



Disentangled Variational Autoencoder for Social Recommendation

Yongshuai Zhang^{1,2} · Jiajin Huang^{1,2} · Jian Yang^{1,2}

Accepted: 21 March 2024
© The Author(s) 2024

Abstract

Social recommendation aims to improve the recommendation performance by learning user interest and social representations from users' interaction records and social relations. Intuitively, these learned representations entangle user interest factors with social factors because users' interaction behaviors and social relations affect each other. A high-quality recommender system should provide items to a user according to his/her interest factors. However, most existing social recommendation models aggregate the two kinds of representations indiscriminately, and this kind of aggregation limits their recommendation performance. In this paper, we develop a model called **Disentangled Variational autoencoder for Social Recommendation (DVSR)** to disentangle interest and social factors from the two kinds of user representations. Firstly, we perform a preliminary analysis of the entangled information on three popular social recommendation datasets. Then, we present the model architecture of DVSR, which is based on the Variational AutoEncoder (VAE) framework. Besides the traditional method of training VAE, we also use contrastive estimation to penalize the mutual information between interest and social factors. Extensive experiments are conducted on three benchmark datasets to evaluate the effectiveness of our model.

Keywords Social recommendation · Disentangled representation learning · Variational autoencoder · Contrastive learning

1 Introduction

A recommender system aims to help users to select information they are potentially interested in. In recent years, the explosive growth of e-commerce has made recommendations an

✉ Jian Yang
jiyang@bjut.edu.cn

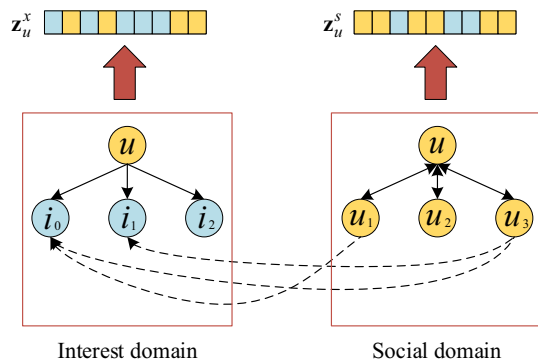
Yongshuai Zhang
zhangyongshuai@emails.bjut.edu.cn

Jiajin Huang
jhuang@bjut.edu.cn

¹ Faculty of Information Technology, Beijing University of Technology, Beijing, China

² Beijing International Collaboration Base on Brain Informatics and Wisdom Services, Beijing, China

Fig. 1 Phenomenon of entanglement information in social recommendation. Factors for interest and social information in embeddings \mathbf{z}_u^x and \mathbf{z}_u^s are in blue and yellow, respectively. (Color figure online)



indispensable tool for users, sellers and platforms [1–4]. On social media platforms, users can build their social relations and share their interested items conveniently and quickly. Social recommendation is becoming a popular branch that aims to combine interaction records and social relations for the item recommendation task [5–7]. To effectively fuse interest and social information, it is inevitable and important to integrate user interest representations and user social representations. These representations are always characterized by low-dimensional vectors called embeddings in a latent space. Under the assumption that socially connected users tend to have similar interests, there are two types of methods to integrate interest and social embeddings of a user for recommendations. One performs simple pooling operations (e.g., concatenating or addition operation) on them to generate the final user representation [8–11]. The other designs auxiliary tasks (e.g., adversarial or contrastive learning tasks) to make interest and social embeddings similar to each other [12, 13].

In the above two kinds of integration approaches, all factors of interest and social embeddings are used for the item recommendation task. Usually, embeddings are learned from complex interaction records and social links that we can observe. As these observed data are generated by a user's own interests, the influence of his/her social circle, or both of them, the learned embeddings contain the user's interest and social information. To illustrate the complexity of information hidden in interactions and social data, we take the scenario in Fig. 1 as an example, where an active user u purchases three items (i_0, i_1, i_2) and has three friends (u_1, u_2, u_3). From the figure, we have the following two observations.

- Entangled interest and social information in interaction records. On the one hand, only the active user u but none of his/her friends purchases item i_2 . These interaction records may reflect u 's own interests. On the other hand, u and some of his/her friends (u_1 and u_3) buy item i_0 and i_1 . These interaction records may be caused by u 's own interests, the influence from his/her friends, or both. Based on these observed interaction data, the generated u 's embedding \mathbf{z}_u^x contains his/her interest and social preferences.
- Entangled interest and social information in social links. User u has a friend u_2 , and u_2 does not buy any items purchased by u . Thus the social relation between u and u_2 is a true friendship and independent of their interests. In addition, u has friends u_1 and u_3 , who buy some items (i_0 and i_1) same to u . Hence, the social relations among u , u_1 and u_3 also contain interest information. Based on these observed social data, interest and social information are also entangled in the generated u 's embedding \mathbf{z}_u^s .

To more clearly illustrate the complex information in interaction records and social relations, we perform an empirical statistical analysis on three real-world datasets in Sect. 4.2.

From the statistical results, we make an interesting observation that interaction records contain a lot of social information, and social relations also comprise a great deal of interest information. Hence, whichever kinds of embeddings learned from these datasets contain entangled interest and social factors.

A high-quality recommendation should recommend what a user really likes according to his/her own interests. So, we argue that it is inappropriate to use such entangled representations for recommendations. From the above analysis, we can see that the key problem is how to extract pure interest factors to form a user representation for the recommendation task. However, the biggest challenge is that it is hard to collect more data to show whether a user's interaction records are influenced by his/her social relations. It is also unrealistic to let users tell us the reason why they created each social link. Fortunately, we can assign semantics to some factors of user representations in the embedding space, and the disentanglement technique is commonly used for this purpose. In the computer vision field [14–16], the Variational AutoEncoder (VAE) [17] allows image representations to be disentangled over factors such as object color, object size or background color. In cross-domain recommendation [18, 19], recommendation performance can be promoted by leveraging information from other domains. Motivated by these works, this paper proposes to use the encoder-decoder framework of VAE to build a new recommendation model called Disentangled Variational autoencoder for Social Recommendation (DVSR).

In the encoding phase of DVSR, we separately encode interaction records and social relations of a user as two embeddings. Then we designate some factors in each of them to represent the user's interest preference and the rest to represent his/her social preference. In the decoding phase of DVSR, all interest factors in the two embeddings are used to reconstruct interaction records and all social factors are used to reconstruct social relations. During the model training, we use interaction records and social relations as the supervised signals to learn to disentangle interest factors and social factors from the two embeddings, which aims to improve the performance of modeling interest and social factors. In order to further disentangle interest and social factors, we employ self-supervised signals to reduce the mutual information between interest and social factors. Finally, the learned factors can be selectively integrated to perform the recommendation task. Here, we only integrate interest factors for the item recommendation task.

The main contributions of this paper are summarized as follows:

- We analyze the entangled information in interaction records and social relations, propose a VAE based model to generate disentangled representations of users and integrate useful factors of these representations for the recommendation task.
- We integrate multiple auxiliary tasks into the model training process to improve the modeling of social factors, promote the learning of disentangled representations and perform the item recommendation task.
- We conduct comprehensive experiments on three benchmark datasets and the experimental results show the validity of our model.

2 Related Work

2.1 Social Recommendation

Most existing social recommendation models directly integrate interest and social information to enhance the recommendation performance. GraphRec [20] unifies the information by

concatenating user representations from the item space and social space. CNSR [21] directly uses an addition pooling operation on two kinds of user embeddings to construct the integrated user representation. DiffNet++ [22] integrates the user embeddings from the social graph and the interest graph through the attention mechanism.

Besides, some works employ social information to indirectly regularize the learning of user representations in the interest domain. DASO [12] separately constructs two adversarial learning tasks in the interest and social domains by mapping user embeddings from one domain to another. SEPT [13] combines user embeddings from the preference view, sharing view and friend view as the supervised signals to supervise the learning of user representations. SoReg [5] regularizes the learning of user representations according to the similarity between users.

