-MacridVAE network. They are also trained by minimizing negative version of MacridVAE network in Equation 6. We denote this objective as $\mathcal{L}_{text}$.

We remove user index u from each factor $\mathbf{z}_k^r$ and $\mathbf{z}_k^t$ for clarity. Without any specification, disentangled factors concern one user u. The following sections describe two AddVAE variants corresponding to how we regularize disentangled factors $\{\mathbf{z}_k^r\}_{k=1}^K$ and $\{\mathbf{z}_k^t\}_{k=1}^K$.

3.2 Euclidean Distance-Based Regularization

An effective combination of rating hidden factors and textual hidden factors allows knowledge transfer between the two modalities. On one hand, incorporating textual signal into rating hidden factors resolves user-item interactions sparsity. On the other hand, user preferences underlying rating hidden factors is projected onto textual space by means of reconstructing input text, allowing us to understand preferences in granular level through predicted words per disentangled factor.

The first approach is to minimize Euclidean distance between rating-based representation and text-based representation as in some content-aware non-disentangled models [15, 20, 43]. Although this approach is proposed for a single vector representation, we can extend it to the set of disentangled factors by adding the following regularization.

$$\mathcal{L}_{reg} = \sum_{k=1}^{K} ||\mathbf{z}_k^r - \mathbf{z}_k^t||_2^2 \tag{7}$$

We name this model variant with Euclidean distance regularization $AddVAE_{dist}$.

[2]User-associated text is the concatenation of user's adopted items' texts.

3.3 Aligned Mutual Information Maximization-Based Regularization

Although Euclidean distance-based regularization works in certain cases, we seek a more effective regularization. Since we do not have access to prior information about the alignment between factors, we propose to learn an attention-based alignment step before applying regularization.

Attention-Based Factors Alignment The objective of the alignment step is that given a factor $\mathbf{z}_k^r$, we would like to find a corresponding representation from the set of K textual-based factors $\{\mathbf{z}_k^t\}_{k=1}^K$. We propose to model the alignment by means of attention mechanism [38]. Given rating-based factor $\mathbf{z}_k^r$, the aligned representation from textual factors is

$$\mathbf{z}_k^{t|r} = \sum_{j=1}^{K} \mathbf{A}_{kj}^{t|r} \mathbf{z}_j^t \tag{8}$$

in which, $\mathbf{A}_{kj}^{t|r}$ is the attentive score

$$\mathbf{A}_{kj}^{t|r} = \frac{exp(\frac{g(\mathbf{z}_k^r, \mathbf{z}_j^t)}{\sqrt{d}})}{\sum_{m=1}^{K} exp(\frac{g(\mathbf{z}_k^r, \mathbf{z}_m^t)}{\sqrt{d}})} \tag{9}$$

$g(\cdot, \cdot)$ is implemented as inner product in this paper. The notation $t|r$ means given rating (r) factor, the aligned output is from textual content (t). $\mathbf{z}_k^{t|r}$ is the textual aligned factor conditioning on rating-based factor $\mathbf{z}_k^r$. By repeating Equation 8 for each $\mathbf{z}_k^r \in \{\mathbf{z}_k^r\}_{k=1}^K$, we obtain K textual aligned factors $\{\mathbf{z}_k^{t|r}\}_{k=1}^K$. Similarly, we also obtain K rating aligned factors $\{\mathbf{z}_k^{r|t}\}_{k=1}^K$ in an identical manner.

Each aligned factor $\mathbf{z}_k^{t|r}$ ($\mathbf{z}_k^{r|t}$) is a weighted sum of disentangled factors $\{\mathbf{z}_k^t\}_{k=1}^K$ ($\{\mathbf{z}_k^r\}_{k=1}^K$). Therefore, the aligned factor contains information of all disentangled factors, which is in contrast with the intention of disentangled factors [26], such that each disentangled factor is encouraged to be independent from others. Therefore, the non-negative attentive weight in Equation 9 may not be as effective. To fill this gap, we propose to incorporate Compositional De-Attention [36], abbreviated as CoDA. Unlike vanilla attention in Equation 9 that only allows *addition*, CoDA allows *addition, subtraction* and *nullifying* a certain vector. We hypothesize that CoDA is helpful to learn more effectively aligned factors.

