ICIS 2022 Proceedings

Data Analytics for Business and Societal Challenges

Dec 12th, 12:00 AM

Enhancing Recommendation Interpretability with Tags: A Neural Variational Model

Shen Liu
Tsinghua University, liush.15@sem.tsinghua.edu.cn

Shangkun Che
Tsinghua University, csk19@mails.tsinghua.edu.cn

Hongyan Liu
Tsinghua University, liuhy@sem.tsinghua.edu.cn

Enhancing Recommendation Interpretability with Tags: A Neural Variational Model

Short Paper

Shen Liu
School of Economics and Management, Tsinghua University
Shuangqing Road 30, Beijing, China
liush.15@sem.tsinghua.edu.cn

Shangkun Che
School of Economics and Management, Tsinghua University
Shuangqing Road 30, Beijing, China
csk19@mails.tsinghua.edu.cn

Hongyan Liu
School of Economics and Management, Tsinghua University
Shuangqing Road 30, Beijing, China
liuhy@sem.tsinghua.edu.cn

Abstract

Recommender systems are widely used for assisting consumers finding interested products, and providing suitable explanations for recommendation is particularly important for enhancing consumers' trust and satisfaction with the system. Tags can be used to annotate different types of items, yet their potential for providing interpretability is not well studied previously. Therefore, it is worthy to study how to leverage tags to enhance recommendation systems in terms of both interpretability and accuracy. This paper proposes a novel model that seamlessly fuse topic model and recommendation model, where the topic model can analyze tags to infer understandable topics, and the recommendation model can conduct accurate and interpretable recommendations based on these topics. We develop variational auto-encoding method to take advantage of neural networks to infer model parameters. Experiments on real-world datasets illustrate that the proposed method can not only achieve great recommendation performance, but also provide interpretability for the recommendation results.

Keywords: recommender system, variational inference, interpretability, tags

Introduction

Recommender system has been widely used for many Internet services such as e-commerce platforms (Pathak et al. 2010) and online audio and video services like YouTube, Tiktok and Spotify. In different online scenarios recommender system has demonstrated its business value (Amatriain and Basilico 2015). Many recommendation models have been proposed in literature. Many of them are based on latent factor methods (Koren et al. 2009), which map users and items into a high dimensional latent space. This kind of methods have shown good recommendation accuracy and been used widely in real applications. However, the exact meaning of each latent factor is not specified in these models. Therefore, we cannot understand why particular item suits one user's taste. Providing reasonable explanations for the recommendation decisions has great value for business applications, as it can enhance the consumers' trust and confidence on the recommender system as well as the business platform (Bobadilla et al. 2013).

For machine learning models, connecting important parameters with understandable explanations can help improve transparency of the models (Lipton 2018). Therefore, some works try to provide interpretability for the latent factor models by connecting each latent factor to understandable features such as representative words (Hou et al. 2019; Zhang et al. 2014) or topics inferred by topic models (McAuley and Leskovec 2013). Some works propose to incorporate topic model to infer topic distributions for users or items (Wang and Blei 2011; Chen et al. 2016). But in these works, the recommendation task and the topic inference task are not modeled under a unified framework, or additional generative process is needed between the item vector and the item topic distribution, which limit the interpretability of such models.

Neural network models can help discover the complicated nonlinear relationship from inputs and demonstrate great potential in learning representations. However, few of them can provide interpretability due to the "black-box" property of deep learning models (Da'u and Silim 2020). Some works introduce neural networks to learn more effective user and item vector representations from side information (Wang et al. 2015), but they cannot leverage the semantics merit in the side information, especially text information. Since the Variational Auto-encoding (VAE) framework (Kingma and Welling 2013) provides possibility to combine neural networks with probabilistic graphical models, some works consider using it to extract the side information and to conduct item recommendations (Chen and Rijke 2018; Liang et al. 2018; Xiao and Shen 2019). However, they do not employ topic model as the tool to extract the understandable features.

Thus, how to combine the advantages of topic model and latent factor-based recommendation model to improve both recommendation accuracy and interpretability is still an open problem. To fill up this research gap, we propose a novel model named SVDVAE. Many online platforms provide the tagging system for annotating the items, and these tags imply some basic features of the items with concise and understandable words or phrases. Our proposed SVDVAE model utilizes tag information to provide explanations for the recommendation. It takes full advantage of the VAE framework to combine probabilistic graphical model and the neural network model. Compared to existing neural network-based recommender systems, the SVDVAE model can infer the meaningful topics from tags and directly connect each element in the latent vectors to a specific inferred topic. Thus, it can leverage the semantics of tags and provide interpretability for the recommender system. Compared to existing topic model-based recommender systems, the SVDVAE model improves the recommendation performance by using neural networks to conduct recommendation.