These models utilize all factors of user embeddings to integrate information. In contrast, our DVSR explicitly disentangles interest and social factors from embeddings to improve the recommendation performance.

2.2 Variational Autoencoder Based Recommendation

Variational AutoEncoder (VAE) [17] is the variational version of traditional autoencoder which has a more powerful ability in representation learning and feature extraction [23]. AutoRec [24] is the first VAE based model in the recommendation field which sets just one hidden layer in VAE and is designed as Item-based VAE (I-AutoRec) and User-based VAE (U-AutoRec). Multi-VAE [25] is a more advanced VAE based recommendation model that introduces the multinomial likelihood function and illustrates the validity of this function. CDAE [26] is another VAE based recommendation model that focuses on the noise in latent factors. Lee et al. proposed several augmented variational autoencoders with auxiliary social information such as CVAE-CF and JVAE-CF [27].

In this work, we take advantage of VAE's strong ability to explicitly model interest and social effects, and build a hybrid VAE model based on interest and social data.

2.3 Disentangled Representation Based Recommendation

Liu et al. [28] and Zhao et al. [29] individually disentangled biased and unbiased factors for recommendations. Nema et al. [30] suggested separating different semantics from user embeddings. Ma et al. [31] proposed to categorize interaction records distinctly.

Disentangled representation learning has also been introduced to the social recommendation task. For example, DcRec [32] considers the heterogeneous behavioral patterns between the interest and social domains, performs domain disentangling operation in the form of GNN [33] and utilizes data augmentation on two disentangled domain views to perform cross-domain and domain-specific contrastive learning in model training.

Besides, some studies try to capture motivations about users' consumption and social behaviors by disentangling embeddings at the facet level. DMJP [34] separately disentangles embeddings of users into multiple facets to explain users' actions in the interest and social domains and utilizes the attention mechanism to realize the refinement of the representations. DSR [35] disentangles user representations from the facet-to-facet perspective and designs an independence regularization loss to ensure the validity of embeddings from different facets. DISGCN [36] disentangles not only users but also items, but unlike the flexible facet setting of DMJP and DSR, the number of facets is limited by the number of graphs.

Our model designs a unique disentanglement strategy from both the cross-domain and domain-specific perspectives, and critiques semantic information for disentangled factors.

3 Variational Autoencoder

VAE is a classical Collaborative Filtering (CF) inference model in the recommendation field [17, 37]. As a generative model, VAE first extracts the semantic information hidden in observed items of users and compresses it into a low-dimensional space, and then provides recommended items to users by analyzing the low-dimensional representations. A VAE model has two core components, an encoder and a decoder. Now we introduce the two components and some necessary notations used in this paper.

3.1 Notations

The user-item interactions can be represented as a matrix $\mathbf{X} \in \mathcal{R}^{|\mathcal{U}| \times |\mathcal{I}|}$, where \mathcal{U} represents the set of users, \mathcal{I} is the set of items, and $|\mathcal{U}|$ and $|\mathcal{I}|$ denote the number of users and items, respectively. Let $u \in \mathcal{U}$ be a particular user, $i \in \mathcal{I}$ be a particular item, and x_{ui} be the implicit feedback of user u on item i which is an entry in the matrix \mathbf{X} . Specially, if user u has an interaction with item i , the corresponding entry $x_{ui} = 1$, otherwise $x_{ui} = 0$.

3.2 Encoder

The encoder of VAE is responsible for encoding each input as an embedding. A VAE model is commonly trained using a bag-of-words representation: each user u is represented by items he/she has interacted with, i.e., row \mathbf{x}_u of matrix \mathbf{X} [30]. After the model learning, the encoder provides a distribution where the embedding representation \mathbf{z}_u of user u is sampled. In other words, by feeding each user's item list to the encoder, we obtain a low-dimensional representation for each user. Statistically speaking, the representations of all users should follow a normal distribution and the distribution is directly influenced by the interactions of these users. Hence the true distribution from which \mathbf{z}_u is sampled is approximated by a parameterized function:

$$q_\phi(\mathbf{z}_u | \mathbf{x}_u) = \mathcal{N}(\mu_\phi(\mathbf{x}_u), \text{diag}(\sigma_\phi(\mathbf{x}_u))), \quad (1)$$

where ϕ is the parameter set in the encoder component, $\mu_\phi(\mathbf{x}_u)$ and $\sigma_\phi(\mathbf{x}_u)$ are the mean value and standard deviation of the normal distribution, respectively. Also, due to the stochastic nature of \mathbf{z}_u , the model can not compute the gradient by direct backpropagation algorithm to optimize the parameters. Therefore, VAE employs the re-parametrization trick in model training.

3.3 Decoder

The decoder of VAE takes embedding representations as input. The decoder can analyze the user representation and select appropriate items that the user is likely to buy. By feeding user representation \mathbf{z}_u to the decoder, it generates the logits $f_\theta^{\text{dec}}(\mathbf{z}_u)$ of user u . To facilitate model optimization, the logits are usually transformed as:

$$\pi(\mathbf{z}_u) \propto \exp(f_\theta^{\text{dec}}(\mathbf{z}_u)), \quad (2)$$

where θ is the parameter set in the decoder component. Then the probability distribution over all items can be obtained for user u :

$$p_{\theta}(\mathbf{x}_u | \mathbf{z}_u) = \sum_i x_{ui} \log \pi_i(\mathbf{z}_u). \quad (3)$$

In the top- N recommendation task, ideal items are the top N items after sorting in descending order. To rank interested items of users higher in the recommendation list, VAE maximizes the likelihood function in the learning process. Simultaneously, to avoid the overfitting, VAE regularizes the model learning by constraining the distribution $q_{\phi}(\mathbf{z}_u | \mathbf{x}_u)$. Since the prior distribution $p(\mathbf{z}_u)$ of \mathbf{z}_u is supposed to follow the normal distribution, VAE hopes that the posterior distribution can follow the same distribution as the prior distribution as much as possible. The KL divergence can reflect the difference between the two distributions, so the usual objective function of VAE is defined as:

$$L(\mathbf{x}_u, \phi, \theta) = \mathbb{E}_{q_{\phi}(\mathbf{z}_u | \mathbf{x}_u)}[\log p_{\theta}(\mathbf{x}_u | \mathbf{z}_u)] - \text{KL}(q_{\phi}(\mathbf{z}_u | \mathbf{x}_u) | p(\mathbf{z}_u)). \quad (4)$$

4 Preliminaries

4.1 Notations Relevant to Social Relations

In addition to the notations mentioned in Sect. 3, we also introduce some notations related to user-user relations. We denote user-user relations by $\mathbf{S} \in \mathbb{R}^{|\mathcal{U}| \times |\mathcal{U}|}$ and let s_{ut} represent social links of user u and s_{ut} be a particular entry in the matrix \mathbf{S} . If user u has a connection with another user t , we have $s_{ut} = 1$, otherwise $s_{ut} = 0$. Furthermore, to distinguish parameters in the interest and social domains, we use x and s to decorate interest parameters and social parameters, respectively.

4.2 Empirical Analysis of Entangled Information

To illustrate the necessity of our proposed model, we investigate the degree of information entanglement in the interest and social domains on different datasets. Our investigation is highly relevant to the following two Research Questions (RQ):

- RQ1: How to measure the degree of information entanglement or the amount of the entangled information in the interest and social domains, respectively?
- RQ2: How does measuring entangled information help us?

