Disentangled Representation Learning for Recommendation

Xin Wang, Member, IEEE, Hong Chen, Yuwei Zhou, Jianxin Ma and Wenwu Zhu, Fellow, IEEE

Abstract—There exist complex interactions among a large number of latent factors behind the decision making processes of different individuals, which drive the various user behavior patterns in recommender systems. These factors hidden in different behaviors demonstrate highly entangled patterns, covering from high-level user intentions to low-level individual preferences. Uncovering the disentanglement of these latent factors can benefit in enhanced robustness, interpretability, and controllability during representation learning for recommendation. However, the large degree of entanglement within latent factors poses great challenges for learning representations that disentangle them, and remains largely unexplored in literature. In this paper, we present the SEMantic MACRo-mi-cro Disentangled Variational Auto-Encoder (SEM-MacridVAE) model for learning disentangled representations from user behaviors, taking item semantic information into account. Our SEM-MacridVAE model achieves macro disentanglement by inferring the high-level concepts associated with user intentions (e.g., to buy a pair of shoes or a laptop) through a prototype routing mechanism, as well as capturing the individual preferences with respect to different concepts separately. The micro disentanglement is guaranteed through a micro-disentanglement regularizer stemming from a information-theoretic interpretation of VAEs, which forces each dimension of the representations to independently reflect an isolated low-level factor (e.g., the size or the color of a shirt). The semantic information including visual and categorical signals extracted from candidate items is utilized to further boost the recommendation performance of the proposed SEM-MacridVAE model. Empirical experiments demonstrate that our proposed approach is able to achieve significant improvement over the state-of-the-art baselines. We also show that the learned representations are interpretable and controllable, capable of potentially leading to a new paradigm for recommendation where users have fine-grained control over some target aspects of the recommendation candidates.

Index Terms—Disentangled Representation, Recommendation.

1 INTRODUCTION

Learning representations that can accurately reflect users’ preference, based chiefly on user behavior, has been an important research focus for recommender systems since the advent of collaborative filtering [57]. Despite the huge success in the past decade, existing user-behavior-based representation learning methods, including the deep structure approaches [10], [20], [38], [40], [65], [79], generally ignore the complex interactions among the latent factors behind the users’ decision-making processes. These latent factors can be highly entangled, ranging from macro concepts that govern the intention of a user in a particular behavior, to micro individual preferences at a granular level when implementing a specific intention. Existing methods fail to disentangle these latent factors, resulting in the fact that those learned representations may mistakenly preserve the confounding of the highly entangled factors, which leads to non-robustness and low interpretability.

Disentangled representation learning, which targets at learning factorized representations capable of uncovering and disentangling the latent explanatory factors hidden in the observed data [3], has recently attracted lots of attentions in the research community. Disentangled representations benefits in more robustness, i.e., less sensitive to the misleading correlations discovered in the limited observed training data. Besides, the enhanced interpretability brought by disentangled representation also finds direct application in recommendation-related tasks, such as transparent advertising [41], customer-relationship management, and explainable recommendation [19], [78] etc. Moreover, the controllability exhibited by many disentangled representations [7], [8], [9], [13], [21], [23] can provide users with explicit controls over their desired recommendation results and offer them more interactive experience, which has great potential in driving a new paradigm for recommendation. However, the existing literature on disentangled representation learning mainly focuses on computer vision [9], [13], [14], [21], [22], [34], [37], [52], [59] rather than recommender systems.

User behaviors in recommender systems can be driven by both macro intentions and micro preferences, where macro intentions may involve high-level user intentions such as purchasing a pair of shoes or a laptop and micro preferences may refer to low-level user preferences such as the size or color of the shoes. Therefore, given that the above discrete relational user behavior data is essentially different from continuous image data, learning disentangled representations based on user behavior data for recommendation becomes largely unexplored and poses two challenges.

• The macro intentions and micro preferences co-exist in user behaviors, requiring the disentangled representation learning to separate these two levels of factors in a way that can preserve the hierarchical
relations between high-level user intentions and low-level individual preferences under the intentions.

- The observed user behavior data, such as user-item ratings and user-item consumptions, are discrete and sparse in essence, differing themselves from continuous image data, which implies that the observed user behavior data is only associated with a very small number of entries in the high-dimensional representation space. This will be especially problematic when exploring the interpretability of a particular isolated dimension through varying the value of that dimension with other dimensions fixed.

To solve the challenges, we propose the SEMantic MACRo-mIcro Disentangled Variational Auto-Encoder (SEM-MacridVAE) model for learning disentangled representations based on user behavior with item semantic information being taken into consideration in this paper. Our proposed method explicitly models the separation of macro and micro factors when performing disentanglement at each level. In particular, macro disentanglement is achieved by discovering the high-level concepts associated with user intentions through a prototype routing mechanism, and separately capturing the individual preferences of a user with respect to different concepts. Micro disentanglement is strengthened through a micro-disentanglement regularizer derived from interpreting VAEs [33], [58] in terms of an information-theoretic perspective, which aims at forcing each individual dimension to indicate an independent micro factor. The semantic information including visual and categorical signals from items is employed to further boost the model performances of the proposed SEM-MacridVAE model. To handle the conflict between sparse discrete user behavior observations and dense continuous latent representations, we propose a beam-search strategy for investigating the interpretability of each isolated dimension through finding a smooth trajectory within different representations.

We conduct extensive empirical experiments to show that our SEM-MacridVAE model can achieve significant improvement over several state-of-the-art baselines. Experimental results also demonstrate that the learned disentangled representations from SEM-MacridVAE can be interpretable and controllable, which may potentially bring a promising new paradigm for recommendation where users are given fine-grained controls over target aspects of the recommendation candidates.

To summarize, this paper makes the following contributions:

- We conduct extensive experiments on several real-world datasets to verify the advantages of our SEM-MacridVAE model in terms of recommendation accuracy, interpretability and controllability.

In particular, we would like to point out that compared to the MacridVAE model [51], our proposed SEM-Macrid model has the following expansions:

1) We propose a new SEM-MacridVAE which incorporates the visual semantic and categorical semantic signals extracted from items to boost the model performance.
2) Besides the public Movielens datasets adopted by MacridVAE, we additionally include four public available Amazon datasets to enrich our experiments.
3) We conduct more extensive experiments, including more recent comparative baselines, comprehensive ablation studies as well as visualizations.

The remainder of this paper is organized as follows. We review related works in Section 2 and present our proposed SEM-MacridVAE model in Section 3. Section 4 describes details about empirical evaluations over several real-world datasets in terms of various metrics. Last but not least, we conclude the whole paper and point out research directions deserving further investigations in Section 5.

2 RELATED WORK

In this section, we review existing works on user behavior representation learning and disentangled representation learning.

2.1 Learning Representations from User Behavior

Learning from user behavior has been a central task of recommender systems since the advent of collaborative filtering [11], [23], [56], [57], [60]. Being able to predict user preferences through uncovering complex and unexpected patterns hidden in users’ past behaviors without any domain knowledge, factorization based recommendation [36] has become one of the most popular methods in recommender systems. These factorization based collaborative filtering models factorize user and item information into latent representations to approximate user preferences and item recommendations of the latent factors behind user behavior, however, is mostly neglected by the black-box representation learning process adopted by the majority of the existing methods. To the extent of our knowledge, we are the first to study disentangled representation learning on user behavior data.