CoDA starts by calculating two matrices $\mathbf{N} \in \mathbb{R}^{K \times K}$ and $\mathbf{E} \in \mathbb{R}^{K \times K}$ [3].

$$\mathbf{N}_{kj} = (\mathbf{z}_k^r)^T \mathbf{z}_j^t \qquad \mathbf{E}_{kj} = -||\mathbf{z}_k^r - \mathbf{z}_j^t||_1 \tag{10}$$

$\mathbf{N}_{kj}$ is the similarity between two disentangled factors $\mathbf{z}_k^r$ and $\mathbf{z}_j^t$ while $\mathbf{E}_{kj}$ is the dissimilarity between $\mathbf{z}_k^r$ and $\mathbf{z}_j^t$ computed via L_1-norm. Then attentive score matrix $\mathbf{A} \in \mathbb{R}^{K \times K}$ is calculated as

$$\mathbf{A} = tanh(\mathbf{N}) \odot sigmoid(\mathbf{E}) \tag{11}$$

Since all element values in $\mathbf{E}$ is negative, applying $sigmoid$ constraints the value in to $[0, 0.5]$. To guarantee $sigmoid(\mathbf{E})$ is between $[0, 1]$, we apply transformation $\mathbf{E} \leftarrow \mathbf{E} - mean(\mathbf{E})$ before applying $sigmoid$. Another transformation is to multiply $sigmoid(\mathbf{E})$ by 2.

[3]We set scaling factor $\alpha = 1$ and $\beta = 1$ and omit them for simplicity. Details about α and β can be found in [36]

Unlike vanilla attention in Equation 9 whose values are non-negative, the attentive score in Equation 11 can be both positive and negative. $\mathbf{A}_{kj}$ is biased towards 1 iff $\mathbf{N}_{kj}$ is biased towards 1 and $sigmoid(\mathbf{E})$ is biased towards 1 as well. Conversely, $\mathbf{A}_{kj}$ can be negative, e.g., close to -1 in case one of $\mathbf{N}_{kj}$ and $sigmoid(\mathbf{E})$ goes towards 1 and the other goes towards -1. Furthermore, when $sigmoid(\mathbf{E}_{kj})$ is around 0, it acts as a gate that learns to forget information from a vector. By equipping CoDA, which is more flexible and adaptive than vanilla attention, we end up with more effective factor alignments since CoDA learns to *add*, *substract* or *forget* the disentangled factors. All in all, with CoDA mechanism in Equation 11, we compute aligned factors as follows:

$$\mathbf{z}_k^{t|r} = \sum_{j=1}^{K} \mathbf{A}_{kj}^{t|r} \mathbf{z}_j^t, \ \forall k = 1, 2, ..., K; \ \mathbf{z}_k^{r|t} = \sum_{j=1}^{K} \mathbf{A}_{kj}^{r|t} \mathbf{z}_j^r, \ \forall k = 1, 2, ..., K \tag{12}$$

Mutual Information (MI) Maximization. With aligned factors as computed in Equation 12 at hand, we design a regularization so that their underlying information are driven to be close.

In some cases, the magnitudes of text-based and rating-based representation are widely different, therefore distance-based regularization as in Equation 7 is less effective. Hence, we explore alternative regularization called MI maximization [12].

Given a user $u \in \mathcal{U}$, let $I(\mathbf{z}_k^{r|t}, \mathbf{z}_k^{t|r})$ be the mutual information between two factors $\mathbf{z}_k^{r|t}$ and $\mathbf{z}_k^{t|r}$. Since maximizing mutual information (MI) is known to be intractable, we aim at maximizing the lower bound of MI. Therefore, we adopt Jensen-Shannon MI estimator [12, 33] as our maximization objective

$$I(\mathbf{z}_k^{r|t}, \mathbf{z}_k^{t|r}) = \mathbb{E}_{u \in \mathcal{U}}[-sp(-g(\mathbf{z}_k^{r|t}, \mathbf{z}_k^{t|r}))] - \mathbb{E}_{u \in \mathcal{U}, j \neq k}[sp(g(\mathbf{z}_k^{r|t}, \mathbf{z}_j^{t|r}))] \tag{13}$$

$g(\cdot, \cdot)$ is inner product and $sp(x) = log(1+e^x)$ is softplus function. The distributions of user u and factor $j \neq k$ are uniform distribution. Finally, the mutual information regularization term is summed up over negative mutual information for minimization as follows.

$$\mathcal{L}_{reg} = -\sum_{k=1}^{K} I(\mathbf{z}_k^{r|t}, \mathbf{z}_k^{t|r}) \tag{14}$$

Similar to KDR [29], the final disentangled factors for user u are derived by summing up aligned factors. These are,

$$\mathbf{z}_k = \mathbf{z}_k^{r|t} + \mathbf{z}_k^{t|r}, \ \forall k = 1, 2, ..., K \tag{15}$$

In this way, factors from two signals, rating and text, complement each other. Then $\mathbf{z}_k$ will be shared in both CF-MacridVAE and Text-MacridVAE to predict output in Equation 4. Each factor $\mathbf{z}_k$ is trained by both rating and textual supervision signals. We name our model variant with attention-based alignment and MI maximization regularization $AddVAE_{MI}$.