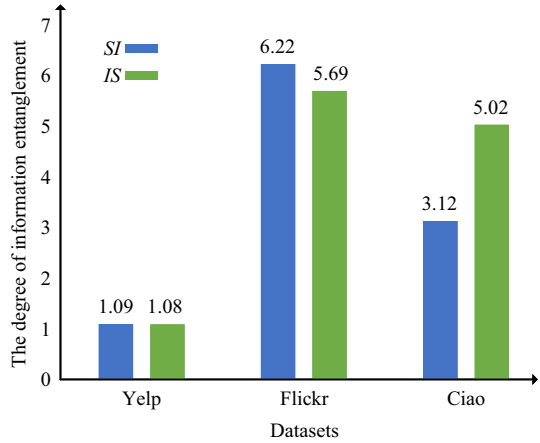
RQ1. As mentioned in Sect. 1, the reason that entangled social information exists in the interest domain is the common interactions of users and their friends. In order to quantify the entangled information, we first get the friends who have purchased the same items by:

$$\mathbf{S}' = (\mathbf{X}\mathbf{X}^T) \odot \mathbf{S}. \quad (5)$$

Then we count the number of these friends by:

$$SI = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \sum_{t \in \mathcal{U}} g(s'_{ut}), s'_{ut} \in \mathbf{S}', \quad (6)$$

Fig. 2 Empirical statistics of entangled information



where SI is the estimated amount of Social information in the Interest domain and can indirectly reflect the degree of information entanglement in that domain, and

$$g(x) = \begin{cases} 0 & x = 0 \\ 1 & x \neq 0. \end{cases} \tag{7}$$

Similarly, we estimate the entangled Interest information in the Social domain by:

$$IS = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}} g(x'_{ui}), x'_{ui} \in (\mathbf{S}\mathbf{X}) \odot \mathbf{X}. \tag{8}$$

RQ2. We calculate SI and IS on three popular social recommendation datasets: Yelp, Flickr and Ciao. The details of these datasets can be found in Sect. 6.1.1, and their SI and IS are shown in Fig. 2. From Fig. 2, we can observe that SI and IS of Yelp are the lowest among all datasets, which means that Yelp contains less entangled information than the other two datasets. Flickr is the highest among three datasets in two metrics, hence Flickr contains the most entangled information. To better understand our work, we explicitly state our hypothesis: *Different semantic information exists in both interest and social domains, thereby the factors of representations learned from these two domains entangle different information.* This hypothesis is the basis of our proposed DVSR model.

5 Methodology

In this section, we introduce the proposed DVSR model which is based on the encoder-decoder framework of VAE and has two stages. Its overall architecture is illustrated in Fig. 3, where the target user u is taken as an example for a better understanding. In the first stage, we take the interaction records and social relations of u as input and encode them as two embeddings. Then, we disentangle interest and social factors from the two embeddings. In the second stage, we pass integrated interest factors through an interest decoder to reconstruct interaction records and pass integrated social factors through a social decoder to reconstruct social relations. Furthermore, the reconstruction loss is used to learn for recommendation and two self-supervised losses, intra-domain and inter-domain disentanglement losses, are used to encourage the disentanglement of interest and social factors.

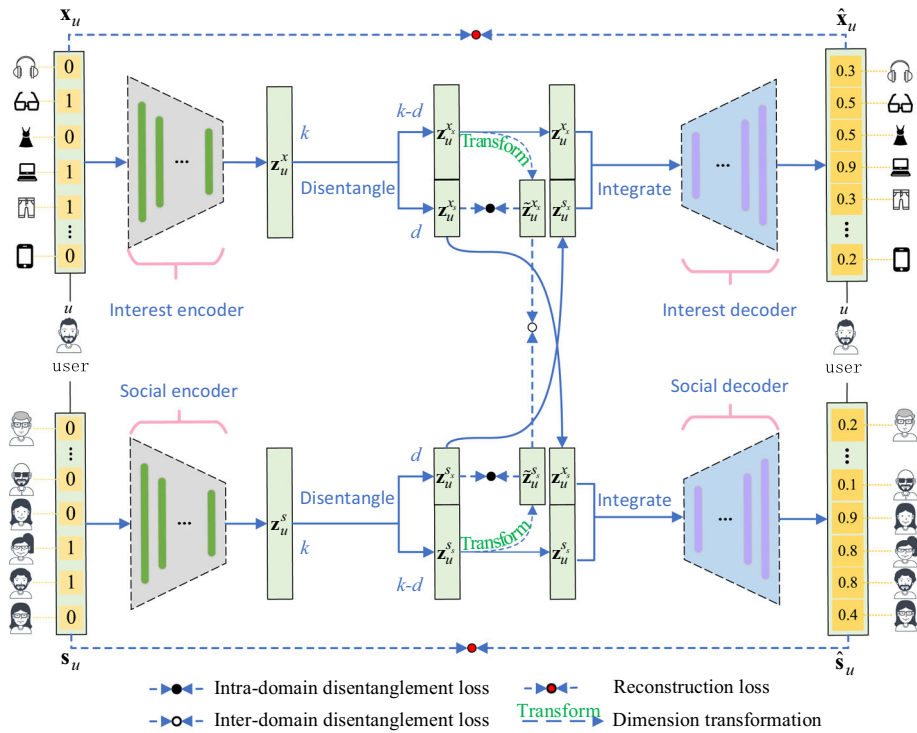


Fig. 3 Overall architecture of DVSR

5.1 Encoding for Disentanglement

In this stage, our model focuses on how to compress the information from user-item interactions and user-user relations into a low-dimensional space and how to extract disentangled information from these embeddings. In a traditional VAE, the encoder is responsible for producing user representations that are used directly in the subsequent decoder. However, in our model, embedding representations encoded by the traditional encoder are only a basic version of user representations (as shown in Fig. 3), which need to be further processed to realize the ideal integration.

To encode user interest and social profiles separately, we design two encoders at this stage, an interest encoder and a social encoder. For the interest encoder q_{ϕ^x} , we take user u 's interaction records \mathbf{x}_u as its input and use Eq. (1) to get the distribution of his/her interest latent factors $q_{\phi^x}(\mathbf{z}_u^x|\mathbf{x}_u)$. Similarly, for the social encoder q_{ϕ^s} , by employing user u 's social relations \mathbf{s}_u in Eq. (1), we obtain the distribution of social latent factors $q_{\phi^s}(\mathbf{z}_u^s|\mathbf{s}_u)$.

Now the user's representations in the interest domain and the social domain can be sampled from the above two distributions, and we denote them separately by \mathbf{z}_u^x and \mathbf{z}_u^s which are two k -dimensional embeddings. However, these user representations are only basic versions, in which a lot of entangled information exists. So it is necessary to disentangle them to extract different semantic information. We divide these embeddings into two non-overlapping subsets, and critique them as interest semantics and social semantics, respectively. Note that these semantic subsets will be further learned during model training. Let $\mathbf{z}_u^{x_i}$ and $\mathbf{z}_u^{x_s}$ separately