2.2 Disentangled Representation Learning

Disentangled representation learning aims to identify and disentangle the underlying explanatory factors [3]. Being
capable of producing robust, controllable, and explainable representations, disentangled representation learning has become one of the core problems in machine learning. In general, variational methods are widely applied for disentangled representation over images. \(\beta\)-VAE [21] demonstrates that disentanglement can emerge once the KL divergence term in the VAE [33] objective is aggressively penalized. In particular, Kingma and Welling [33] propose to utilize Bayesian posterior inference and variational estimation to learn the controllable factors hidden in the observed data. Higgins et al. [21] propose \(\beta\)-VAE by setting a weight \(\beta\) for the KL divergence to improve representation disentanglement learned in the observed data while sacrificing mutual information between input data and latent representations. Later approaches separate the information bottleneck term [33], [64] and the total correlation term, and achieve a greater level of disentanglement [7], [8]. Other works either design an attentive architecture to learn aspect matrix for word embeddings [17] or utilize methods based on triplets to learn aspect representations from sentences where each aspect has a separate encoder [29]. Though a few existing approaches [6], [9], [12], [13], [30] do notice that a dataset can contain samples from different concepts, i.e., follow a mixture distribution, their settings are fundamentally different from ours. To be specific, these existing approaches assume that each instance is from a concept, while we assume that each instance interacts with objects from different concepts. The majority of the existing efforts are from the field of computer vision [9], [13], [14], [21], [22], [34], [37], [52], [60]. Disentangled representation learning on relational data, such as graph-structured data, was not explored until recently [19], [66], [76]. This work focuses on disentangling user behavior from both the macro intention and micro preference in recommender systems.

3 Method

In this section, we describe our SEM-MacridVAE model for learning disentangled representations from user behaviors in detail, whose whole framework is demonstrated in Figure 1.

3.1 Notations and Problem Formulation

A user behavior dataset \(D\) consists of the interactions between \(N\) users and \(M\) items. The interaction between the \(u^{th}\) user and the \(i^{th}\) item is denoted by \(x_{u,i} \in \{0, 1\}\), where \(x_{u,i} = 1\) indicates that user \(u\) explicitly adopts item \(i\), whereas \(x_{u,i} = 0\) means there is no recorded interaction between the two. For convenience, we use \(x_u = \{x_{u,i} : x_{u,i} = 1\}\) to represent the items adopted by user \(u\). The goal is to learn user representations \(\{z_{u,i}\}_{i=1}^N\) that achieves both macro and micro disentanglement. We use \(\theta\) to denote the set that contains all the trainable parameters of our model.

3.1.1 Macro Disentanglement

Users may have very diverse interests, and interact with items that belong to many high-level concepts, e.g., product categories. We aim to achieve macro disentanglement, by learning a factorized representation of user \(u\), namely \(z_u = \{z_{u,1}, z_{u,2}, \ldots z_{u,K}\} \in \mathbb{R}^d\), where \(d' = Kd\), assuming that there are \(K\) high-level concepts. The \(k^{th}\) component \(z_{u,k} \in \mathbb{R}^d\) is for capturing the user’s preference regarding the \(k^{th}\) concept. Additionally, we infer a set of one-hot vectors \(C = \{c_{i,1}, c_{i,2}, \ldots c_{i,K}\}\). If item \(i\) belongs to concept \(k\), then \(c_{i,k} = 1\) and \(c_{i,k'} = 0\) for any \(k' \neq k\). We infer \(\{z_{u,i}\}_{i=1}^N\) in an unsupervised way, and learn \(C\) in a supervised manner through the categorical signals of the semantic information.

3.1.2 Micro Disentanglement

High-level concepts correspond to the intentions of a user, e.g., to buy clothes or a cellphone. We are also interested in disentangling a user’s preference at a more granular level regarding the various aspects of an item. For example, we would like the different dimensions of \(z_{u,k}\) to individually capture the user’s preferred sizes, colors, etc., if concept \(k\) is clothing.

3.2 Model

We start by proposing a generative model that encourages macro disentanglement. For a user \(u\), our generative model assumes that the observed data are generated from the following distribution:

\[
p_{\theta}(x_u) = \mathbb{E}_{p_{u}(C)} \left[ \prod_{u} \int p_{\theta}(x_u | z_u, C) p_{\theta}(z_u) \, dz_u \right],
\]

and

\[
p_{\theta}(x_u | z_u, C) = \prod_{i \in N} p_{\theta}(x_{u,i} | z_u, C),
\]

where the meanings of \(x_u, z_u, C\) are described in the previous subsection. We also assume that

\[
p_{\theta}(z_u) = p_{\theta}(z_u | C)
\]

in Equation (1), i.e., \(z_u\) and \(C\) are generated by two independent sources. Note that \(c_{i} = [c_{i,1}; c_{i,2}; \ldots; c_{i,K}]\) is one-hot, since we assume that item \(i\) belongs to exactly one concept. We also remark that

\[
p_{\theta}(x_{u,i} | z_u, C) = Z_u^{-1} \cdot \sum_{k=1}^{K} C_{i,k} \cdot g_{\theta}^{(i)}(z_{u,k}^{(k)})
\]

is a categorical distribution over the \(M\) items, where

\[
Z_u = \sum_{i=1}^{M} \sum_{k=1}^{K} C_{i,k} \cdot g_{\theta}^{(i)}(z_{u,k}^{(k)}) \quad (k)
\]

and \(g_{\theta}^{(i)} : \mathbb{R}^d \rightarrow \mathbb{R}_+\) is a shallow neural network that estimates how much a user with a given preference is interested in item \(i\). We use sample softmax \([29]\) to estimate \(Z_u\) based on a few sampled items when \(M\) is very large.

3.2.1 Macro Disentanglement

We assume above that the user representation \(z_u\) is sufficient for predicting how the user will interact with the items. And we further assume that using the \(k^{th}\) component \(z_{u,k}^{(k)}\) alone is already sufficient if the prediction is about an item from concept \(k\). This design explicitly encourages \(z_{u,k}^{(k)}\) to capture preference regarding only the \(k^{th}\) concept, as long as the inferred concept assignment matrix \(C\) is meaningful.
We follow the variational auto-encoder (VAE) paradigm \cite{33}, and optimize $\theta$ by maximizing a lower bound of $\sum_u \ln p_\theta(x_u)$, where $\ln p_\theta(x_u)$ is bounded as follows:

$$\ln p_\theta(x_u) \geq \mathbb{E}_{q_\theta(z_u | x_u, C)} \left[ \ln p_\theta(x_u | z_u, C) \right] - D_{KL}(q_\theta(z_u | x_u, C) || p_\theta(z_u)).$$

The proof is as follows.

Proof: Given the following equation,

$$q_\theta(z_u, C | x_u) = q_\theta(z_u | x_u, C)p_\theta(C),$$

then we have the following inequality,

$$\ln p_\theta(x_u) = \mathbb{E}_{q_\theta(z_u | x_u, C)} \left[ \ln p_\theta(x_u | z_u, C) \right] + \mathbb{E}_{q_\theta(z_u | x_u, C)} \left[ \ln q_\theta(z_u | x_u, C) \right] - D_{KL}(q_\theta(z_u | x_u, C) || p_\theta(z_u)).$$
A natural strategy to encourage micro disentanglement is to introduce two expectations, i.e.,
\[ E_q(z_u | x_u, C) p_\theta(z_u) p_\theta(C) \]
and the mutual information \( I_q(x_u; z_u) \) is under the joint distribution
\[ q_\theta(z_u, x_u | C) = q_\theta(z_u | x_u, C) p_\text{data}(x_u | C) \]
which completes the proof.

Note that this article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3153112, IEEE Transactions on Pattern Analysis and Machine Intelligence.

3.3 Implementation
In this section, we describe the implementation of \( p_\theta(C) \), \( p_\theta(z_u | z, C) \) (the decoder), \( q_\theta(z_u | C) \) (the encoder), and propose an efficient strategy to combat mode collapse. The parameters \( \theta \) of our implementation include: \( K \) concept prototypes \( \{m_k\}_{k=1}^K \in \mathbb{R}^{K \times d} \), \( M \) item representations \( \{h_i\}_{i=1}^M \in \mathbb{R}^{M \times d} \) used by the decoder, \( M \) context representations \( \{c_i\}_{i=1}^M \in \mathbb{R}^{M \times d} \) used by the encoder, and the parameters of a neural network \( f_{nn} : \mathbb{R}^d \rightarrow \mathbb{R}^{2d} \). We optimize \( \theta \) to maximize the training objective (see Equation 6) using Adam [32].