3.4 Learning Objective

Given the learning objective of CF-MacridVAE and Text-MacridVAE networks in Section 3.1 and the regularization term, our model parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}_{CF}, \boldsymbol{\theta}_{text}\}$ are jointly learned to minimize the following objective:

$$\mathcal{L} = \mathcal{L}_{CF} + \lambda_{text}\mathcal{L}_{text} + \lambda_{reg}\mathcal{L}_{reg} \tag{16}$$

Table 2: Statistics of our chosen datasets

Data	#users	#items	#interactions	#words
Citeulike-a	5,551	16,980	204,986	8,000
Movies	45,498	27,320	576,901	10,000
Kindle	29,920	31,778	388,143	4,401
Cell Phone	4,775	4,883	31,749	4,796

λ_{text} and λ_{reg} are hyper-parameters, controlling the influence of textual signals and regularization term. $\mathcal{L}_{reg}$ can be either based on distance regularization in Equation 7 (variant $AddVAE_{dist}$) or alignment and MI maximization in Equation 14 (variant $AddVAE_{MI}$).

4 EXPERIMENTS

4.1 Settings

As our goal is to incorporate textual content into recommendation model in disentangled fashion, we evaluate our method against baselines on recommendation quality, i.e., top-N recommendation task of predicting top N items user is likely to interact with.

Datasets. We conduct a series of experiments on real-world publicly available datasets.

- **Movies, Kindle, Cell Phone** are three categories from Amazon datasets [4], consisting of product reviews data and products meta-data. We extract user-item interactions as a user wrote review for a product. For textual content, we concatenate product titles and product descriptions adopted by a user. We only keep user with at least 5 interactions so that we have interactions for training, validation and testing.
- **Citeulike-a**[5] has user's articles. A user-item interaction is when user saves an article in their collection. Textual content is the concatenation of title and abstract of an article.

Data Preprocessing. In each dataset, we leverage ratio split strategy as suggested by [51], splitting the interactions of a user into training, validation and test set with ratio 0.8:0.1:0.1, respectively. For Amazon dataset with provided timestamp per interaction, we first sort user's interactions chronologically before splitting. For Citeulike-a dataset, there is no timestamp per interaction and therefore, items for validation and test set are chosen randomly. We tokenize item textual content using *SpaCy* [13], remove stop words and only keep words appears at least 5 times and in less than 60% of item's texts. All user-item interactions without textual content or textual content without vocabulary words are excluded. The statistics of data after preprocessing is presented in Table 2.

Competitors. We compare our proposed model against disentangled recommendation model (MacridVAE, DGCF) and textual content-aware recommendation model (CDL, CVAE, GATE).

- **MacridVAE** [26] β-VAE based model learns to generate user's interaction assuming there are several disentangled factors underlying user's adoptions.
- **DGCF** [44] learns a disentangled graph-based collaborative filtering model. Each user-item interaction is modeled as a distribution over a set of intents and iteratively refine intent representation using graph propagation.

[4] https://jmcauley.ucsd.edu/data/amazon/
[5] http://wanghao.in/CDL.htm

- **GATE** [24] adopts attention mechanism to model textual content then incorporate into autoencoder-based recommendation model. Item representation is also enriched by learning to attend over the set of neighbors.
- **CDL** [43] adopts Stacked Denoising Autoencoder [40] to model textual content and incorporate into recommendation model under a probabilistic framework.
- **CVAE** [20] has a similar idea as CDL but adopts VAE [17] to model textual content.

As we incorporate textual content into user side in our proposed model so as to interpret user preferences, in CDL, CVAE and GATE, textual content is also incorporated on user side[6].

Hyperparameters Settings. All models are trained using Adam optimization [16] with learning rate 0.001, batch size 512 and embedding size is set to 64. The maximum number of training epoch is 200. Training stop after e epochs without improving $Recall@20$ on validation set, in which e is 15 for CDL, CVAE and GATE, 5 for other models. Hyper-parameters are chosen based on model performance on validation set.

For CDL and CVAE, we search λ_u and λ_v in $\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}\}$. We use 2-layer CDL (L = 4) with ReLU activation and hidden unit is 200 for Cell Phone and Citeulike-a, 300 for other datasets. For CVAE, 2-layer network of size $200 - 100$ is used inference and generation network with *sigmoid* activation.

For GATE, ρ is set to 100 for all datasets. Attention dimension d_a is searched from $\{10, 15, 20, 25\}$. Threshold to select neighbor for users is tuned from $\{0.1, 0.15, 0.2, 0.3\}$. The sequence length of user's text is chosen in from $\{500, 1000, 1500, 2000\}$. Hidden layer size is set to 200 for Cell Phone and Citeulike-a, 300 for other datasets.