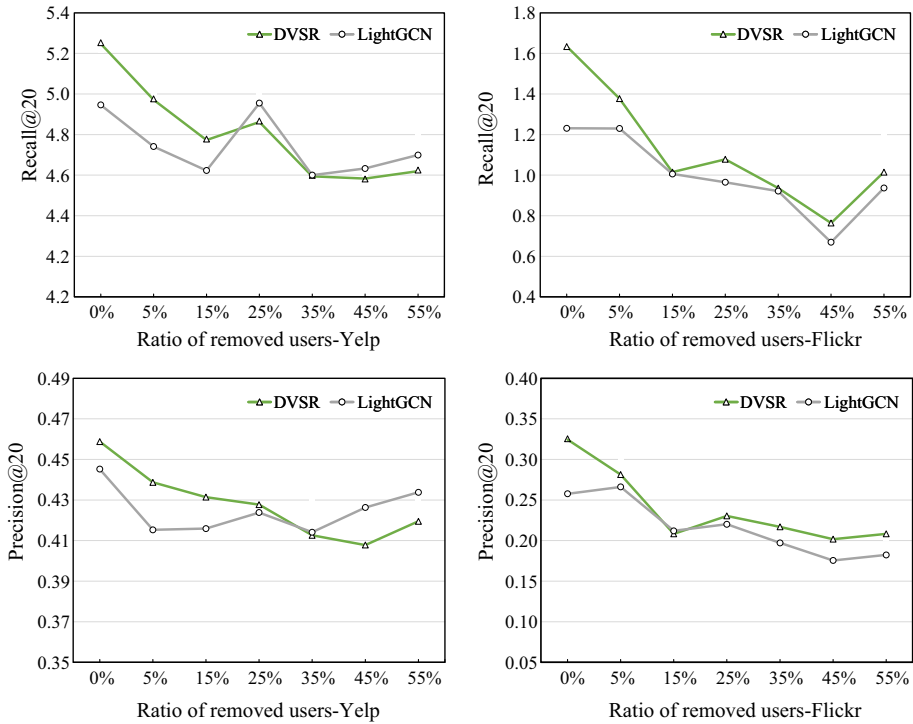


Fig. 4 The results of extensive experiment

represent interest and social information in \mathbf{z}_u^x . So do $\mathbf{z}_u^{s_x}$ and $\mathbf{z}_u^{s_s}$ in \mathbf{z}_u^s . Here, $\mathbf{z}_u^{x_s}$ and $\mathbf{z}_u^{s_x}$ are selected as cross-domain factors. As shown in Fig. 3, an alternative simple strategy is to let the first $k - d$ ($k > d > 0$) factors of \mathbf{z}_u^x be $\mathbf{z}_u^{x_x}$ and the left d factors be $\mathbf{z}_u^{x_s}$, and let the first d factors of \mathbf{z}_u^s be $\mathbf{z}_u^{s_x}$ and the left $k - d$ factors be $\mathbf{z}_u^{s_s}$. So, we have

$$\mathbf{z}_u^x = [\mathbf{z}_u^{x_x} ; \mathbf{z}_u^{x_s}], \tag{9}$$

$$\mathbf{z}_u^s = [\mathbf{z}_u^{s_x} ; \mathbf{z}_u^{s_s}], \tag{10}$$

where d is the number of cross-domain factors from the two domains. In this paper, we adopt the strategy to discriminate the different semantic information in an embedding.

5.2 Decoding for Recommendation

The above obtained $\mathbf{z}_u^{x_x}$, $\mathbf{z}_u^{x_s}$, $\mathbf{z}_u^{s_x}$ and $\mathbf{z}_u^{s_s}$ can be divided into two categories according to their semantic information. One reflects the user’s interest information and includes $\mathbf{z}_u^{x_x}$ and $\mathbf{z}_u^{s_x}$. The other is composed of $\mathbf{z}_u^{x_s}$ and $\mathbf{z}_u^{s_s}$, and embodies the user’s social information. Then factors with the same semantic information can be utilized to generate the final user representations for the interest and social profiles. We simply integrate them by concatenating operation as $[\mathbf{z}_u^{x_x} ; \mathbf{z}_u^{s_x}]$ and $[\mathbf{z}_u^{x_s} ; \mathbf{z}_u^{s_s}]$. As a result, we exchange some factors from the two autoencoders, and we call these exchanged factors (i.e., $\mathbf{z}_u^{x_s}$ and $\mathbf{z}_u^{s_x}$) cross-domain vectors.

To reconstruct interaction records and social relations of users separately, we design two corresponding decoders, namely an interest decoder and a social decoder. For the interest

decoder p_{θ^x} , we take the final user interest representation $[\mathbf{z}_u^{x_x} : \mathbf{z}_u^{s_x}]$ as its input, and use Eq. (3) to get the distribution of his/her interest items $p_{\theta^x}(\mathbf{x}_u | [\mathbf{z}_u^{x_x} : \mathbf{z}_u^{s_x}])$. Similarly, for the social decoder p_{θ^s} , its input is $[\mathbf{z}_u^{s_s} : \mathbf{z}_u^{x_s}]$ and it generates the distribution of social friends $p_{\theta^s}(\mathbf{s}_u | [\mathbf{z}_u^{s_s} : \mathbf{z}_u^{x_s}])$.

5.3 Model Training

In this section, we introduce three losses and the final objective function used to train our model. To disentangle embedding representations from the interest and social domains, we design two losses for the intra-domain disentanglement and inter-domain disentanglement, respectively, which improves the effectiveness of the disentangling operation. Since the training process of the two VAEs in the model depends on each other, we design a unified reconstruction loss to optimize them jointly.

5.3.1 Intra-domain Disentanglement Loss

As $\mathbf{z}_u^{x_x}$ and $\mathbf{z}_u^{x_s}$ are extracted from the interaction records of user u , they may carry some similar information if there are not any special constraints. The same is true for $\mathbf{z}_u^{s_x}$ and $\mathbf{z}_u^{s_s}$, since they are extracted from the user's social relations. However, to keep the validity of disentangling operations, we hope that disentangled interest factors and disentangled social factors from an embedding are discrepant in the latent space. That is to say, $\mathbf{z}_u^{x_x}$ and $\mathbf{z}_u^{s_s}$ are expected to be as far away as possible from $\mathbf{z}_u^{x_s}$ and $\mathbf{z}_u^{s_x}$, respectively. For this purpose, firstly, we separately use a linear projection on $\mathbf{z}_u^{x_x}$ and $\mathbf{z}_u^{s_s}$ to make them have the same dimension with $\mathbf{z}_u^{x_s}$ and $\mathbf{z}_u^{s_x}$:

$$\tilde{\mathbf{z}}_u^{x_x} = \mathbf{w}_x \mathbf{z}_u^{x_x}, \tag{11}$$

$$\tilde{\mathbf{z}}_u^{s_s} = \mathbf{w}_s \mathbf{z}_u^{s_s}, \tag{12}$$

where $\mathbf{w}_x, \mathbf{w}_s \in \mathbb{R}^{d \times (k-d)}$ are dimension transformation matrices. These two transformation matrices are also parameters to be optimized in our model, so that they can adaptively accomplish the transfer of semantic information in the transformation process. Then, we define the intra-domain disentanglement loss $\mathcal{L}_{\text{intra}}$ as:

$$\mathcal{L}_{\text{intra}} = \frac{\xi(\mathbf{z}_u^{x_s}, \tilde{\mathbf{z}}_u^{x_x})}{\tau} + \frac{\xi(\mathbf{z}_u^{s_x}, \tilde{\mathbf{z}}_u^{s_s})}{\tau}, \tag{13}$$

where $\xi(\cdot) : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$ is a function that takes two vectors as input and outputs the agreement score between them, and τ is a hyper-parameter scaling the similarity from $[-1, 1]$ to $[-1/\tau, 1/\tau]$. In this paper, we simply take the cosine function as the estimator $\xi(\cdot)$. Essentially, Eqs. (11) and (12) employ the linear projection to transform one of the two disentangled parts of an embedding, and Eq. (13) calculates the similarity between the transformed part and the untransformed part. In order to encourage the disentanglement, we minimize $\mathcal{L}_{\text{intra}}$ to reduce their similarity.

5.3.2 Inter-domain Disentanglement Loss

As well known, different semantic factors extracted from different domains should be as different as possible. For example, interest factors of the interest domain should be far away from social factors of the social domain. For this purpose, we should design a disentanglement

loss. Since the transferred vectors $\tilde{\mathbf{z}}_u^{x_x}$ and $\tilde{\mathbf{z}}_u^{s_s}$ represent the essential semantic information in their original interest and social domains and are not subject to the cross-domain operations, we construct their inter-domain disentanglement loss $\mathcal{L}_{\text{inter}}$ by:

$$\mathcal{L}_{\text{inter}} = \log \frac{\exp(\xi(\tilde{\mathbf{z}}_u^{s_s}, \tilde{\mathbf{z}}_u^{x_x})/\delta)}{\sum_{t \in \mathcal{U}} \exp(\xi(\tilde{\mathbf{z}}_u^{s_s}, \tilde{\mathbf{z}}_t^{x_x})/\delta)}, \tag{14}$$

where δ is a hyper-parameter that scales the similarity from $[-1, 1]$. Essentially, minimizing Eq. (14) can reduce the mutual information between $\tilde{\mathbf{z}}_u^{x_x}$ and $\tilde{\mathbf{z}}_u^{s_s}$, which encourages the disentanglement of the two parts. By rewriting Eq. (14), we can derive

$$\mathcal{L}_{\text{inter}} = \frac{\xi(\tilde{\mathbf{z}}_u^{s_s}, \tilde{\mathbf{z}}_u^{x_x})}{\delta} - \log\left(\sum_{t \in \mathcal{U}} \exp\left(\frac{\xi(\tilde{\mathbf{z}}_u^{s_s}, \tilde{\mathbf{z}}_t^{x_x})}{\delta}\right)\right). \tag{15}$$

From the first term of Eq. (15), we can see that the similar strategy of Eq. (13) is also adopted to ensure the inter-domain disentanglement results. In addition, the second term can be seen as a sampling strategy that constrains $\tilde{\mathbf{z}}_u^{x_x}$ and $\tilde{\mathbf{z}}_u^{s_s}$ to avoid the overfitting.