3.3.1 Prototype-based Concept Assignment
A straightforward approach would be to assume \( p_\theta(C) = \prod_{i=1}^M p(c_i) \) and parameterize each categorical distribution \( p(c_i) \) with its own set of \( K-1 \) parameters. This approach, however, would result in over-parameterization and low sample efficiency. We instead propose a prototype-based implementation. To be specific, we introduce \( K \) concept prototypes \( \{m_k\}_{k=1}^K \) and reuse the item representations \( \{h_i\}_{i=1}^M \) from the decoder. We then assume \( c_i \) is a one-hot vector drawn from the following categorical distribution \( p_\theta(c_i) \):
\[ c_i \sim \text{Categorical}([s_{i,1}; s_{i,2}; \ldots; s_{i,K}]), \]
where \( \text{COSINE}(a, b) = \frac{a^\top b}{\|a\|_2 \|b\|_2} \) is the cosine similarity, and \( \tau \) is a hyper-parameter that scales the similarity from \([-1, 1]\) to \([-\frac{\tau}{2}, \frac{\tau}{2}]\). We set \( \tau = 0.1 \) to obtain a more skewed distribution.
3.3.2 Preventing Mode Collapse

We use cosine similarity, instead of the inner product similarity adopted by most existing deep learning methods [20], [38], [40]. This choice is crucial for preventing mode collapse, which can be a severe issue with a mixture model [24], [25] such as ours if no special treatment is applied, especially when neural networks are involved [62]. In fact, with inner product, the majority of the items are highly likely to be assigned to a single concept \( \mathbf{m}_k \) that has an extremely large norm, i.e., \( \| \mathbf{m}_k \|_2 \to \infty \), even when the items \( \{ \mathbf{h}_i \}_{i=1}^M \) correctly form \( K \) clusters in the high-dimensional Euclidean space. And we observe empirically that this phenomenon does occur frequently with inner product (see Figure 5c and Figure 4). In contrast, cosine similarity avoids this degenerate case due to the normalization. Moreover, cosine similarity is related with the Euclidean distance on the unit hypersphere, and the Euclidean distance is a proper metric that is more suitable for inferring the cluster structure, compared to inner product.

3.3.3 Decoder

The decoder predicts which item out of the \( M \) ones is mostly likely to be clicked by a user, when given the user’s representation \( \mathbf{z}_u = [\mathbf{z}_u^{(1)}; \mathbf{z}_u^{(2)}; \ldots; \mathbf{z}_u^{(K)}] \) and the one-hot concept assignments \( \{ \mathbf{c}_i \}_{i=1}^M \). We assume that

\[
p_\theta(x_{u,i} \mid \mathbf{z}_u, \mathbf{C}) \propto \sum_{k=1}^K c_{i,k} \cdot g_\theta(z_u^{(k)}) \tag{7}
\]

is a categorical distribution over the \( M \) items, and define

\[
g_\theta(z_u^{(k)}) = \exp(\text{COSINE}(\mathbf{z}_u^{(k)}, \mathbf{h}_i)/\tau). \tag{8}
\]

This design implies that \( \{ \mathbf{h}_i \}_{i=1}^M \) will be micro-disentangled if \( \{ \mathbf{z}_u^{(k)} \}_{u=1}^N \) is micro-disentangled, as the two’s dimensions are aligned.

3.3.4 Prior & Encoder

The prior \( p_\theta(\mathbf{z}_u) \) needs to be factorized in order to achieve micro disentanglement. We therefore set \( p_\theta(\mathbf{z}_u) \) to \( \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}) \).

The encoder \( q_\theta(\mathbf{z}_u \mid \mathbf{x}_u, \mathbf{C}) \) is for computing the representation of a user when given the user’s behavior data \( \mathbf{x}_u \), which approximates the posterior. The encoder maintains an additional set of context representations \( \{ \mathbf{t}_i \}_{i=1}^M \), rather than reusing the item representations \( \{ \mathbf{h}_i \}_{i=1}^M \) from the decoder, which is a common practice in the literature [40]. We assume that

\[
q_\theta(\mathbf{z}_u \mid \mathbf{x}_u, \mathbf{C}) = \prod_{k=1}^K q_\theta(\mathbf{z}_u^{(k)} \mid \mathbf{x}_u, \mathbf{C}),
\]

and represent each \( q_\theta(\mathbf{z}_u^{(k)} \mid \mathbf{x}_u, \mathbf{C}) \) as a multivariate normal distribution with a diagonal covariance matrix \( \mathcal{N}(\mu^{(k)}, \text{diag}(\sigma^{(k)})^2) \), where the mean and the standard deviation are parameterized by a neural network \( f_{\text{inv}} : \mathbb{R}^d \to \mathbb{R}^{2d} \).

\[
(a_u^{(k)}, b_u^{(k)}) = f_{\text{inv}}\left(\frac{\sum_{i:x_{u,i}=+1} c_{i,k} \cdot t_i}{\sqrt{\sum_{i:x_{u,i}=+1} c_{i,k}^2}}, \right), \tag{9}
\]

\[
\mu_u^{(k)} = \frac{a_u^{(k)}}{\|a_u^{(k)}\|_2},
\]

\[
\sigma_u^{(k)} \leftarrow \sigma_0 \exp \left(-1 \cdot b_u^{(k)} \right).
\]

The neural network \( f_{\text{inv}}(\cdot) \) captures nonlinearity, and is shared across the \( K \) components. We normalize the mean, so as to be consistent with the use of cosine similarity which projects the representations onto a unit hypersphere. Note that \( \sigma_0 \) should be set to a small value, e.g., around 0.1, since the learned representations are now normalized.

3.4 Incorporating Semantic Information

In this section, we discuss the incorporation of semantic information extracted from items to further boost the model performance. Specifically, we consider two types of semantic information, i.e., visual signals and categorical signals.

**Incorporating Visual Signals.** The two key elements, i.e., concept prototypes \( \{ \mathbf{m}_k \}_{k=1}^K \) and item representations \( \{ \mathbf{h}_i \}_{i=1}^M \), in the prototype mechanism which has a crucial influence on both encoder and decoder, are so far initialized randomly without taking any semantic information from items into consideration.

Therefore, to further improve the model performances, we encode visual semantic information through a pre-trained AlexNet over the raw item image to obtain visual feature \( v_i \) for each item \( i \). To match the dimension of \( v_i \) with that of the item embedding, we conduct Principal Component Analysis (PCA) on \( v_i \). Then we initialize \( \mathbf{h}_i \) with the low-dimensional visual feature conduct initialization for \( \mathbf{m}_k \) by calculating the cluster center (obtained from K-means) of item representations belonging to concept \( k \). Concretely, the visual features are obtained from the output of the last second fully-connected layers of the AlexNet, which has five convolutional layers followed by three fully-connected layers and is pre-trained on the ImageNet dataset with semantic categorical labels. Assuming the visual features output from AlexNet is denoted as \( \{ v_i \}_{i=1}^M = \text{AlexNet}(\cdot) \), then the process of initializing \( \{ \mathbf{h}_i \}_{i=1}^M \) and \( \{ \mathbf{m}_k \}_{k=1}^K \) can be formulated as follows,

\[
\bar{v}_i = \frac{1}{M} \sum_{i=1}^M v_i, \tag{10}
\]

\[
V = \frac{1}{M} \sum_{i=1}^M (v_i - \bar{v}_i)(v_i - \bar{v}_i)^T, \quad V = QAQ^T, \quad P = Q^T : [d],
\]

\[
\{ \mathbf{h}_i \}_{i=1}^M = \{ P^T v_i \}_{i=1}^M, \quad \{ \mathbf{m}_k \}_{k=1}^K = \text{Kmeans}(\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_M),
\]

where each column of \( Q \) represents an eigenvector of \( V \) and \( P \) contains \( d \) eigenvectors corresponding to the largest \( d \) eigenvalues.
Algorithm 1 The training procedure. We add $10^{-8}$ to prevent division-by-zero wherever appropriate.