For DGCF, $K = 4$, $T = 2$, $L = 1$, weight of independent term is searched in $\{0.005, 0.01, 0.02, 0.05, 0.1, 0.2\}$.

For MacridVAE, we set to K to 4. The dimensions of hidden layer is set to 200 for Cell Phone and Citeulike-a, 300 for other dataset. Dropout rate is 0.5, $\tau = 0.1$, $\sigma_0 = 0.075$. The value of β follows an annealing update, i.e., $\beta = min(\beta_0, \frac{update}{T})$, in which $update$ is the number of parameters update so far, T is the estimated number of parameters update. We empirically found $\beta_0 = 0.2, T = 20000$ works well for recommendation task.

For AddVAE, we set parameters for CF-MacridVAE and Text-MacridVAE identically to MacridVAE, except $\sigma_0 = 0.05$ for Kindle and Cell Phone. We found that multiplying $sigmoid(E)$ by 2 works well for Cell Phone, for other datasets we apply $\mathbf{E} \leftarrow \mathbf{E} - mean(\mathbf{E})$ in Equation 11. λ_{reg} is set to 0.2 for all dataset. λ_{text} is set to 0.2 for Citelike-a, Movies and Kindle and 1.0 for Cell Phone.

All models are trained 10 times with different random seeds on GPU machine of RTX 8000. The averaged numbers over 10 runs on test set are reported.

Evaluation Metrics. We evaluate the recommendation performance on two ranking metrics, namedly Recall at top K - Recall@K and Normalized Discounted Cummulative Gain at top K - NDCG@K. Let $\mathcal{P}_{u,K}$ is the set of top K items predicted for user u, $\mathcal{G}_u$ is the set of items in the test set of user u.

$$Recall@K = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{|\mathcal{P}_{u,K} \cap \mathcal{G}_u|}{|\mathcal{G}_u|} \tag{17}$$

$$NDCG@K = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{DCG@K}{IDCG@K} \quad DCG@K = \sum_{r=1}^{K} \frac{2^{rel_r} - 1}{log_2(r+1)} \tag{18}$$

$rel_r = 1$ if r^{th} item in $\mathcal{P}_{u,K}$ is in $\mathcal{G}_u$, 0 otherwise. IDCG@K is the computed identically with DCG@K by sorting test set and treating it as the set of predicted items.

4.2 Quantitative Evaluation

The performances of our model variants, $AddVAE_{dist}$, $AddVAE_{MI}$, and baselines are illustrated in Table 3. We observe that

- Our model variants perform well in all datasets, achieving better results than all competitors in both top-10 and top-50 metrics.
- $AddVAE_{MI}$ equipping MI maximization regularization achieves the highest results in 3 out of 4 datasets while $AddVAE_{dist}$ stands on top in Kindle dataset.
- For Cell Phone and Citeulike-a datasets, a large improvement is observed over the disentangled baselines MacridVAE and DGCF. The largest gap w.r.t. MacridVAE is almost 2% and 1% in HR@50, respectively. On the larger and sparser dataset Movies, the improvement is smaller, e.g., less than 0.8%.
- For Kindle dataset, $AddVAE_{dist}$ surpasses $AddVAE_{MI}$ and is slightly better than the best-performing baseline MacridVAE. Notably, $AddVAE_{MI}$ is worse than MacridVAE, showing that Euclidean distance-based regularization is favorable in this dataset.
- Among baseline models, disentangled models including MacridVAE and DGCF outperform non-disentangled ones, except Recall@50 on Cell Phone. This ascertains the advantage of learning disentangled representation in recommendation task. Between these two models, MacridVAE shows their stronger performance than DGCF in 3 out of 4 datasets while DGCF is only better than MacridVAE on the smallest dataset, Cell Phone.
- All in all, the experimental results show that our proposed incorporation of textual content into disentangled recommendation model is beneficial to improve the recommendation quality.

4.3 Hyperparameters and Architecture Analysis

We study the impact of different parameters that play a role in the proposed models. Additionally, we also investigate model performance on various architectural choices.

Number of Factors. Table 4 presents model performance w.r.t the number of factors. *Firstly*, $K = 2$ performs worst in all datasets, indicating that too few factors are insufficient to capture user preferences at granular level. *Secondly*, a significant degradation is observed in Movies and Citeulike-a when K becomes large, e.g., $K = 12$. In these datasets, user preferences may be more coarse-grained. *Thirdly*, for Kindle and Cell Phone dataset, setting $K = 12$ slightly degrades model performance compared to $K = 8$ or $K = 4$. *Overall*, the number of user factors is data-dependent and should be chosen carefully for disentangled recommendation models. Empirically, $K = 4$ is a reasonable choice for our chosen datasets.