5.3.3 Reconstruction Loss

From Fig. 3, we can see that interest and social VAEs are dependent on each other. So we build different reconstruction losses according to their own structure.

In the item recommendation process, the final user interest representation $[\mathbf{z}_u^{x_x} : \mathbf{z}_u^{s_x}]$ makes the recommendation process be under the effect of two distributions. By extending the first term of Eq. (4), we have the first reconstruction loss of DVSR as:

$$\mathcal{L}_a(\mathbf{x}_u, \mathbf{s}_u, \phi^x, \phi^s, \theta^x) = \mathbb{E}_{q_{\phi^x}([\mathbf{z}_u^{x_x} : \mathbf{z}_u^{x_s}]|\mathbf{x}_u), q_{\phi^s}([\mathbf{z}_u^{s_x} : \mathbf{z}_u^{s_s}]|\mathbf{s}_u)} [\log p_{\theta^x}(\mathbf{x}_u | [\mathbf{z}_u^{x_x} : \mathbf{z}_u^{s_x}])]. \tag{16}$$

In addition, we have the subset of factors $\mathbf{z}_u^{x_s}$ disentangled from interest embeddings and the subset of factors $\mathbf{z}_u^{s_s}$ disentangled from social embeddings. They can be used to reconstruct the social relations of user u . So the second reconstruction loss of DVSR can be represented as:

$$\mathcal{L}_b(\mathbf{x}_u, \mathbf{s}_u, \phi^x, \phi^s, \theta^s) = \mathbb{E}_{q_{\phi^x}([\mathbf{z}_u^{x_x} : \mathbf{z}_u^{x_s}]|\mathbf{x}_u), q_{\phi^s}([\mathbf{z}_u^{s_x} : \mathbf{z}_u^{s_s}]|\mathbf{s}_u)} [\log p_{\theta^s}(\mathbf{s}_u | [\mathbf{z}_u^{s_x} : \mathbf{z}_u^{x_s}])]. \tag{17}$$

Besides the two reconstruction losses, we should keep the rationality of approximate distribution. As shown in the second term of Eq. (4), the KL divergence is used to measure the similarity between the approximate distribution and the true distribution of embedding representations. Under the assumption that the prior distribution of each factor in embedding representations always follows the normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, we use this distribution as the true distribution in our training to supervise the learning of approximate distributions and define the KL divergence by:

$$\mathcal{L}_{\text{KL}} = -\text{KL}(q_{\phi^x}([\mathbf{z}_u^{x_x} : \mathbf{z}_u^{x_s}]|\mathbf{x}_u)|\mathcal{N}(\mathbf{0}, \mathbf{I})) - \text{KL}(q_{\phi^s}([\mathbf{z}_u^{s_x} : \mathbf{z}_u^{s_s}]|\mathbf{s}_u)|\mathcal{N}(\mathbf{0}, \mathbf{I})). \tag{18}$$

For the convenience of tuning parameters, we can unify the above three losses and represent the unified reconstruction loss of DVSR as:

$$\mathcal{L}_{\text{rec}} = \mathcal{L}_a + \mathcal{L}_b + \mathcal{L}_{\text{KL}}. \tag{19}$$

5.3.4 Objective Function

The final objective function consists of the three losses above:

$$\mathcal{L}_{\text{DVSR}} = \sum_{u \in \mathcal{U}} -\mathcal{L}_{\text{rec}} + \alpha \mathcal{L}_{\text{intra}} + \beta \mathcal{L}_{\text{inter}}, \quad (20)$$

where α and β are two parameters controlling the strength of the intra-domain disentanglement and the inter-domain disentanglement, respectively. We minimize the objective function to train model parameters. The training process of DVSR is given in Algorithm 1.

Algorithm 1: Training process of DVSR.

Input: user-item interaction matrix \mathbf{X} , user-user relation matrix \mathbf{S} , model parameters $\phi^x, \phi^s, \theta^x, \theta^s, \mathbf{w}^x$ and \mathbf{w}^s .

Randomly initialize model parameters;

for each iteration **do**

for $u \in \mathcal{U}$ **do**

Get $q_{\phi^x}(\mathbf{z}_u^x | \mathbf{x}_u)$ and $q_{\phi^s}(\mathbf{z}_u^s | \mathbf{s}_u)$ according to the two encoders;

Sample \mathbf{z}_u^x and \mathbf{z}_u^s from $q_{\phi^x}(\mathbf{z}_u^x | \mathbf{x}_u)$ and $q_{\phi^s}(\mathbf{z}_u^s | \mathbf{s}_u)$ by the reparameterization trick, respectively;

Obtain \mathbf{z}_u^{xx} and \mathbf{z}_u^{xs} from \mathbf{z}_u^x using Eq. (9);

Get \mathbf{z}_u^{sx} and \mathbf{z}_u^{ss} from \mathbf{z}_u^s using Eq. (10);

Obtain $\tilde{\mathbf{z}}_u^{xx}$ and $\tilde{\mathbf{z}}_u^{ss}$ using Eqs.(11) and (12);

Calculate $\mathcal{L}_{\text{intra}}$ and $\mathcal{L}_{\text{inter}}$ using Eqs.(13) and (14);

Send $[\mathbf{z}_u^{xx} : \mathbf{z}_u^{xs}]$ and $[\mathbf{z}_u^{sx} : \mathbf{z}_u^{ss}]$ to the interest decoder and social decoder, respectively;

Obtain the distribution $p_{\theta^x}(\mathbf{x}_u | [\mathbf{z}_u^{xx} : \mathbf{z}_u^{xs}])$ and $p_{\theta^s}(\mathbf{s}_u | [\mathbf{z}_u^{sx} : \mathbf{z}_u^{ss}])$ according to the two decoders;

Calculate the VAE loss using Eq. (19);

end

Compute the total loss of all users in this iteration by Eq. (20);

Update model parameters using the Adam optimization method;

end

Output: optimized model parameters.

6 Experiments

In this section, we show relevant experimental results and explain the model performance according to the characteristics of our model.

6.1 Experimental Setup

6.1.1 Datasets

Our experiments are conducted on three benchmark datasets: Yelp [22], Flickr [38] and Ciao [39]. All three datasets were crawled down from their respective websites with the same name and can be publicly downloaded (Yelp and Flickr,¹ and Ciao²).

¹ <https://github.com/PeiJieSun/diffnet>.

² <https://github.com/Coder-Yu/QRec>.

Table 1 Statistics of datasets

Dataset	#Users	#Items	#Ratings	#Links	Rating density	Link density
Yelp	17,237	38,342	204,448	143,765	0.03%	0.05%
Flickr	8,358	82,120	327,815	187,273	0.05%	0.27%
Ciao	7,317	104,975	283,319	111,781	0.04%	0.21%

- **Yelp** is one of the largest review sites in the United States, where people can rate the services of some locations (e.g., restaurants, cinemas, etc.) and social relations are built directly from users' friend lists.
- **Flickr** is one of the largest social image sharing platforms, where users can share their preferences for images and videos with their social followers and also can follow the people they are interested in.
- **Ciao** is an online shopping site, where millions of products or services are critically reviewed or rated for the benefit of other consumers and people can build their social circles based on the attention of their reviews.

The statistics of these datasets are summarized in Table 1. Our experimental task is a top- N recommendation, thus we convert all explicit ratings to implicit ratings and remove repeated ratings. Specifically, as long as the active user rates one item, we treat that item as positive feedback for the user. Finally, we randomly select 10% of the rating data as the test set, while the remaining data is used as the training set.