Related Works

The model proposed in this paper is related to the hybrid recommendation method and VAE-based recommendation method. Our work is also related to text-based interpretable models.

Hybrid recommendation methods take advantages of collaborative and content-based recommendation methods. Loosely coupled hybrid methods usually combine separate collaborative and content-based models heuristically, while tightly coupled hybrid methods utilize them in a unified framework (Wang et al. 2015). As the feature extraction task and the recommendation task can mutually facilitate and guide the learning of each other, tightly coupled methods usually achieve better performance (Wang et al. 2013). A large proportion of side information on the Internet are texts, like reviews and tags. Therefore, some works try to integrate topic model with recommender systems. But these works do not provide unified framework for recommendation and topic inference task (McAuley and Leskovec 2013) or introduce new obstacles to explaining the recommendation results (Chen et al. 2016). In recent years, methods such as Collaborating Deep Learning model (Wang et al. 2015) and Collaborative Variational Autoencoder model (Li and She 2017) also develop hybrid models that leverage neural networks to extract features from side information. While they do not consider leveraging the semantics in the side information to provide interpretability. Compared to these works, our proposed model is a tightly-coupled hybrid model that can simultaneously enjoy the interpretable property of topic models and the representation power of neural networks.

For recommender systems developed from Bayesian models, VAE method provides possibility for them to enjoy the benefit of neural networks (Kingma and Welling 2013). For example, Liang et al. (2018) apply VAE method to the interaction matrix generation process, and then the recommendation is conducted based on the regenerated interaction vector. Chen and Rijke (2018) extend this work by pre-training the parameters of the autoencoder with side information. Xiao and Shen (2019) propose a framework to use many variational autoencoders simultaneously to extract features from side information. All these methods can achieve better recommendation accuracy due to the introduction of neural networks into recommendation models. However, they do not consider how to extract interpretable features from the semantics of side information. Our proposed model fills this research gap.

Interpretable machine learning methods can increase people's trust in the model and enable them to have better model diagnosis capabilities (Ribeiro et al. 2016). Text-based interpretable models can be roughly divided into two categories, attention-based model and factor-based model. Factor-based models are widely used in recommender systems. By linking the latent factors of items with understandable concepts, we can use these concepts to generate explanations for model structure and classification results, which can improve model transparency and model functionality. Related works associate latent factors with understandable words (Zhang et al. 2014) or topics (McAuley and Leskovec, 2013). Since these models perform recommendation based on the similarity between the user latent factors and the item latent factors, the aspects can then be used to construct the corresponding recommendation explanations. However, the concept extraction task and the recommendation task are not simultaneously conducted for these works, which limit the performance of such models.

Problem Formulation

It is a common practice in recommendation models to map users and items into a K-dimension space and use latent vectors to represent user j and item i, $\boldsymbol{u}_j \in \mathcal{R}^K$ and $\boldsymbol{v}_i \in \mathcal{R}^K$. Each element in the user vector $\boldsymbol{u}_j$ measures the preference of user j on a certain latent factor, while the element in the item vector $\boldsymbol{v}_i$ measures the degree of association between item i and the latent factor. For example, the classical matrix factorization model (Koren et al. 2009) assumes that user j's rating on item i is $R_{ji} = \boldsymbol{u}_j^T \boldsymbol{v}_i$, and it estimates the value of these two groups of vectors based on the observed ratings. This idea is intuitive and has achieved good recommendation accuracy in practice. Yet the problem is that human cannot understand the meaning of the latent factors from the learned user vector $\boldsymbol{u}_j$ and item vector $\boldsymbol{v}_i$. Thus, such model lacks interpretability.

To understand the meaning of latent factors, we can introduce auxiliary information like tags into the model. An item may be annotated by a few tags, and we can infer its topic distribution $\boldsymbol{\theta}_i$ with topic model. Then by constructing a direct mapping relationship between topic distribution $\boldsymbol{\theta}_i$ and item vector $\boldsymbol{v}_i$, we can leverage the semantics of tags to endow interpretability for the model and create explanation for the recommendation. As mentioned in related works, how to conduct item recommendation and item topic inference under a unified framework and how to directly connect item vector $\boldsymbol{v}_i$ and topic distribution $\boldsymbol{\theta}_i$ to provide interpretability with topics is a problem to be solved.