Architecture Analysis We report our model performance with various architectural choices in Table 5. Pertaining to 3 datasets, namedly Movies, Cell Phone and Citeulike-a, where $AddVAE_{MI}$ achieves top performance, we make the following observations.

[6]In their original version, textual content is incorporated on item side

Table 3: Comparison between models on top-10 and top-50 recommendation task with two metrics Recall (R) and NDCG (N). Bold numbers are the best while underlined ones are second best. All numbers are percentage, we omitted % for clarity. § denotes statistical significant number (p-value < 0.05) between bold numbers and underlined ones.

	Movies				Kindle				Cell Phone				Citeulike-a			
	Metric@10		Metric@50		Metric@10		Metric@50		Metric@10		Metric@50		Metric@10		Metric@50	
	R	N	R	N	R	N	R	N	R	N	R	N	R	N	R	N
CDL	4.59	2.35	12.65	4.14	5.61	2.92	13.81	4.76	6.04	3.20	14.11	4.95	15.47	10.17	36.06	15.88
CVAE	4.61	2.37	12.72	4.17	5.49	2.84	14.06	4.77	6.28	3.39	15.43	5.36	15.34	9.99	36.29	15.81
GATE	3.88	1.99	10.87	3.55	5.44	2.91	12.50	4.50	5.00	2.70	12.60	4.35	15.98	10.18	34.42	15.45
DGCF	4.40	2.35	11.72	3.97	6.46	3.57	14.7	5.42	6.49	**3.54**	15.23	5.43	17.61	12.59	37.99	18.05
MacridVAE	6.29	3.39	15.08	5.36	10.11	5.58	20.08	7.85	5.69	2.99	13.67	4.72	22.41	15.52	42.97	21.38
$AddVAE_{MI}$	**6.41**§	**3.43**§	**15.83**§	**5.54**§	9.51	5.19	19.51	7.47	**6.80**§	**3.54**	**17.34**§	**5.82**§	**23.10**§	**16.06**§	**43.98**§	**21.97**§
$AddVAE_{dist}$	6.29	3.38	15.16	5.37	**10.28**§	**5.64**§	**20.22**	**7.91**	5.74	3.01	13.82	4.77	22.45	15.60	43.17	21.51

Table 4: Recall@10 (R) and NDCG@10 (N) with various K.

	Movies		Kindle		Cell Phone		Citeulike-a	
	R	N	R	N	R	N	R	N
K = 2	5.30	2.78	9.66	5.28	6.56	3.40	18.72	12.77
K = 4	**6.41**	**3.43**	10.28	5.64	6.80	3.54	**23.10**	**16.06**
K = 8	4.83	2.61	**10.45**	5.78	**6.91**	**3.54**	20.43	14.15
K = 12	3.68	1.98	10.41	**5.79**	6.64	3.42	17.03	11.58

Table 5: Recall@10 (R) and NDCG@10 (N) with architecture variants. (1) $AddVAE_{MI}$. (2) $AddVAE_{dist}$. (3) $AddVAE_{MI}$ without CoDA. (4) $AddVAE_{MI}$ without alignment. (5) Replace MI maximization in $AddVAE_{MI}$ by Euclidean distance regularization.

	Movies		Kindle		Cell Phone		Citeulike-a	
	R	N	R	N	R	N	R	N
(1)	**6.41**	**3.43**	9.51	5.19	**6.80**	**3.54**	**23.10**	**16.06**
(2)	6.29	3.38	**10.28**	**5.64**	5.74	3.01	22.45	15.60
(3)	5.27	2.86	9.01	4.99	5.27	2.65	20.97	14.54
(4)	5.08	2.75	8.56	4.72	5.39	2.76	20.80	14.31
(5)	6.39	3.42	9.44	5.15	6.77	3.52	22.97	15.97

- (1) vs. (3): When replacing CoDA (Equation 11) by vanilla attention (Equation 9), a sharp degradation is witnessed, indicating that CoDA is significant in learning effective combination of disentangled factors.
- (1) vs. (4): We remove alignment step as in Equation 12, then equation 15 becomes $\mathbf{z}_k = \mathbf{z}_k^r + \mathbf{z}_k^t$, $\forall k = 1, 2, ..., K$. Numbers in (1) are much higher than their counterparts in (4), showcasing that the proposed attention-based alignment significantly boosts model performance.
- (1) vs. (5): In (5), we replace MI maximization regularization (Equation 14) by Euclidean distance regularization (Equation 7) and leave the alignment unchanged. It is observed that MI-based regularization works better than Euclidean distance-based regularization with bigger gaps in Citeulike-a dataset while smaller gaps are witnessed in Cell Phone and Movies datasets.