6.1.2 Evaluation Metrics

To evaluate the performance of our model and baselines, three common metrics, Precision@ N , Recall@ N and NDCG@ N , are adopted for the top- N recommendation task. By default, we set $N = 20$, report the average metrics for all users in the test set [40], and omit the percent sign of model performance in all tables.

6.1.3 Baselines

Two recent VAE based social recommendation models and a recent disentangled learning based social recommendation model are compared with DVSR to validate its effectiveness.

- **CVAE-CF** [27] is a VAE based social recommendation model. This method considers the social information as a specific condition that can influence the recommendation result, and models a conditional distribution of users' responses to items considering the social links of users.
- **JVAE-CF** [27] builds two VAEs for the item recommendation task and the friend recommendation task. It attempts to model a joint distribution of users' responses to items and persons by using social relations.
- **DcRec** [32] is a recent graph neural network based social recommendation model. It builds multiple contrastive learning tasks to induce disentangled representation learning.

6.1.4 Experiment Settings

For a fair comparison, all models are optimized using the Adam optimizer [41] and initialized in the same manner. The batch size is set to 2,048, the learning rate is tuned in

$\{1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$, and the maximum training epoch is set to 100. For DcRec, we follow its original paper to set the hyperparameter. For all architectures of VAE based models, we build them on the MultiLayer Perceptron (MLP) architecture. Their input and output layers coincide with the number of items or users, which depends specifically on the task VAE is performing. Specifically, except for the input and output layers, the hidden layers of the encoder have the same architecture: $600 \rightarrow 200 \rightarrow 128$. So for DVSR, k is 128. We use one layer perceptron as the decoder. To investigate the performance of different architectures of VAE based models, we keep most of the critical tricks consistent for these models. Our experiments are implemented in PyTorch.

Based on the MLP architecture, the computational complexity of DVSR mainly originates from its encoding and decoding stages, which are closely related to the number of layers and neurons in the neural network. Let d_l be the perceptron number of the l -th hidden layer. For a single user, the complexity of the interaction side on DVSR is $\mathcal{O}(|\mathcal{V}|d_1 + \sum_{l=2}^L d_{l-1}d_l + |\mathcal{V}|d_L)$, and the complexity of the social side is $\mathcal{O}(|\mathcal{U}|d_1 + \sum_{l=2}^L d_{l-1}d_l + |\mathcal{U}|d_L)$. Furthermore, the complexity involving Eqs. (11) and (12) is $\mathcal{O}(2d(k-d))$. Therefore, the total complexity of DVSR approximates $\mathcal{O}(|\mathcal{U}|(|\mathcal{V}| + |\mathcal{U}|))$.

6.2 Overall Performance Comparison

The overall performance of DVSR and baselines is given in Table 2. From the table, we have the following observations:

- Our DVSR achieves the best performance** DVSR consistently outperforms all baselines across Precision, Recall and NDCG metrics. Compared to the best baseline, the improvement of DVSR is 2.94%–7.87% and 4.15%–7.32% on Yelp and Ciao, respectively. Especially, on Flickr, the DVSR improvement is 11.17%–25.38%, which can be considered to be significant. As mentioned in Sect. 4.2, Flickr contains the most entangled information among the three datasets. Therefore, we can attribute the improvement of DVSR to its excellent ability to handle entangled information.
- It is essential to disentangle the interest and social semantic information from both interactions and social behaviors** In CVAE-CF, JVAE-CF and DcRec, the interest information is extracted only from interaction data, and the social semantic information is extracted only from social relation data. CVAE-CF and JVAE-CF combine both kinds of information to generate recommendations. The difference between them lies in the way the extracted social information is used. CVAE-CF only uses social information for the recommendation, but JVAE-CF also uses social information to infer social links. As the social information of JVAE-CF is not disentangled for its two purposes, JVAE-CF performs worse than CVAE-CF. Although DcRec only uses the interest information for the recommendation, it maximizes the similarity between the interest and social information of the same user. As a result, the social semantics may be weakened and the performance is degraded. Our DVSR can improve the ability to disentangle interest factors from the interest and social information both of which are entangled in interaction records and social relations.
- Explicit disentanglement losses are necessary** The disentangling operations of CVAE-CF, JVAE-CF and DcRec are essentially to model interest and social semantic information, respectively. They lack an explicit disentanglement loss to distinguish between the two kinds of information. Our DVSR beats them by considering the explicitly different semantic information between the interest and social domains, and the inner heterogeneity of the information existing in the two domains.

Table 2 Overall performance

Models	Yelp		Flickr		Ciao	
	Precision	Recall	Precision	Recall	Precision	Recall
CVAE-CF	<i>0.4457</i>	<i>4.994</i>	<i>0.2919</i>	<i>1.304</i>	<i>1.420</i>	<i>6.792</i>
JVAE-CF	0.4234	4.766	0.2748	1.173	1.413	6.769
DcRec	0.4253	4.693	0.2646	1.186	1.297	6.543
DVSR	0.4588	5.253	0.3245	1.635	1.479	7.206
Improv	2.94	5.19	11.17	25.38	4.15	6.10
		NDCG		NDCG		NDCG
		2.115		0.7340		4.287
		2.000		0.6863		4.268
		2.122		0.7741		4.226
		2.289		0.9119		4.601
		7.87		17.80		7.32

The best and the second-best results are in bold and italics, respectively

Table 3 Performance of DVSR with respect to different values of d

d	Yelp			Flickr			Ciao		
	Precision	Recall	NDCG	Precision	Recall	NDCG	Precision	Recall	NDCG
112	0.4452	5.023	2.214	0.3109	1.504	0.8041	1.413	6.729	4.176
96	0.4501	5.064	2.237	0.3262	1.555	0.8698	1.449	6.882	4.346
64	0.4549	5.103	2.285	0.3245	1.635	0.9119	1.479	7.206	4.601
32	0.4540	5.178	2.256	0.3287	1.498	0.8414	1.463	7.041	4.450
16	0.4588	5.253	2.289	0.3125	1.457	0.8196	1.469	7.092	4.431

6.3 Parameter Analysis

We conduct the sensitivity experiments on the three hyper-parameters of our model. The d is the number of cross-domain factors, and α and β control the strength of intra-domain disentanglement and inter-domain disentanglement, respectively. The sensitivity experiments of the hyper-parameters are given in Tables 3, 4 and 5.

From Table 3, we can see that on Flickr and Ciao, almost all three metrics reach their peaks at $d = 64$, but the model achieves the best performance on Yelp at $d = 16$. We think the sensitivity of d is strongly influenced by the complex semantic information in the two domains. Theoretically speaking, the cross-domain factors from the interest domain should carry social information. The number of these factors should be positively correlated with the amount of social information. Similarly, the number of cross-domain factors from the social domain should have a positive correlation with the amount of interest information in the social domain. Meanwhile, by analyzing the result of the statistics in Fig. 2, we find that it is consistent with our hypothesis. Yelp has less entangled information and DVSR achieves better performance when d is assigned to a smaller value. Flickr and Ciao have more entangled information than Yelp and DVSR achieves better performance when d is assigned to a larger value. In conclusion, we suggest setting a larger d value for datasets containing a lot of entangled information and a smaller d value for datasets with less entangled information.

From Table 4, we can observe that DVSR has a similar sensitivity to α on all datasets, which illustrates the validity of the intra-domain disentanglement and again proves the existence of entangled semantic information. It is worth noting that intra-domain disentanglement plays a very important role in the disentanglement stage, so we should tune α rather carefully. If the disentangling strength is too small, cross-domain factors may incorporate residual entangled information, which will hinder the integration of information. If the disentangling strength is too large, the quality of cross-domain factors will also deteriorate and thus the performance of the model will degrade.