1: 

2: \textbf{input:} $x_u = \{x_{u,i} : \text{user } u \text{ clicks item } i, \text{i.e., } x_{u,i} = 1\}$.

3: \textbf{parameters:}

4: Concept prototypes $m_k \in \mathbb{R}^d$ for $k = 1, 2, \ldots, K$;
5: Item representations $h_i \in \mathbb{R}^d$ for $i = 1, 2, \ldots, M$;
6: Context representations $\bar{t}_i \in \mathbb{R}^d$ for $i = 1, 2, \ldots, M$;
7: Parameters of a neural network $f_{\text{nn}} : \mathbb{R}^d \rightarrow \mathbb{R}^{2d}$,
8: Item categories: $\hat{c}_i \in \mathbb{R}^K$ for $i = 1, 2, \ldots, M$;

9: \triangleright All these parameters are collectively denoted as $\theta$.

10: \textbf{function} \textsc{Initialization With Semantics}

11: \begin{enumerate}
12: \item \{v_i\}_{i=1}^M = \textit{AlexNet}(.)
13: \item for $i = 1, 2, \ldots, M$
14: \begin{enumerate}
15: \item $v_i = \frac{1}{M} \sum_{i=1}^M v_i$.
16: \item $V = \frac{1}{M} \sum_{i=1}^M (v_i - \bar{v}_i)(v_i - \bar{v}_i)^T$.
17: \item $P = Q^{-T}[/1]$, where $Q = QAQ^T$.
18: \item $\{h_i\}_{i=1}^M = \{Pv_i\}_{i=1}^M$.
19: \item $\{m_k\}_{k=1}^K = \textit{Kmeans}(h_1, h_2, \ldots, h_M)$.
20: \end{enumerate}
21: \end{enumerate}

22: \textbf{return} $\{h_i\}_{i=1}^M, \{m_k\}_{k=1}^K$

23: \textbf{function} \textsc{PrototypeClustering}

24: \begin{enumerate}
25: \item for $i = 1, 2, \ldots, M$
26: \begin{enumerate}
27: \item $s_{i,k} \leftarrow h_i \cdot m_k/(\tau \cdot \|h_i\|_2 \cdot \|m_k\|_2)$, where $k = 1, 2, \ldots, K$.
28: \item $c_i \sim \textsc{Gumbel-Softmax}(s_{i,1}: s_{i,2}: \ldots: s_{i,K})$.
29: \end{enumerate}
30: \end{enumerate}

31: \triangleright At test time, $c_i$ is set to the mode.

32: \textbf{return} $\{c_i\}_{i=1}^M$

33: \textbf{function} \textsc{Encoder}(x_u, $\{c_i\}_{i=1}^M$)

34: \begin{enumerate}
35: \item for $k = 1, 2, \ldots, K$
36: \begin{enumerate}
37: \item $(a_k, b_k) \leftarrow f_{\text{nn}}\left(\sum_{i:x_{u,i}=1} c_{i,k} x_{u,i}; c_{i,k}\right)$.
38: \item $\mu(k) \leftarrow a_k/\|a_k\|_2$.
39: \item $\sigma(k) \leftarrow \sigma_0 \cdot \exp(-\frac{1}{2}b_k)$.
40: \end{enumerate}
41: \end{enumerate}

42: \textbf{return} $z_u, D_{KL}(\mathcal{N}(\mu_u, \text{diag}(\sigma_u)) \| \mathcal{N}(0, \sigma_0 \cdot 1))$

43: \textbf{function} \textsc{Decoder}(z_u, $\{c_i\}_{i=1}^M$)

44: \begin{enumerate}
45: \item $p_{u,i} \leftarrow \sum_{k=1}^K c_{i,k} \cdot \exp(z_u^T h_i/\tau \cdot \|z_u\|_2 \cdot \|h_i\|_2)$, where $i = 1, 2, \ldots, M$.
46: \item $p_{u,1}: p_{u,2}: \ldots: p_{u,M} \leftarrow \textsc{Softmax}([\ln p_{u,1}; \ln p_{u,2}; \ldots; \ln p_{u,M}])$.
47: \end{enumerate}

48: \triangleright We replace the \textsc{Softmax}() above with \textsc{Sampled-Softmax}(), and compute $p_{u,i}$ only if $x_{u,i} = 1$ or item $i$ is sampled, when $M$ is very large.

49: \textbf{return} $\{p_{u,i}\}_{i=1}^M$

50: \textbf{function} \textsc{PrototypeClustering}()

51: \begin{enumerate}
52: \item $\{c_i\}_{i=1}^M \leftarrow \textsc{PrototypeClustering}()$
53: \item $x_{u,i} \leftarrow \textsc{Encoder}(x_u, \{c_i\}_{i=1}^M)$
54: \item $\{p_{u,i}\}_{i=1}^M \leftarrow \textsc{Decoder}(z_u, \{c_i\}_{i=1}^M)$
55: \item $L = -\beta \cdot D_{KL} + \sum_{i:x_{u,i}=1} \ln p_{u,i} + \sum_{i=1}^M \text{Cross}_\text{Entropy}(c_i, \hat{c}_i)$
56: \item $\theta \leftarrow \text{Update } \theta$ to maximize $L$, using the gradient $\nabla_{\theta} L$.
57: \end{enumerate}

Incorporating Categorical Signals. The number of macro concepts, i.e., $K$, are so far preset by human experience, followed by macro disentanglement in an unsupervised manner, which may run the risk of misalignment between the macro concepts and actual categories of items despite massive cost on trying and testing. Therefore, we utilize the categorical semantic information to achieve better macro disentanglement through supervised categorical signals in the following:

$$\min \sum_{i=1}^M \text{Cross}_\text{Entropy}(c_i, \hat{c}_i),$$

where $\hat{c}_i$ is one-hot vector that reflects the ground-truth category of the $i^{th}$ item and $\text{Cross}_\text{Entropy}(c_i, \hat{c}_i)$ denotes the binary classification loss between the learned category and true category of the item $i$.

Empirical results in our experiments later show that by taking semantic information, i.e., visual and categorical signals, into account, the proposed SEM-MacridVAE model is able to outperform MacridVAE which initializes item representations and concept prototypes in a random manner. The incorporation of item semantic information is illustrated in the upper part of Figure 3. Algorithm 1 presents the implementation details of the whole procedure.

3.5 User-controllable Recommendation

The controllability enabled by the disentangled representations can bring a new paradigm for recommendation. It allows a user to interactively search for items that are similar to an initial item except for some controlled aspects, or to explicitly adjust the disentangled representation of his/her preference, learned by the system from his/her past behaviors, to actually match the current preference. Here, we formalize the task of user-controllable recommendation, and illustrate a possible solution.

3.5.1 Task Definition

Let $h_u \in \mathbb{R}^d$ be the representation to be altered, which can be initialized as either an item representation or a component of a user representation. The task is to gradually alter its $j^{th}$ dimension $h_{u,j}$ while retrieving items whose representations are similar to the altered representation. This task is non-trivial, since usually no item will have exactly the same representation as the altered one, especially when we want the transition to be smooth, monotonic, and thus human-understandable.