We define the problem discussed in this paper as follows: Let $\boldsymbol{J} = \{j_1, j_2, \dots, j_n\}$ be the set of users and $\boldsymbol{I} = \{i_1, i_2, \dots, i_m\}$ be the set of items, where n stands for the number of users and m is the number of items. $\mathcal{S}_j \subset \mathcal{I}$ represents the items with which user j has interactions like purchasing. We use $\boldsymbol{T} = \{\boldsymbol{T}_1, \boldsymbol{T}_2, \dots, \boldsymbol{T}_m\}$ to denote the corpus, where each document $\boldsymbol{T}_i$ corresponds to a set of tags that describe item i. Note here $\boldsymbol{T}_i$ can be an empty set. $\boldsymbol{T}_i = \emptyset$ represents that item i has no tags. Given user set $\boldsymbol{J}$, item set $\boldsymbol{I}$, tag set of items $\boldsymbol{T}$ and the interaction records $\mathcal{S}_j$ of each user j, our task is to predict the score R_{ji} for each item $i \notin \mathcal{S}_j$ and conduct recommendation based on the predicted score. In addition, the model uses $\boldsymbol{u}_j$ and $\boldsymbol{v}_i$ for estimating R_{ji}, where $\boldsymbol{u}_j \in \mathcal{R}^K$ and $\boldsymbol{v}_i \in \mathcal{R}^K$ are from the same K-dimensional space. Therefore, the other task is to leverage the semantics of tags to give an understandable meaning for each dimension in the space.

Model Description

This paper proposes a probabilistic graphical model that incorporates the recommendation task and the topic inference task into a unified framework.

Probabilistic Graphical Model

The proposed SVDVAE model has two parts, the topic inference part and the recommendation part, aiming at inferring topic distributions from item's tags and conducting recommendation for each user respectively. We demonstrate the generative process of SVDVAE model in Table 1.

We assume $\boldsymbol{u}_j$ is sampled from a K-dimensional Gaussian distribution, i.e., $\boldsymbol{u}_j \sim N(0, \sigma^{-1}\mathbf{I})$. Each element in $\boldsymbol{u}_j$ illustrates user j's preference on a certain factor. For item vectors, we also assume they are Gaussian distributed and they have K dimensions, i.e., for item i, its corresponding item vector is $\boldsymbol{v}_i \sim N(\mu_{Dir}(\alpha), diag(\sigma_{Dir}^2(\alpha)))$. $\mu_{Dir}(\alpha)$ and $\sigma_{Dir}^2(\alpha)$ are the mean and the diagonal of the covariance matrix of the Gaussian distribution, which can be approximated given concentration parameters α (Hennig et al. 2012). $\boldsymbol{v}_i$ demonstrates the relevance of item i to all the factors, therefore, by calculating the dot multiplication between $\boldsymbol{u}_j$ and the item vector $\boldsymbol{v}_i$, i.e., the summation of user j's fitness to item i on all the factors, we can evaluate the overall preference of user j on item i. We use $sigmoid(\boldsymbol{u}_j^T\boldsymbol{v}_i)$ to calculate the probability that user j has an interaction with item i, and use a Bernoulli distribution to generate the observation $R_{ji} \sim Bern(sigmoid(\boldsymbol{u}_j^T\boldsymbol{v}_i))$. $R_{ji} = 1$ represents that user j interacts with item i, i.e., $i \in \mathcal{S}_j$.

1.	Sample the topic-tag matrix $\boldsymbol{B}$ and calculate the background vector $\boldsymbol{d}$
2.	For user j:
3.	Sample the user vector $\boldsymbol{u}_j \sim N(0, \sigma^{-1}\mathbf{I})$
4.	For item i:
5.	Sample the item vector $\boldsymbol{v}_i \sim N(\mu_{Dir}(\alpha), diag(\sigma_{Dir}^2(\alpha)))$
6.	the topic distribution of item i is $\boldsymbol{\theta}_i = softmax(\boldsymbol{v}_i)$
7.	For the l-th tag of item i:
8.	$T_{il} \sim Multinomial(softmax(\boldsymbol{\theta}_i\boldsymbol{B} + \boldsymbol{d}))$
9.	For user j's implicit feedback on item i:
10.	$R_{ji} \sim Bern(sigmoid(\boldsymbol{u}_j^T\boldsymbol{v}_i))$