From Table 5, it can be seen that DVSR has different sensitivities to β on different datasets. We find that on Yelp, Flickr and Ciao, almost all three metrics reach their peaks when β is set to 0.1, 0.01 and 0.03, respectively. The model performs best on Yelp when β is tuned around a larger value, but it achieves better performance on Ciao and Flickr when β is tuned around a small value. The two types of disentanglement work together to ensure the effect of the disentanglement. The intra-domain disentanglement has done most of the disentanglement, and the inter-domain disentanglement operation is done to further decouple the remaining entangled information. Yelp has less entangled information, so setting β to a small value will make it difficult to produce effective effects. On Ciao and Flickr, DVSR is very sensitive to β because there is a lot of entangled information in the ratings and links.

Table 4 Performance of DVSR with respect to different values of α

α	Yelp			Flickr			Ciao		
	Precision	Recall	NDCG	Precision	Recall	NDCG	Precision	Recall	NDCG
0.01	0.4457	4.986	2.199	0.2581	1.202	0.6955	1.444	6.892	4.374
0.02	0.4467	5.055	2.235	0.2817	1.228	0.7061	1.444	6.899	4.402
0.05	0.4579	5.134	2.244	0.2979	1.427	0.7763	1.454	6.943	4.466
0.1	0.4588	5.253	2.289	0.3167	1.427	0.8093	1.479	7.206	4.601
0.2	0.4602	5.119	2.237	0.3245	1.635	0.9119	1.475	7.107	4.555
0.5	0.4244	4.725	2.080	0.3048	1.428	0.8400	1.475	7.054	4.504

Table 5 Performance of DVSR with respect to different values of β

β	Yelp			β			Flickr			β			Ciao		
	Precision	Recall	NDCG	Precision	Recall	NDCG	Precision	Recall	NDCG	Precision	Recall	NDCG	Precision	Recall	NDCG
0.01	0.4569	5.166	2.268	0.01	1.635	0.9119	0.3245	1.635	0.9119	1.444	7.078	4.521	1.444	7.078	4.521
0.03	0.4533	5.136	2.253	0.02	1.589	0.8968	0.3185	1.589	0.8968	1.474	7.018	4.594	1.474	7.018	4.594
0.05	0.4515	5.117	2.272	0.03	1.561	0.8893	0.3159	1.561	0.8893	1.479	7.206	4.601	1.479	7.206	4.601
0.1	0.4588	5.253	2.289	0.04	1.561	0.8818	0.3381	1.561	0.8818	1.480	7.219	4.596	1.480	7.219	4.596
0.3	0.4651	5.237	2.278	0.05	1.555	0.8771	0.3125	1.555	0.8771	1.472	7.204	4.583	1.472	7.204	4.583
0.5	0.4588	5.187	2.266	0.06	1.490	0.8630	0.3099	1.490	0.8630	1.468	7.179	4.564	1.468	7.179	4.564

6.4 Study of Disentanglement and Integration

As is shown in Fig. 3, for a given user u , we disentangle interest factors (\mathbf{z}_u^{xx} and \mathbf{z}_u^{sx}) and social factors (\mathbf{z}_u^{ss} and \mathbf{z}_u^{xs}) from the entangled semantic information. Then, the integration operation concatenates interest and social factors separately to obtain the final embedding representations $[\mathbf{z}_u^{xx} : \mathbf{z}_u^{sx}]$ and $[\mathbf{z}_u^{ss} : \mathbf{z}_u^{xs}]$. To investigate the impact of disentangling and integration, we perform a study on them. We design three variants of DVSR and compare their performance on all datasets. The first variant is denoted as DVSR-I which keeps disentangling operations but does not exchange cross-domain factors. In other words, disentangled factors participate directly in the decoding stage in the original VAE without any integration operation. Besides, the integration is based on the disentangling, so we can not eliminate the disentangling operation to validate the integration. However, we can remove the disentanglement losses which greatly improve the quality of disentangled factors. Therefore, we design the second variant denoted by DVSR-D, which removes all disentanglement losses in model training and keeps the integration operation. The third variant is denoted by DVSR-D&I, which eliminates all disentanglement losses and integration operations. Essentially, DVSR-D&I makes DVSR degrade to Multi-VAE [25]. The results are given in Table 6, and from the table we have the following observations:

- All disentangling and integration operations are important in our model. It is easy to see that the performance of both DVSR-D and DVSR-I is worse than DVSR on all datasets. DVSR-D can combine factors from the two domains, but these factors may contain a lot of residual entangled information due to the lack of the disentangling operation. On the contrary, DVSR-I can ensure the quality of disentangled factors, but these factors can not be used to perform the proper recommendation task due to the lack of integration.
- Disentangling operations can make a greater contribution to datasets with less entangled information than to datasets with more. From Table 6, we also observe that DVSR-D achieves the worst performance on Yelp among these variants. We attribute the difference in performance to the different degree of information entanglement in these datasets. As shown in Sect. 4.2, the degree of information entanglement in Yelp is at a low level. If disentangled losses are removed, cross-domain factors on Yelp are more likely to incorporate harmful information that conflicts with the recommendation task compared to Flickr and Ciao. In other words, without the disentanglement, the integration operation will fail to incorporate beneficial information for the recommendation task.

6.5 Study of Model Applicability

The performance of DVSR is closely related to the degree of information entanglement in a dataset, and it is not hard to find that the user is the link of information entanglement through Eqs. (6) and (8), so we try to make the simplest and most general hypothesis: *DVSR performs better in the context with a large number of users and worse in the context with a small number of users*. To prove the hypothesis, we add a Graph Neural Network (GNN) based model, LightGCN [42], as a baseline. As well known, GNN based models are very popular in the recommendation field because of their excellent information fusion ability. These models focus on how to integrate different kinds of information through the aggregation operation. LightGCN greatly simplifies the aggregation process of GNN based models by removing the feature transformation and nonlinear activation.

To test the model applicability of DVSR, we randomly drop some users from each dataset. For the remaining users, we randomly select 10% of their processed data as the test set

and the remaining 90% as the training set. Considering the small number of users in Ciao, further reducing the number of users may disrupt the distribution of the data. Therefore, the experiment is conducted only on Yelp and Flickr. Results with varying reduction ratios are shown in Fig. 4.

From the figure, we can see that DVSR performs much better than LightGCN on the two datasets when the number of users is large. However, as the number of users gradually decreases, different trends become apparent on the two datasets. On Yelp, when the number of users decreases by about 25%, the performance of LightGCN and DVSR becomes similar. On Flickr, when the number of users decreases by around 15%, their performance becomes similar. The reason for the different trends in the two datasets is that the level of information entanglement in Flickr is much higher than that in Yelp. As a result, reducing the number of users in Flickr has a more severe impact on DVSR. So DVSR can be generalized to datasets with a large number of users.

7 Conclusions and Future Work

In this paper, we analyze the source of the entanglement of different information in the interest domain and the social domain and propose a method to estimate the amount of entangled information. We find that the phenomenon of entanglement exists in many datasets, so we try to enhance a recommender model from the disentanglement view. Motivated by the disentanglement in other fields, we develop the DVSR model. It critiques different domain information for disentangled factors from the same representation and integrates the factors with the same semantic information. In addition, DVSR has some auxiliary tasks to encourage disentanglement and critiquing. Experiments in our work demonstrate the effectiveness of DVSR.

As we know, not only users are influenced by their social connections, but also users' interests are dynamic. Our future work is about how to model the two aspects of users from the disentanglement view. Besides disentangling social information, we will adopt a similar strategy to disentangle a user's invariant preference and variant preference with time for high-quality recommendation performance.

Author Contributions Z-Y, H-J and Y-J conceived and designed the study and established the proposed model. H-J and Z-Y wrote the algorithms and executed experiments. Z-Y, H-J and Y-J made great contributions to the writing of the study. All authors approved the submitted version.

Funding The authors have no relevant financial or non-financial interests to disclose.