3.5.2 Solution

Here we illustrate our approach to this task. We first probe the suitable range $(a, b)$ for $h_{u,j}$. Let us assume that prototype $k_*$ is the prototype closest to $h_u$. The range $(a, b)$ is decided such that: prototype $k_*$ remains the prototype closest to $h_u$ if and only if $h_{u,j} \in (a, b)$. We can decide each endpoint of the range using binary search. We then divide the range $(a, b)$ into $B$ subranges, $a = a_0 < a_1 < a_2 \ldots < a_B = b$. We ensure that the subranges contain roughly the same number of items from concept $k_*$ when dividing $(a, b)$. Finally, we aim to retrieve $B$ items $\{i_b\}_{B=1}^B \in \{1, 2, \ldots, M\}$ that belong...
to concept $k_s$, each from one of the $B$ subranges, i.e., $h_{i,j} \in \{a_{l-1}, a_l\}$. We thus decide the $B$ items by maximizing

$$
\sum_{1 \leq i \leq B} e^{\max(h_{i,j} \cdot h_{\star, j})} + \gamma \sum_{1 \leq i' \leq B} e^{\max(h_{i,j}, h_{i', j})},
$$

(12)

where $h_{i,j} = [h_{i,1}; h_{i,2}; \ldots; h_{i,j-1}; h_{i,j+1}; \ldots; h_{i,d}] \in \mathbb{R}^{d-1}$ and $\gamma$ is a hyper-parameter. We approximately solve this maximization problem sequentially using beam search [45].

Intuitively, selecting items from the $B$ subranges ensures that the items change monotonously in terms of the $j$th dimension. On the other hand, the first term in the maximization problem forces the retrieved items to be similar with the initial item in terms of the dimensions other than $j$, while the second term encourages any two retrieved items to be similar in terms of the dimensions other than $j$.

We highlight in Figure 6, Figure 7 and Figure 8 some example cases that we found using this approach.

4 **Empirical Experiments**

In this section, we demonstrate that our learned disentangled representations are not only effective for recommendation, but also interpretable and controllable.

4.1 **Experimental Setup**

4.1.1 **Datasets**

We conduct extensive experiments on five public real-world datasets, including a MovieLens dataset (i.e., ML-latest-small) [18] and four Amazon product datasets [18] of different meta categories (i.e., Movies&TV, Musical Instruments, Home&Kitchen and Clothing&Shoes&Jewelry). We follow MultiVAE [40], and binarize these five datasets by labeling ratings of four or higher as $\star$, and some other work [61] utilizes social network information for social recommendation. These works are orthogonal to our focus in this paper and therefore are not included for comparisons in the experiments.

4.1.2 **Baselines**

We compare our approach with four baselines, MultiDAE [40], MultiVAE [40], MacridVAE [51] and DGCF [66]. MultiDAE [40] and MultiVAE [40] are the two state-of-the-art methods for collaborative filtering. In particular, MultiVAE is similar to $\beta$-VAE [21], and has a hyper-parameter $\beta$ that controls the strength of disentanglement. However, MultiVAE does not learn disentangled representations, because it requires $\beta \ll 1$ to perform well. MacridVAE [51] can be treated as a variant of SEM-MacridVAE which conducts random initialization without considering any semantic information from items, and we compare it with the proposed SEM-MacridVAE model to further verify the improvement brought by incorporating item semantic information. Besides, DGCF [66] is chosen as the comparative baseline given that it is one of the most recent works focusing on disentangled collaborative filtering.

We would also like to point out that there are also several works related to disentangled representation for recommendation [39], [61], [51], [71], [77]. However, we find that some of these works require multimodal [67], [77] or heterogeneous [71] information as input, some [39] in essence can be regarded as a $\beta$-VAE model with varying $\beta$, and some other work [61] utilizes social network information for social recommendation.

4.1.3 **Hyper-parameters**

We constrain the number of learnable parameters to be around $2Md$ for each method so as to ensure fair comparison, which is equivalent to using $d$-dimensional representations for the $M$ items. Note that all the methods under investigation use two sets of item representations, and we do not constrain the dimension of user representations since they are not parameters. We treat $K$ as a hyper-parameter to be tuned and do not directly set $K$ to the ground truth when $\beta$, [67] evaluating its performance on recommendation tasks, so as to ensure a fair comparison with the baselines. We set $d = 200$ and fix $\tau$ to $0.1$. The neural network $f_{nn}(\cdot)$ in our model is a multilayer perceptron (MLP), whose input and output are constrained to be $d$-dimensional and $2d$-dimensional, respectively. We use the tanh activation function. We apply dropout before every layers, except the last layer. The model is trained using Adam. We then tune the other hyper-parameters of both our approach’s and our baselines’ automatically using the TPE method [5] implemented by Hyperopt [4]. We let Hyperopt conduct 200 trials to search for the optimal hyper-parameter configuration for each method on the validation of each dataset. The hyper-parameter search space is specified as follows:

- The standard deviation of the prior $\sigma_0 \in [0.075, 0.5]$.
- The strength of micro disentanglement $\beta \in [0, 100]$.
- The number of macro factors $K \in \{1, 2, 3, \ldots, 20\}$.
- The learning rate $\in [10^{-8}, 1]$.
- L2 regularization $\in [10^{-12}, 1]$.
- Dropout rate $\in [0.05, 1]$.
- The number of hidden layers in a neural network $\in \{0, 1, 2, 3\}$.
- The number of neurons in a hidden layer $\in \{50, 100, 150, \ldots, 700\}$.

4.1.4 **Number of Macro Factors**

Our initial implementation adaptively adjusts the number of macro factors $K$ during training. To be specific, we set $K$ as a sufficiently large value at the beginning and shrink its value after every training epoch if the Jensen–Shannon (JS) divergence between $\{p_{ik}y_{ij}^{M}\}_{i=1}^{M}$ and $\{p_{ik}y_{ij}^{M}\}_{i=1}^{M}$ for some $k \neq k'$ is negligible compared to a predefined threshold.
We evaluate the performance of our approach on the task of collaborative filtering for implicit feedback datasets [23].

where $p_{i|k} = p_{\theta}(c_{i,k} = 1)/\sum_{c_{i,k}} p_{\theta}(c_{i,k} = 1)$. We, however, do not find this adaptive strategy to be significantly better than the naïve strategy that treats $K$ as a hyper-parameter to be tuned by Hyperopt, since the adaptive strategy introduces extra computational cost as well as a new hyper-parameter.

### 4.1.5 Experimental Environment

We implement our model with Tensorflow, and conduct our experiments with:

- CPU: Intel(R) Xeon(R) CPU E5-2699 v4 @ 2.20GHz.
- RAM: DDR4 1TB.
- GPU: 8x GeForce GTX 1080 Ti.
- Operating system: Ubuntu 18.04 LTS.
- Software: Python 3.6; NumPy 1.15.4; SciPy 1.2.0; scikit-learn 0.20.0; TensorFlow 1.12.

#### Dataset

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Users</th>
<th>Item</th>
<th>Ratings</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML-latest-small</td>
<td>531</td>
<td>4807</td>
<td>46217</td>
<td>1.8106%</td>
</tr>
<tr>
<td>Movies&amp;TV</td>
<td>15187</td>
<td>46234</td>
<td>62329</td>
<td>0.0888%</td>
</tr>
<tr>
<td>Musical Instruments</td>
<td>655</td>
<td>10377</td>
<td>15540</td>
<td>0.2286%</td>
</tr>
<tr>
<td>Home&amp;Kitchen</td>
<td>6640</td>
<td>52900</td>
<td>154622</td>
<td>0.0440%</td>
</tr>
<tr>
<td>Clothing&amp;shoes&amp;Jewelry</td>
<td>9575</td>
<td>130742</td>
<td>197708</td>
<td>0.0158%</td>
</tr>
</tbody>
</table>