Table 1. Generative Process of the SVDVAE Model

The SVDVAE model acquires interpretability by connecting the latent factors to understandable concepts. In this paper, we assume that items may have textual side information like tags. Using topic models, we can extract topics from these tags and each topic is described by a distribution over these tags. Since these topics have coherent semantic meanings, we can use topics to explain the latent factors in user vectors and item vectors. Suppose there are totally K different latent topics and each item can be represented as a mixture over K topics. The topic mixture distribution is denoted as $\boldsymbol{\theta}_i$, and it follows a multinomial distribution. We define $\boldsymbol{\theta}_i = softmax(\boldsymbol{v}_i)$, so that the topic distribution and the item vector are directly connected. In other words, we connect each topic with a specific factor in the SVDVAE model. The k-th element in $\boldsymbol{\theta}_i$ represents the probability that item i is relevant to the k-th topic. In addition, $softmax$ keeps the order in the item vectors, thus when the element in item vector $\boldsymbol{v}_i$ has a larger value, item i will have a larger probability to demonstrates topic k. Finally, for the tags of item i, the l-th tag T_{il} is assumed to generated by a multinomial distribution whose parameter is $softmax(\boldsymbol{\theta}_i\boldsymbol{B} + \boldsymbol{d})$, where the background vector $\boldsymbol{d}$ represents the overall distribution of tags for all the items, $\boldsymbol{B} \in \mathcal{R}^{K\times V}$ is a topic-tag association matrix and $\boldsymbol{\theta}_i\boldsymbol{B}$ stands for the deviation of the tag distribution of item i from the background vector $\boldsymbol{d}$. Therefore, $softmax(\boldsymbol{\theta}_i\boldsymbol{B} + \boldsymbol{d})$ is the tag distribution of item i. Figure 1 illustrates the graphical representation of the generative process.

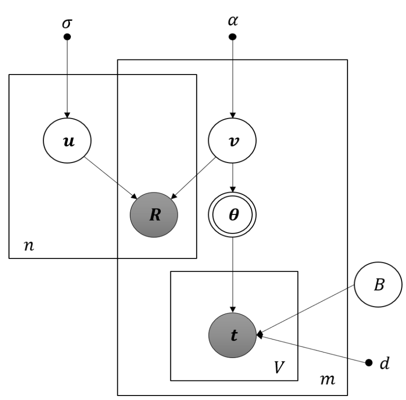

Figure 1. Probabilistic Generative Graph of the SVDVAE Model

Model Inference

The SVDVAE model includes two groups of hidden variables, i.e., user vector $\boldsymbol{u}_j$ and item vector $\boldsymbol{v}_i$, as it is intractable to estimate the posterior of these variables $p(\boldsymbol{u}_j, \boldsymbol{v}_i|\boldsymbol{R}, \boldsymbol{T})$ based on the observations directly, we conduct variational inference to the model based on the VAE framework. We assume that the variational distributions of user vector $q(\boldsymbol{u}_j)$ and item vector $q(\boldsymbol{v}_i)$ also follow the Gaussian distribution, and they are independent. The goal is to approximate the posterior with the variational distributions. To make a better approximation, the objective of the variational inference method is to minimize the Kullback-Leibler (KL) divergence between the variational distribution $q(\boldsymbol{u}_j, \boldsymbol{v}_i) = q(\boldsymbol{u}_j)q(\boldsymbol{v}_i)$ and the real posterior $p(\boldsymbol{u}_j, \boldsymbol{v}_i|\boldsymbol{R}, \boldsymbol{T})$. And it is equivalent to maximize the evidence lower bound (ELBO) defined below:

$$F(\boldsymbol{u}_j, \boldsymbol{v}_i, \boldsymbol{R}, \boldsymbol{T}) := E_{q(\boldsymbol{u}_j, \boldsymbol{v}_i)}(\log p(\boldsymbol{R}, \boldsymbol{T}|\boldsymbol{u}_j, \boldsymbol{v}_i)) - KL(q(\boldsymbol{u}_j, \boldsymbol{v}_i)||p(\boldsymbol{u}_j, \boldsymbol{v}_i)) \tag{1}$$