For Kindle dataset, we also observe that CoDA and attention-based alignment are helpful by contrasting (1) with (3) and (4), although worse than (2). Numbers in (1) and (5) show that MI maximization is better than Euclidean distance-based regularization.

Table 6: Recall@10 and NDCG@10 with various λ_{reg}.

λ_{reg}	Movies		Kindle		Cell Phone		Citeulike-a	
	R	N	R	N	R	N	R	N
0.0	6.42	3.43	10.21	5.62	6.72	3.51	23.14	16.07
0.2	6.41	3.43	**10.28**	**5.64**	6.80	3.54	23.10	16.06
0.4	6.41	3.43	10.23	5.64	6.78	3.54	23.09	16.05
0.6	6.41	3.43	10.22	5.63	6.79	3.54	**23.19**	16.10
0.8	6.42	3.43	10.21	5.61	**6.81**	**3.55**	23.16	16.09
1.0	**6.43**	**3.43**	10.26	5.64	6.75	3.52	23.17	**16.10**

Table 7: Recall@10 and NDCG@10 with various λ_{text}.

λ_{text}	Movies		Kindle		Cell Phone		Citeulike-a	
	R	N	R	N	R	N	R	N
0.0	6.37	3.41	10.20	5.62	6.54	3.40	22.98	16.04
0.2	**6.41**	**3.43**	**10.28**	5.64	6.63	3.45	23.10	**16.06**
0.4	6.36	3.40	10.24	5.63	6.31	3.28	23.05	16.05
0.6	6.32	3.37	10.26	5.64	6.25	3.25	23.03	16.01
0.8	6.25	3.33	10.23	5.62	6.57	3.45	22.98	16.03
1.0	6.23	3.31	10.26	**5.65**	**6.80**	**3.54**	**23.11**	16.05

In summary, we found that aligning disentangled factors using attention mechanism works more effectively than using regularization methods relying solely on Euclidean distance or MI maximization. Furthermore, incorporating more expressive attention mechanism such as compositional de-attention (CoDA) is key to improving representation power of aligned factors.

Studies of λ_{reg} and λ_{text}. Finally, we vary the values of λ_{reg} and λ_{text} and the model performance is reported in Tables 6 and 7, respectively. Overall, we found that these two hyper-parameters do not have significant impact. We conjecture that this is due to the representation in alignment step may compensate the model performance when λ_{text} or λ_{reg} changes. These parameters seems to be data-dependent and should be carefully selected when training.

4.4 Interpretability Analysis

Besides recommendation performance, incorporating textual content into disentangled model aims at providing interpretability of user preferences. To understand the user's preferences underlying each factor, we visualize the top predicted words by each factor as

(a) Word clouds of user with id 1544

(b) Word clouds of user with id 1396

(c) Word clouds of user with id 5542

Figure 2: On Citeulike-a dataset with $K = 4$, items that user interacts with are scientific articles. Each user has four word clouds representing their topic of interest corresponding to four factors. Size of word encodes their predicted score given by AddVAE. Best viewed in color.

a word cloud. The size of words in each word cloud encodes their predicted score by our model. Intuitively, the bigger the size of a word, the more likely that user is interested in the aspect described by this word. Word color and position are chosen randomly. An illustration of words of interest of three users corresponding to four hidden factors on Citeulike-a dataset is presented in Figure 2.

Interpretability We can understand user's preferences by perusing their four word clouds (corresponding to hidden factors). For example, user with id 1544 with their word clouds in Figure 2 is interested in User Interface (UI), designing, user-centered problem and support. Similarly, user with id 1396 cares more about graph and network related problem such as community, page rank algorithm, internet, etc. Finally, latent models, mixture models, probabilistic algorithm are likely the topics user with id 5542 is interested in.

Interpretability vs. Recommendation Performance We analyze the influence of β on the user's preferences interpretation. As noted in [11], setting large β leads to more disentangled representation. However, larger β is unfavorable in the context of recommendation[21] as higher β results in generating more similar user interaction vector as in training data. In contrast, in recommendation task we care more about predicting unobserved user-item interactions which are likely to happen. Recall that $\beta = min(\beta_0, \frac{update}{T})$, $update$ is the number of parameters update. We fix $\beta_0 = 0.2$ and vary T in $\{1000, 1500, 3000, 5000, 10000, 20000\}$. The word clouds are presented in Figure 3. When T is smaller, i.e. 1000 and 1500, despite lower Recall@10, the word clouds corresponding to factors are more discriminative, i.e., less words are shared between word clouds. In contrast, when T becomes larger, meaning that β becomes smaller, more words are shared between word clouds. This is problematic when one would like to apply VAE on recommendation [21]. Therefore, β should be chosen to balance between interpretability and recommendation performance. For instance, in this example, $T = 5000$ achieves both good recommendation performance and discriminative clouds.