#### Metrics

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>Metrics 1</th>
<th>Metrics 2</th>
<th>Metrics 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML-latest-small</td>
<td>MultiDAE</td>
<td>0.31930(±0.02657)</td>
<td>0.27885(±0.03869)</td>
<td>0.37373(±0.03658)</td>
</tr>
<tr>
<td></td>
<td>MultiVAE</td>
<td>0.33233(±0.03031)</td>
<td>0.28539(±0.03659)</td>
<td>0.38718(±0.03774)</td>
</tr>
<tr>
<td></td>
<td>MacridVAE</td>
<td>0.34180(±0.03190)</td>
<td>0.29444(±0.03711)</td>
<td>0.39994(±0.03696)</td>
</tr>
<tr>
<td></td>
<td>DGCF</td>
<td>0.34709(±0.02971)</td>
<td>0.29185(±0.03433)</td>
<td>0.39353(±0.03836)</td>
</tr>
<tr>
<td></td>
<td>SEM-MacridVAE</td>
<td>0.35129(±0.03204)</td>
<td>0.30293(±0.03591)</td>
<td>0.40026(±0.03724)</td>
</tr>
<tr>
<td>Movies&amp;TV</td>
<td>MultiDAE</td>
<td>0.09774(±0.00336)</td>
<td>0.08342(±0.00382)</td>
<td>0.13936(±0.00498)</td>
</tr>
<tr>
<td></td>
<td>MultiVAE</td>
<td>0.09953(±0.00338)</td>
<td>0.08431(±0.00387)</td>
<td>0.14004(±0.00496)</td>
</tr>
<tr>
<td></td>
<td>MacridVAE</td>
<td>0.11619(±0.00365)</td>
<td>0.10397(±0.00426)</td>
<td>0.16011(±0.00527)</td>
</tr>
<tr>
<td></td>
<td>DGCF</td>
<td>0.10060(±0.00328)</td>
<td>0.08541(±0.00380)</td>
<td>0.14666(±0.00498)</td>
</tr>
<tr>
<td></td>
<td>SEM-MacridVAE</td>
<td>0.11674(±0.00367)</td>
<td>0.10466(±0.00423)</td>
<td>0.16101(±0.00515)</td>
</tr>
<tr>
<td>Musical Instruments</td>
<td>MultiDAE</td>
<td>0.04508(±0.01194)</td>
<td>0.03171(±0.01767)</td>
<td>0.09709(±0.03196)</td>
</tr>
<tr>
<td></td>
<td>MultiVAE</td>
<td>0.04420(±0.01156)</td>
<td>0.03436(±0.01781)</td>
<td>0.09590(±0.03022)</td>
</tr>
<tr>
<td></td>
<td>MacridVAE</td>
<td>0.05706(±0.01871)</td>
<td>0.04034(±0.01885)</td>
<td>0.09419(±0.03199)</td>
</tr>
<tr>
<td></td>
<td>DGCF</td>
<td>0.06109(±0.01657)</td>
<td>0.08352(±0.03310)</td>
<td>0.11870(±0.03754)</td>
</tr>
<tr>
<td></td>
<td>SEM-MacridVAE</td>
<td>0.06450(±0.01415)</td>
<td>0.08479(±0.03132)</td>
<td>0.13436(±0.03829)</td>
</tr>
<tr>
<td>Home&amp;Kitchen</td>
<td>MultiDAE</td>
<td>0.03577(±0.00401)</td>
<td>0.03488(±0.00499)</td>
<td>0.06190(±0.0067)</td>
</tr>
<tr>
<td></td>
<td>MultiVAE</td>
<td>0.03761(±0.00420)</td>
<td>0.03607(±0.00514)</td>
<td>0.06094(±0.00671)</td>
</tr>
<tr>
<td></td>
<td>MacridVAE</td>
<td>0.04271(±0.00456)</td>
<td>0.03641(±0.00510)</td>
<td>0.06737(±0.00659)</td>
</tr>
<tr>
<td></td>
<td>DGCF</td>
<td>0.04370(±0.00404)</td>
<td>0.03853(±0.00494)</td>
<td>0.07699(±0.00708)</td>
</tr>
<tr>
<td></td>
<td>SEM-MacridVAE</td>
<td>0.04463(±0.00434)</td>
<td>0.04669(±0.00557)</td>
<td>0.07913(±0.00727)</td>
</tr>
<tr>
<td>Clothing&amp;shoes&amp;Jewelry</td>
<td>MultiDAE</td>
<td>0.01123(±0.00213)</td>
<td>0.01156(±0.00297)</td>
<td>0.01725(±0.00353)</td>
</tr>
<tr>
<td></td>
<td>MultiVAE</td>
<td>0.01107(±0.00182)</td>
<td>0.01278(±0.00296)</td>
<td>0.02507(±0.00432)</td>
</tr>
<tr>
<td></td>
<td>MacridVAE</td>
<td>0.01785(±0.00265)</td>
<td>0.01540(±0.00313)</td>
<td>0.03009(±0.00458)</td>
</tr>
<tr>
<td></td>
<td>DGCF</td>
<td>0.01833(±0.00296)</td>
<td>0.02293(±0.00462)</td>
<td>0.03691(±0.00558)</td>
</tr>
<tr>
<td></td>
<td>SEM-MacridVAE</td>
<td>0.01853(±0.00240)</td>
<td>0.02491(±0.00434)</td>
<td>0.03720(±0.00517)</td>
</tr>
</tbody>
</table>

**TABLE 1: Statistics of the datasets.**

**TABLE 2: Results of recommendation performance, where bold font denotes the winner.**
property implies that our approach can be more robust to the scenario where multiple factors are co-influencing the data generating process, especially when there is a limited amount of available data \(3\). For example, it would not overreact to the preference for bag size when making a prediction that is only related with the preference for bag color.

**SEM-MacridVAE v.s. MacridVAE**

The comparisons between SEM-MacridVAE with item semantic information and MacridVAE without semantic information in Table 2 further validate the benefit of considering semantics in boosting model performances. Indeed, incorporating semantic meanings has been regarded as one effective way to improve both model accuracy and explainability of machine learning algorithms in the community.

**With Macro & Micro Disentanglement v.s. Without Macro & Micro Disentanglement**

Ablation studies (Without Macro and Without Micro) in Table 3 confirm the benefit of conducting both Macro and Micro disentanglement when making recommendations.

**With Visual & Categorical Signals v.s. Without Visual & Categorical Signals**

Similarly, comparisons for Without Visual, Without Categorical and SEM-MacridVAE (Full Model) in Table 3 further validate the necessity of incorporating semantic information to boost the model performance.

**Discussions**

The ablation studies on the two-level (macro and micro) disentanglement and the semantic (visual and categorical) signals show the improvement of the model performance brought by these core components in our proposed SEM-MacridVAE model. On the one hand, the macro disentanglement helps to capture user high-level intentions from a diversity of potential interests, while the micro disentanglement targets at learning the user low-level preferences in a more fine-grained way. Taking both levels of intentions into consideration enables the proposed SEM-MacridVAE model to more accurately infer user interests, thus improving the final recommendation performance. On the other hand, incorporating the categorical signals can more accurately align the learned macro disentangled intentions (i.e., the prototype concepts) to the ground-truth item categories, thus leading to better representation learning. Moreover, employing the visual signals to initialize the item embeddings is able to ease the process of learning detailed user visual preferences (e.g., the color of a bag) for our SEM-MacridVAE model. These two types of semantic supervision are taken into account to improve both the disentanglement and explainability of SEM-MacridVAE with human prior, which is illustrated by the ablation studies in Table 2 as we expect.

4.3 Macro Disentanglement

In order that we can qualitatively examine to which degree our proposed SEM-MacridVAE model is able to achieve macro disentanglement, the high-dimensional representations learned by our approach are visualized on three Amazon datasets, i.e., Amazon Musical Instruments, Amazon Home&Kitchen and Amazon Clothing&Shoes&Jewelry. We pick subsets from these three Amazon datasets respectively such that every item only belongs to one category, and the number of items in every category is balanced to be close to each other. Concretely, we set \(K\) to 4 for Amazon Musical Instruments, 5 for Amazon Home&Kitchen and 3 for Amazon Clothing&Shoes&Jewelry i.e., the number of ground-truth categories, when training our model. We then match each learned prototype to a ground truth category by greedily minimizing the distance between the prototype and the center of the items from that category. We visualize the item representations and the user representations together using t-SNE \(5\), where we treat the \(K\) components of a user as \(K\) individual points and keep only the two components that have the highest confidence levels. The confidence of component \(k\) is defined as \(\sum_{i \in X_k} > 0 c_{i,k}\), where \(c_{i,k}\) is the confidence of item \(i\) assigned to category \(k\).