We use neural networks to calculate the parameters of the variational distributions following the VAE framework. For the variational distribution of user vectors $q(\boldsymbol{u}_j)$, we use a multi-layer perceptron (MLP) network with parameter set $\boldsymbol{\Phi}$ to calculate the mean $\boldsymbol{\mu}_{uj}$ and the standard deviation $\boldsymbol{\sigma}_{uj}$ of the Gaussian distribution $q(\boldsymbol{u}_j)$. The input of the neural network is the observed interaction records of user j, i.e., $\boldsymbol{R}_j = (R_{j1}, R_{j2}, \dots, R_{jm})$. We then use $q_{\boldsymbol{\Phi}}(\boldsymbol{u}_j|\boldsymbol{R}_j)$ to represent the variational distribution and the corresponding neural network interchangeably.

For the variational distribution of item vectors $q(\boldsymbol{v}_j)$, we calculate its mean and standard deviation with a similar neural network with parameter set $\boldsymbol{\Psi}$. However, we may also observe the tags of item i, thus the input of the network include item i's interaction records $\boldsymbol{R}_i^T = (R_{1i}, R_{2i}, \dots, R_{ni})$ and its tags $\boldsymbol{T}_i$. Thus, we use $q_{\boldsymbol{\Psi}}(\boldsymbol{v}_i|\boldsymbol{R}_i^T, \boldsymbol{T}_i)$ to denote the neural network as well as the variational distribution of $\boldsymbol{v}_i$.

After getting the parameters of variational distributions, we can still not able to get the analytical result of equation (1). Therefore, we use the sampling and reparametrized trick to sample the user vector $\boldsymbol{u}_j^{(s)}$ and the item vector $\boldsymbol{v}_i^{(s)}$ (Kingma and Welling, 2013). And by substituting them into $\log p(\boldsymbol{R}, \boldsymbol{T}|\boldsymbol{u}_j, \boldsymbol{v}_i)$, we can get the Monte Carlo approximation of the loss function:

$$l := -\log p(R_{ji}|\boldsymbol{u}_j^{(s)}, \boldsymbol{v}_i^{(s)}) - \log p(\boldsymbol{T}_i|\boldsymbol{v}_i^{(s)}) + KL(q_{\boldsymbol{\Phi}}(\boldsymbol{u}_j|\boldsymbol{R}_j)||p(\boldsymbol{u}_j|\sigma)) + KL(q_{\boldsymbol{\Psi}}(\boldsymbol{v}_i|\boldsymbol{R}_i^T, \boldsymbol{T}_i)||p(\boldsymbol{v}_j|\alpha)) \tag{2}$$

Since the generative process of scores R_{ji} and tags $\boldsymbol{T}_i$ are independent, $\log p(\boldsymbol{R}, \boldsymbol{T}|\boldsymbol{u}_j, \boldsymbol{v}_i)$ in equation (1) is split into two parts, and they are respectively defined as follows:

$$p(R_{ji}|\boldsymbol{u}_j^{(s)}, \boldsymbol{v}_i^{(s)}) = R_{ji}\log \sigma(\boldsymbol{u}_j^{(s)}\boldsymbol{v}_i^{(s)}) + (1 - R_{ji})\log(1 - \sigma(\boldsymbol{u}_j^{(s)}\boldsymbol{v}_i^{(s)}))$$

$$p(\boldsymbol{T}_i|\boldsymbol{v}_i^{(s)}) = \sum_l T_{il} \log\hat{T}_{il}$$

where $\hat{T}_{il}$ is the l-th element $softmax(\boldsymbol{\theta}_i\boldsymbol{B} + \boldsymbol{d})$, i.e. the inferred probability of item i having the l-th tag. Here the topic-tag matrix $\boldsymbol{B}$ is also a parameter to be estimated, and each row of this matrix corresponds to a topic's distribution over all the tags, thus defines the meaning of this topic. The last two terms in equation (2) can be calculated analytically (Srivastava and Sutton 2017). Due to space limitation, we omit some details of the model inference. Finally, we use an ADAM optimizer to optimize the loss function. Figure 2 shows the training process of SVDVAE, and we can see that the two tasks are trained together.