Interpretability vs. Textual Supervision Signal. We would like to emphasize the role of textual content on interpretability. As presented in Table 7, we observe that setting $\lambda_{text} = 0$ reduces model performance. We investigate this case by visualizing word clouds when $\lambda_{text} = 0$ in Figure 4, contrasting it with $\lambda_{text} = 0.2$.

(a) T = 1000, Recall@10 = 22.35

(b) T = 1500, Recall@10 = 22.65

(c) T = 3000, Recall@10 = 23.09

(d) T = 5000, Recall@10 = 23.03

(e) T = 10000, Recall@10 = 23.13

(f) T = 20000, Recall@10 = 23.10

Figure 3: Word clouds for user with id 562 on Citeulike-a dataset with various values of T. Recall@10 is reported with % omitted. Best viewed in color.

(a) $\lambda_{text} = 0$

(b) $\lambda_{text} = 0.2$

Figure 4: Word clouds when $\lambda_{text} = 0$, i.e. there is no textual supervision signals and $\lambda_{text} = 0.2$. Best viewed in color.

With $\lambda_{text} = 0$, the word clouds of the four factors are very similar and unlinked to user's preferences. This again ascertains the importance of textual content on interpretation of user's preferences.

5 CONCLUSION

We look into learning disentangled representation for user-oriented content-aware recommendation. Our proposed model, called AddVAE, achieves better recommendation accuracy compared to a series of disentangled baselines as well as content-aware recommendation model. We propose aligning disentangled factors learned from ratings and textual content based on regularization and compositional de-attention mechanism. The experimental analysis shows that this is effective and plays a key role. An interesting follow-up research direction is to explore mechanism that models disentangled user preferences at both intra- and inter-user levels.

REFERENCES

[1] Philip Bachman, R Devon Hjelm, and William Buchwalter. 2019. Learning Representations by Maximizing Mutual Information across Views. In *NeurIPS*. 15509–15519.

[2] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. 2018. Mutual Information Neural Estimation. In *ICML*, Vol. 80. 531–540.

[3] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* 35 (2013), 1798–1828.

[4] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[5] Diane Bouchacourt, Ryota Tomioka, and Sebastian Nowozin. 2018. Multi-Level Variational Autoencoder: Learning Disentangled Representations From Grouped Observations. *Proceedings of the AAAI Conference on Artificial Intelligence* (2018).

[6] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Controllable Multi-Interest Framework for Recommendation. In *Proceedings of the 26th ACM SIGKDD*. 2942–2951.

[7] Ricky T. Q. 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*, Vol. 31.

[8] Wanyun Cui, Guangyu Zheng, and Wei Wang. 2020. Unsupervised Natural Language Inference via Decoupled Multimodal Contrastive Learning. In *EMNLP*. 5511–5520.

[9] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *SIGIR*. 639–648.

[10] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web*. 173–182.

[11] Irina Higgins, Loïc 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. In *5th International Conference on Learning Representations*.

[12] R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Philip Bachman, Adam Trischler, and Yoshua Bengio. 2019. Learning deep representations by mutual information estimation and maximization. In *7th International Conference on Learning Representations*.

[13] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. *spaCy: Industrial-strength Natural Language Processing in Python.* https://doi.org/10.5281/zenodo.1212303

[14] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax. In *5th International Conference on Learning Representations*.

[15] Dong Hyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu. 2016. Convolutional Matrix Factorization for Document Context-Aware Recommendation. In *RecSys*. 233–240.

[16] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

[17] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations*.

[18] Lingpeng Kong, Cyprien de Masson d'Autume, Lei Yu, Wang Ling, Zihang Dai, and Dani Yogatama. 2020. A Mutual Information Maximization Perspective of Language Representation Learning. In *8th ICLR*.

[19] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37.

[20] Xiaopeng Li and James She. 2017. Collaborative Variational Autoencoder for Recommender Systems. In *KDD*. 305–314.

[21] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *Proceedings of the 2018 World Wide Web Conference*. 689–698.

[22] Huafeng Liu, Liping Jing, Jingxuan Wen, Pengyu Xu, Jiaqi Wang, Jian Yu, and Michael K. Ng. 2021. Interpretable Deep Generative Recommendation Models. *Journal of Machine Learning Research* 22 (2021), 1–54.

[23] 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 *Proceedings of the 36th International Conference on Machine Learning*. 4114–4124.

[24] Chen Ma, Peng Kang, Bin Wu, Qinglong Wang, and Xue Liu. 2019. Gated Attentive-Autoencoder for Content-Aware Recommendation. In *WSDM*. 519–527.