4.3.1 Interpretability

Again, we take the Amazon datasets as instances. Figure 2a, Figure 3a and Figure 4a depict the visualization results on Amazon Musical Instruments, Amazon Home&Kitchen and Amazon Clothing&Shoes&Jewelry respectively. Specifically, item \(i\) is colored according to \(\arg \max_k c_{i,k}\), i.e., the inferred category. The discovered clusters of items (see Figure 2a, Figure 3a and Figure 4a), learned in an unsupervised manner, align well with the ground-truth categories (see Figure 2b, Figure 3b and Figure 4b, where the color order is chosen such that the connections between the ground-truth categories and the learned clusters are easy to verify). Figure 2c, Figure 3c and Figure 4c highlight the importance of using cosine similarity rather than inner product to combat mode collapse, where items are obtained by training a new model that uses inner product instead of cosine, colored according to the value of \(\arg \max_k c_{i,k}\).

4.3.2 Cosine vs. Inner Product

To further present the necessity of adopting cosine similarity instead of the widely used inner product similarity, we train an additional model using inner product rather than cosine to calculate similarity. The item representations obtained from this additional model on Amazon Musical Instruments, Amazon Home&Kitchen and Amazon Clothing&Shoes&Jewelry are visualized in Figure 2d, Figure 3d and Figure 4d respectively.

We observe that by adopting inner product as the similarity measure, the clustering results may vary for different datasets. The prototype assignments are similar on Amazon Musical Instruments (see Figure 2c), while the majority of the items are assigned to the same prototype on Amazon Home&Kitchen (see Figure 3c) or assigned to wrong prototypes different from the ground-truth prototypes on Amazon Clothing&Shoes&Jewelry (see Figure 4c). On the other hand, each of the prototypes learned by the cosine-based model is assigned quite a significant number of items, being
TABLE 3: Ablation studies on Macro & Micro disentanglement and Visual & Categorical signals, where bold font denotes the winner. The experimental results demonstrate our proposed model with full functionality (i.e., SEM-MacridVAE) achieve the best performance.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>Metrics</th>
<th>NDCC@100</th>
<th>Recall@20</th>
<th>Recall@50</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML-latest-small</td>
<td>Without Macro</td>
<td></td>
<td>0.3457(±0.03125)</td>
<td>0.2986(±0.03481)</td>
<td>0.3899(±0.03898)</td>
</tr>
<tr>
<td></td>
<td>Without Micro</td>
<td></td>
<td>0.3387(±0.02977)</td>
<td>0.2914(±0.03482)</td>
<td>0.3825(±0.03828)</td>
</tr>
<tr>
<td></td>
<td>Without Visual</td>
<td></td>
<td>0.3456(±0.03222)</td>
<td>0.2999(±0.03729)</td>
<td>0.3850(±0.03750)</td>
</tr>
<tr>
<td></td>
<td>Without Categorical</td>
<td></td>
<td>0.3446(±0.03130)</td>
<td>0.3027(±0.03492)</td>
<td>0.3941(±0.03824)</td>
</tr>
<tr>
<td></td>
<td>SEM-MacridVAE (Full Model)</td>
<td></td>
<td>0.3512(±0.03204)</td>
<td>0.3029(±0.03591)</td>
<td>0.4036(±0.03724)</td>
</tr>
<tr>
<td>Movies &amp;TV</td>
<td>Without Macro</td>
<td></td>
<td>0.1130(±0.00359)</td>
<td>0.1029(±0.00422)</td>
<td>0.1593(±0.00521)</td>
</tr>
<tr>
<td></td>
<td>Without Micro</td>
<td></td>
<td>0.1106(±0.00345)</td>
<td>0.1021(±0.00424)</td>
<td>0.1606(±0.00522)</td>
</tr>
<tr>
<td></td>
<td>Without Visual</td>
<td></td>
<td>0.1152(±0.00365)</td>
<td>0.1036(±0.00417)</td>
<td>0.1602(±0.00521)</td>
</tr>
<tr>
<td></td>
<td>Without Categorical</td>
<td></td>
<td>0.1155(±0.00365)</td>
<td>0.1033(±0.00424)</td>
<td>0.1601(±0.00515)</td>
</tr>
<tr>
<td></td>
<td>SEM-MacridVAE (Full Model)</td>
<td></td>
<td>0.1167(±0.00367)</td>
<td>0.1046(±0.00423)</td>
<td>0.1610(±0.00515)</td>
</tr>
<tr>
<td>Musical Instruments</td>
<td>Without Macro</td>
<td></td>
<td>0.0638(±0.01775)</td>
<td>0.0711(±0.02798)</td>
<td>0.11726(±0.03532)</td>
</tr>
<tr>
<td></td>
<td>Without Micro</td>
<td></td>
<td>0.0620(±0.01383)</td>
<td>0.0711(±0.02798)</td>
<td>0.13733(±0.03819)</td>
</tr>
<tr>
<td></td>
<td>Without Visual</td>
<td></td>
<td>0.0580(±0.01923)</td>
<td>0.0449(±0.02238)</td>
<td>0.10265(±0.03253)</td>
</tr>
<tr>
<td></td>
<td>Without Categorical</td>
<td></td>
<td>0.0635(±0.01426)</td>
<td>0.0826(±0.02881)</td>
<td>0.13656(±0.03621)</td>
</tr>
<tr>
<td></td>
<td>SEM-MacridVAE (Full Model)</td>
<td></td>
<td>0.0645(±0.01415)</td>
<td>0.0849(±0.03132)</td>
<td>0.1343(±0.03829)</td>
</tr>
<tr>
<td>Home &amp;Kitchen</td>
<td>Without Macro</td>
<td></td>
<td>0.0437(±0.00404)</td>
<td>0.0441(±0.00556)</td>
<td>0.07332(±0.00713)</td>
</tr>
<tr>
<td></td>
<td>Without Micro</td>
<td></td>
<td>0.0436(±0.00429)</td>
<td>0.0424(±0.00535)</td>
<td>0.07166(±0.00690)</td>
</tr>
<tr>
<td></td>
<td>Without Visual</td>
<td></td>
<td>0.0426(±0.00437)</td>
<td>0.0406(±0.00525)</td>
<td>0.07030(±0.00689)</td>
</tr>
<tr>
<td></td>
<td>Without Categorical</td>
<td></td>
<td>0.0444(±0.00426)</td>
<td>0.0425(±0.00520)</td>
<td>0.0748(±0.00692)</td>
</tr>
<tr>
<td></td>
<td>SEM-MacridVAE (Full Model)</td>
<td></td>
<td>0.0446(±0.00434)</td>
<td>0.0469(±0.00557)</td>
<td>0.0791(±0.00727)</td>
</tr>
<tr>
<td>Clothing &amp;Shoes&amp;Jewelry</td>
<td>Without Macro</td>
<td></td>
<td>0.0181(±0.00254)</td>
<td>0.0223(±0.00398)</td>
<td>0.0305(±0.00470)</td>
</tr>
<tr>
<td></td>
<td>Without Micro</td>
<td></td>
<td>0.0180(±0.00257)</td>
<td>0.0221(±0.00398)</td>
<td>0.0366(±0.00511)</td>
</tr>
<tr>
<td></td>
<td>Without Visual</td>
<td></td>
<td>0.0164(±0.00246)</td>
<td>0.0196(±0.00369)</td>
<td>0.0310(±0.00469)</td>
</tr>
<tr>
<td></td>
<td>Without Categorical</td>
<td></td>
<td>0.0185(±0.00250)</td>
<td>0.0224(±0.00406)</td>
<td>0.0332(±0.00476)</td>
</tr>
<tr>
<td></td>
<td>SEM-MacridVAE (Full Model)</td>
<td></td>
<td>0.0183(±0.00240)</td>
<td>0.0249(±0.00434)</td>
<td>0.0372(±0.00517)</td>
</tr>
</tbody>
</table>

consistent with the ground-truth categories (see Figure 2a and Figure 3a).