Declarations

Conflict of interest The authors declare no Conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. van den Berg R, Kipf TN, Welling M (2017) Graph convolutional matrix completion. arXiv preprint [arXiv:1706.02263](https://arxiv.org/abs/1706.02263)
2. Ma H, Yang H, Lyu MR, King I (2008) SoRec: social recommendation using probabilistic matrix factorization. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM 2008), (pp. 931–940)
3. Sattar A, Bacchiu D (2023) Graph neural network for context-aware recommendation. *Neural Process Lett* 55(5):5357–5376
4. Wang C, Zhang H, Li L, Li D (2023) Knowledge graph attention network with attribute significance for personalized recommendation. *Neural Process Lett* 55(4):5013–5029
5. Ma H, Zhou D, Liu C, Lyu MR, King I (2011) Recommender systems with social regularization. In: Proceedings of the 4th International Conference on Web Search and Web Data Mining (WSDM 2011), (pp. 287–296)
6. Jamali M, Ester M (2010) A matrix factorization technique with trust propagation for recommendation in social networks. In: Proceedings of the 2010 ACM Conference on Recommender Systems (RecSys 2010), (pp. 135–142)
7. Yang B, Lei Y, Liu D, Liu J (2013) Social collaborative filtering by trust. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI 2013), (pp. 2747–2753)
8. Yu J, Yin H, Li J, Gao M, Huang Z, Cui L (2022) Enhance social recommendation with adversarial graph convolutional networks. *IEEE Trans Knowl Data Eng* 34(8):3727–3739
9. Yu J, Gao M, Li J, Yin H, Liu H (2018) Adaptive implicit friends identification over heterogeneous network for social recommendation. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM 2018), (pp. 357–366)
10. Yang L, Liu Z, Dou Y, Ma J, Yu PS (2021) ConsisRec: enhancing GNN for social recommendation via consistent neighbor aggregation. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021), (pp. 2141–2145)
11. Wang M, Wu Z, Sun X, Feng G, Zhang B (2019) Trust-aware collaborative filtering with a denoising autoencoder. *Neural Process Lett* 49(2):835–849
12. Fan W, Derr T, Ma Y, Wang J, Tang J, Li Q (2019) Deep adversarial social recommendation. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI 2019), (pp. 1351–1357)
13. Yu J, Yin H, Gao M, Xia X, Zhang X, Hung NQV (2021) Socially-aware self-supervised tri-training for recommendation. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2021), (pp. 2084–2092)
14. Kim H, Mnih A (2018) Disentangling by factorising. In: Proceedings of the 35th International Conference on Machine Learning (ICML 2018), (pp. 2654–2663)
15. Ma L, Sun Q, Georgoulis S, Gool LV, Schiele B, Fritz M (2018) Disentangled person image generation. In: Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018), (pp. 99–108)
16. Higgins I, Matthey L, Pal A, Burgess CP, Glorot X, Botvinick MM, Mohamed S, Lerchner A (2017) beta-VAE: learning basic visual concepts with a constrained variational framework. In: The 5th International Conference on Learning Representations (ICLR 2017)
17. Kingma DP, Welling M (2014) Auto-encoding variational Bayes. In: The 2nd International Conference on Learning Representations (ICLR 2014)
18. Zhang K, Liu Q, Huang Z, Cheng M, Zhang K, Zhang M, Wu W, Chen E (2022) Graph adaptive semantic transfer for cross-domain sentiment classification. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2022), (pp. 1566–1576)
19. Tian J, Wang K, Xu X, Cao Z, Shen F, Shen HT (2022) Multimodal disentanglement variational autoencoders for zero-shot cross-modal retrieval. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2022), (pp. 960–969)
20. Fan W, Ma Y, Li Q, He Y, Zhao YE, Tang J, Yin D (2019) Graph neural networks for social recommendation. In: Proceedings of the 2019 World Wide Web Conference (WWW 2019), (pp. 417–426)
21. Wu L, Sun P, Hong R, Ge Y, Wang M (2021) Collaborative neural social recommendation. *IEEE Trans Knowl Data Eng* 51(1):464–476
22. Wu L, Li J, Sun P, Hong R, Ge Y, Wang M (2022) DiffNet++: a neural influence and interest diffusion network for social recommendation. *IEEE Trans Knowl Data Eng* 34(10):4753–4766
23. Abinaya S, Devi MKK (2021) Enhancing top-N recommendation using stacked autoencoder in context-aware recommender system. *Neural Process Lett* 53(3):1865–1888
24. Sedhain S, Menon AK, Sanner S, Xie L (2015) AutoRec: autoencoders meet collaborative filtering. In: Proceedings of the 24th International Conference on World Wide Web (WWW 2015), (pp. 111–112)

25. Liang D, Krishnan RG, Hoffman MD, Jebara T (2018) Variational autoencoders for collaborative filtering. In: Proceedings of the 2018 World Wide Web Conference (WWW 2018), (pp. 689–698)
26. Wu Y, DuBois C, Zheng AX, Ester M (2016) Collaborative denoising auto-encoders for top-N recommender systems. In: Proceedings of the 9th ACM International Conference on Web Search and Data Mining (WSDM 2016), (pp. 153–162)
27. Lee W, Song K, Moon I (2017) Augmented variational autoencoders for collaborative filtering with auxiliary information. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM 2017), (pp. 153–162)
28. Liu D, Cheng P, Zhu H, Dong Z, He X, Pan W, Ming Z (2021) Mitigating confounding bias in recommendation via information bottleneck. In: Proceedings of the 15th ACM Conference on Recommender Systems (RecSys 2021), (pp. 351–360)
29. Zhao Z, Chen J, Zhou S, He X, Cao X, Zhang F, Wu W (2023) Popularity bias is not always evil: Disentangling benign and harmful bias for recommendation. *IEEE Trans Knowl Data Eng* 35(10):9920–9931
30. Nema P, Karatzoglou A, Radlinski F (2021) Disentangling preference representations for recommendation critiquing with B-VAE. In: Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM 2021), (pp. 1356–1365)
31. Ma J, Zhou C, Cui P, Yang H, Zhu W (2019) Learning disentangled representations for recommendation. In: *Advances in Neural Information Processing Systems* 32 (NeurIPS 2019), (pp. 5712–5723)
32. Wu J, Fan W, Chen J, Liu S, Li Q, Tang K (2022) Disentangled contrastive learning for social recommendation. In: Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM 2022), (pp. 4570–4574)
33. Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G (2009) The graph neural network model. *IEEE Trans Knowl Data Eng* 20(1):61–80
34. Sun Y, Sun Z, Sha X, Zhang J, Ong YS (2023) Disentangling motives behind item consumption and social connection for mutually-enhanced joint prediction. In: Proceedings of the 17th ACM Conference on Recommender Systems, (RecSys 2023), (pp. 613–624)
35. Sha X, Sun Z, Zhang J (2022) Disentangling multi-facet social relations for recommendation. *IEEE Trans Knowl Data Eng* 9(3):867–878
36. Li N, Gao C, Jin D, Liao Q (2023) Disentangled modeling of social homophily and influence for social recommendation. *IEEE Trans Knowl Data Eng* 35(6):5738–5751
37. Rezende DJ, Mohamed S, Wierstra D (2014) Stochastic backpropagation and approximate inference in deep generative models. In: Proceedings of the 31st International Conference on Machine Learning (ICML 2014), (pp. 1278–1286)
38. Wu L, Chen L, Hong R, Fu Y, Xie X, Wang M (2020) A hierarchical attention model for social contextual image recommendation. *IEEE Trans Knowl Data Eng* 32(10):1854–1867
39. Tang J, Gao H, Liu H (2012) mTrust: Discerning multi-faceted trust in a connected world. In: Proceedings of the 5th International Conference on Web Search and Web Data Mining (WSDM 2012), pp. 93–102
40. Wang X, He X, Wang M, Feng F, Chua T (2019) Neural graph collaborative filtering. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2019), (pp. 165–174)
41. Kingma DP, Ba J (2015) Adam: A method for stochastic optimization. In: The 3rd International Conference on Learning Representations (ICLR 2015)
42. He X, Deng K, Wang X, Li Y, Zhang Y, Wang M (2020) LightGCN: Simplifying and powering graph convolution network for recommendation. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2020), (pp. 639–648)