[25] Jianxin Ma, Peng Cui, Kun Kuang, Xin Wang, and Wenwu Zhu. 2019. Disentangled Graph Convolutional Networks. In *ICML*. 4212–4221.

[26] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learning Disentangled Representations for Recommendation. In *NeurIPS*, Vol. 32.

[27] Jianxin Ma, Chang Zhou, Hongxia Yang, Peng Cui, Xin Wang, and Wenwu Zhu. 2020. Disentangled Self-Supervision in Sequential Recommenders. In *KDD*. 483–491.

[28] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. In *ICLR*.

[29] Shanlei Mu, Yaliang Li, Wayne Xin Zhao, Siqing Li, and Ji-Rong Wen. 2021. Knowledge-Guided Disentangled Representation Learning for Recommender Systems. *ACM Trans. Inf. Syst.* (2021).

[30] Preksha Nema, Alexandros Karatzoglou, and Filip Radlinski. 2021. Disentangling Preference Representations for Recommendation Critiquing with ß-VAE. In *CIKM*. 1356–1365.

[31] Thong Nguyen and Luu Anh Tuan. 2021. Contrastive Learning for Neural Topic Model. In *NeurIPS*.

[32] Xia Ning and George Karypis. 2011. SLIM: Sparse Linear Methods for Top-n Recommender Systems. In *IEEE 11th International Conference on Data Mining*.

[33] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. 2016. f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization. In *Advances in Neural Information Processing Systems*, Vol. 29.

[34] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. AutoRec: Autoencoders Meet Collaborative Filtering. In *Proceedings of the 24th International Conference on World Wide Web (WWW '15 Companion)*. 111–112.

[35] Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. 2020. InfoGraph: Unsupervised and Semi-supervised Graph-Level Representation Learning via Mutual Information Maximization. In *ICLR*.

[36] Yi Tay, Anh Tuan Luu, Aston Zhang, Shuohang Wang, and Siu Cheung Hui. 2019. Compositional De-Attention Networks. In *NeurIPS*, Vol. 32.

[37] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation Learning with Contrastive Predictive Coding. abs/1807.03748 (2018). https://arxiv.org/abs/1807.03748

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, Vol. 30.

[39] Petar Velickovic, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. 2019. Deep Graph Infomax. In *ICLR*.

[40] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research* (2010), 3371–3408.

[41] Chong Wang and David M. Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *KDD*. 448–456.

[42] Hao Wang, Xingjian Shi, and Dit-Yan Yeung. 2016. Collaborative Recurrent Autoencoder: Recommend While Learning to Fill in the Blanks. In *NeurIPS*. 415–423.

[43] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative Deep Learning for Recommender Systems. In *KDD*. 1235–1244.

[44] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled Graph Collaborative Filtering. In *SIGIR*. 1001–1010.

[45] Yifan Wang, Suyao Tang, Yuntong Lei, Weiping Song, Sheng Wang, and Ming Zhang. 2020. DisenHAN: Disentangled Heterogeneous Graph Attention Network for Recommendation. In *Proceedings of the 29th ACM CIKM*. 1605–1614.

[46] Junkang Wu, Wentao Shi, Xuezhi Cao, Jiawei Chen, Wenqiang Lei, Fuzheng Zhang, Wei Wu, and Xiangnan He. 2021. DisenKGAT: Knowledge Graph Embedding with Disentangled Graph Attention Network. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*. 2140–2149.

[47] Longqi Yang, Tobias Schnabel, Paul N. Bennett, and Susan Dumais. 2021. Local Factor Models for Large-Scale Inductive Recommendation. In *Fifteenth ACM Conference on Recommender Systems*. 252–262.

[48] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative Knowledge Base Embedding for Recommender Systems. In *KDD*. 353–362.

[49] Shuai Zhang, Xi Rao, Yi Tay, and Ce Zhang. 2021. Knowledge Router: Learning Disentangled Representations for Knowledge Graphs. In *NAACL-HLT*. 1–10.

[50] Yin Zhang, Ziwei Zhu, Yun He, and James Caverlee. 2020. Content-Collaborative Disentanglement Representation Learning for Enhanced Recommendation. In *Fourteenth ACM Conference on Recommender Systems*. 43–52.

[51] Wayne Xin Zhao, Junhua Chen, Pengfei Wang, Qi Gu, and Ji-Rong Wen. 2020. Revisiting Alternative Experimental Settings for Evaluating Top-N Item Recommendation Algorithms. In *CIKM*. 2329–2332.

[52] 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.

[53] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-Rec: Self-Supervised Learning for Sequential Recommendation with Mutual Information Maximization. In *CIKM*. 1893–1902.