These results support our claim that an appropriate metric space such as the one defined through the cosine similarity will play an important role in preventing the mode collapse problem.

4.4 Micro Disentanglement

In addition to macro disentanglement, it is also necessary to examine the capability of our proposed SEM-MacridVAE in achieving micro disentanglement.

4.4.1 Independence

One important motivation of disentangled representation learning is to achieve robust performance by letting the dimensions capture the underlying explanatory factors in a statistically independent way.

To gain further insight, we vary the hyper-parameters related with micro disentanglement, i.e., β for our proposed SEM-MacridVAE, MacridVAE and MultiVAE. In Figure 5, we plot the relationships between the level of independence (micro disentanglement) achieved and the corresponding recommendation performance. Each method is evaluated on ML-latest-small, Amazon Musical Instruments, Amazon Movies&TV, Amazon Home&Kitchen and Amazon Clothing&Shoes&Jewelry. We quantify the level of independence achieved by a set of d-dimensional representations using

\[ 1 - \frac{2}{d(d-1)} \sum_{1 \leq i < j \leq d} |\text{corr}_{ij}|, \]

where \( |\text{corr}_{ij}| \) is the correlation between dimension \( i \) and \( j \). Figure 5 indicates that high performance is in general associated with a relatively high level of independence (micro disentanglement) and SEM-MacridVAE achieves a higher level of micro disentanglement than MultiVAE.

4.4.2 Interpretability

We train our SEM-MacridVAE model with \( d = 10 \), \( \beta = 50 \) and \( \sigma_0 = 0.3 \) on Amazon datasets, and investigate the interpretability of the dimensions using strategies introduced in Section 3.5.

In Figure 6, Figure 7 and Figure 8, we retrieve some representative dimensions that have human-understandable semantics on Amazon Musical Instruments, Amazon Home&Kitchen and Amazon Clothing&Shoes&Jewelry respectively. The examples from these three datasets suggest that our SEM-MacridVAE model has the potential to offer users fine-grained controls over targeted aspects of the candidate items in recommendation lists. However, we note that not all dimensions are human-understandable.

Moreover, as is pointed out by Locatello et al. [43], well-trained interpretable models can only be reliably identified with the help of external knowledge, e.g., item attributes. Therefore, we encourage future efforts in investigating more semi-supervised methods [44] for disentangled representation learning.
4.5 Model Complexity Analysis

In addition to the performance illustration and interpretability visualization of our proposed SEM-MacridVAE model, we further provide the model complexity analysis.

**Space Complexity** As mentioned before, the space complexity, i.e., the number of parameters used by SEM-MacridVAE, is $2Md$ where $M$ is the number of items and $d$ is the dimension of latent factors.

**Time Complexity** We analyze the time complexity according to the sequential execution pipeline of the proposed algorithm by calculating the times of element multiplication. Assuming that there are $N$ users and $M$ items, the Prototype Clustering process requires $O(MdK)$ times of multiplications. Both the Encoding process and Decoding process need $O(NMdK)$ times of multiplications. The incorporation of visual signals is pre-trained and does not consume extra running time during the training process. The incorporation of categorical signals needs $O(MK)$ times of multiplications. Summing up all the above operations, the time complexity of our SEM-MacridVAE model is $O(NMdK + NMdK + MdK + MK) = O(NMdK)$ times of element multiplications.

5 Conclusions

In this paper, we study the problem of learning disentangled representations from user behaviors, and propose our SEM-MacridVAE model capable of performing disentanglement at both macro and micro levels. We relate macro factors to high-level concepts associated with user intentions (buy a pair of shoes or a laptop) and micro factors to low-level individual user preferences (the size or the color of a shirt). Extra semantic item information, including visual semantic and categorical semantic, are further taken into consideration to boost recommendation performance. Empirical results including both quantitative and qualitative experiments over five real-world datasets demonstrate the effectiveness of our approach in learning disentangled representations that are robust, interpretable, and controllable.

As for future work, it will be an interesting and promising research direction for future investigation to explore novel
(a) Items in Amazon Home&Kitchen, colored based on the predicted categories.
(b) Items in Amazon Home&Kitchen, colored based on the ground-truth categories.
(c) Items obtained by training a new model using inner product instead of cosine.

(d) Users in Amazon Home&Kitchen, colored based on the predicted categories.
(e) Items and Users in Amazon Home&Kitchen, colored based on the predicted categories.

Fig. 3: Visualization of macro disentanglement for Amazon Home&Kitchen with $K = 5$, in the same way as Figure 2.

(a) Items in Amazon Clothing&Shoes, colored based on the predicted categories.
(b) Items in Amazon Clothing&Shoes, colored based on the ground-truth categories.
(c) Items obtained by training a new model using inner product instead of cosine.

(d) Users in Amazon Clothing&Shoes, colored based on the predicted categories.
(e) Items and Users in Amazon Clothing&Shoes, colored based on the predicted categories.

Fig. 4: Visualization of macro disentanglement for Amazon Clothing&Shoes&Jewelry with $K = 3$, in the same way as Figure 2.
Fig. 5: Micro disentanglement vs. recommendation performance. By varying the hyper-parameters $\beta$, we compare the micro disentanglement and recommendation performance. It is observed that SEM-MacridVAE overall outperforms both MacridVAE and multiVAE in terms of both micro disentanglement and recommendation performance under recall@20.

Fig. 6: Starting from an item representation, we gradually change the value of a target dimension and retrieve the items having similar representations with the changed representations, as is described in Section 3.5. Here we present the retrieved items in Amazon Musical Instruments dataset when varying the target dimension and fixing others.
applications that can benefit in the interpretability and controllability brought by the disentangled representations.

ACKNOWLEDGMENTS

This work is supported by the National Key Research and Development Program of China No. 2020AAA0106300 and National Natural Science Foundation of China No. 6210222.

REFERENCES


Hong Chen received B.E. from the Department of Electronic Engineering, Tsinghua University, Beijing, China in 2020. He is currently a Ph.D. candidate in the Department of Computer Science and Technology of Tsinghua University. His main research interests include machine learning, multimodal information processing.

Yuwei Zhou is an undergraduate student at the Department of Computer Science and Technology, Tsinghua University. His research interests mainly include knowledge-driven, interpretable and disentangled representation learning.

Jianxin Ma received his B.E. and master's degree from the Department of Computer Science and Technology, Tsinghua University in 2017 and 2020, respectively, under the supervision of Wenwu Zhu and Peng Cui. His research interests are mainly in machine learning, in particular representation learning, on relational data such as graph data and user behavior data from recommender systems. He has published several papers at prestigious conferences such as KDD, AAAI, ICML, and NeurIPS. He is now doing applied research at DAMO academy, Alibaba Inc.

Wenwu Zhu is currently a Professor in the Department of Computer Science and Technology at Tsinghua University, the Vice Dean of National Research Center for Information Science and Technology. Prior to his current post, he was a Senior Researcher and Research Manager at Microsoft Research Asia. He was the Chief Scientist and Director at Intel Research China from 2004 to 2008. He worked at Bell Labs New Jersey as Member of Technical Staff during 1996-1999. He received his Ph.D. degree from New York University in 1996.

His current research interests are in the area of data-driven multimedia networking and Cross-media big data computing. He has published over 350 referred papers, and is inventor or co-inventor of over 50 patents. He received eight Best Paper Awards, including ACM Multimedia 2012 and IEEE Transactions on Circuits and Systems for Video Technology in 2001 and 2019.

He served as EiC for IEEE Transactions on Multimedia (2017-2019). He served in the steering committee for IEEE Transactions on Multimedia (2015-2016) and IEEE Transactions on Mobile Computing (2007-2010), respectively. He serves as General Co-Chair for ACM Multimedia 2018 and ACM CIKM 2019, respectively. He is an AAAS Fellow, IEEE Fellow, SPIE Fellow, and a member of The Academy of Europe (Academia Europaea).