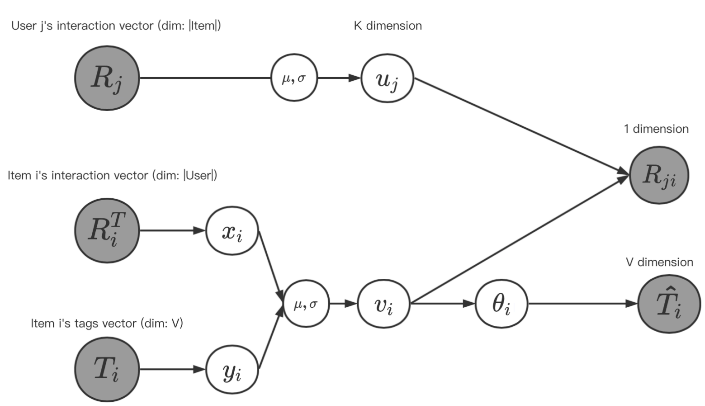

Figure 2. Training Process of the SVDVAE Model

Experiments

Experimental Setup

We conducted preliminary experiments on two datasets collected from real-world applications: Mobile-Rec and Lastfm. Mobile-Rec is collected from an Android app store with downloading records from June 2018 to August 2018. Tags are assigned by the platform and mainly describe the functions of the apps. The dataset Lastfm is collected from an online music application Last.fm, and the item to be recommended is the music producer. Tags are annotated by end users. Table 2 shows some basic statistics of these two datasets. For each dataset, we take 70%, 10% and 20% as the training set, validation set, and testing set respectively.

	No. of Users	No. of Items	No. of Tags	Feedback Density	No. of tagged Items	Avg No. of Tags per Item
Moblie-Rec	28544	13520	921	0.85%	11272	2.63
Lastfm	1863	14079	9048	0.25%	10151	7.02

Table 2. Descriptive Statistics of Datasets

We use precision and recall as the metrics to evaluate the recommendation performance of these models. We compare our model with five benchmarks. **(1) PMF** (Mnih and Salakhutdinov 2007): Probabilistic Matrix Factorization (PMF) assumes that the item vector and the user vector are generated by two Gaussian distributions, and the ratings are calculated as the inner product of corresponding vectors. **(2) CTR** (Wang and Blei 2011): CTR is a probabilistic graphical model where it includes both a PMF model for recommendation and a topic model for topic inference. **(3) HFT** (McAuley and Leskovec 2013): HFT model combines the traditional LDA model and the matrix factorization model by using a softmax function to connect the vector and the topic distribution of each item. **(4) CVAE** (Li and She 2017): CVAE leverages a variational auto-encoder to extract features for items from the side information, and the item vectors used for recommendation is constructed by adding the feature vectors to other vectors generated from a Gaussian distribution. **(5) Co-VAE** (Chen and de Rijke 2018): It conducts recommendation based on regenerated user-item feedback matrix with a variational auto-encoder model, which is pretrained with side information. It does not provide interpretability for the model.

Experimental Results

Table 3 and Table 4 demonstrate the recommendation performance of all the models on Mobile-Rec and Lastfm respectively. We set the topic number to 50 for the SVDVAE model, and set the dimension of item vectors and user vectors to 50 for all the competing models. We set topic number to 50 because by sensitivity analysis, we found that model accuracy first increases and then decreases as topic number changes from 10 to 100. And the model achieved its best accuracy when topic number equals 50. For other parameters in competing models, we adopt the default settings in related works. From the results we can easily find that the proposed SVDVAE model outperforms benchmark methods with significant improvements.

	Precision@5	Precision@10	Precision@15	Recall@5	Recall@10	Recall@15
PMF	0.1420	0.1281	0.1133	0.0150	0.0273	0.0360
CTR	0.1281	0.1161	0.1078	0.0126	0.0230	0.0318
HFT	0.1213	0.1199	0.1188	0.0128	0.0253	0.0375
CVAE	0.1180	0.1170	0.1167	0.0141	0.0278	0.0414
Co-VAE	0.2331	0.2103	0.1950	0.0264	0.0471	0.0652
SVDVAE	**0.2643**	**0.2304**	**0.2093**	**0.0294**	**0.0509**	**0.0690**

Table 3. Model Performance (Mobile-Rec)

	Precision@5	Precision@10	Precision@15	Recall@5	Recall@10	Recall@15
PMF	0.1100	0.0907	0.0737	0.0395	0.0657	0.0802
CTR	0.1144	0.0979	0.0817	0.0407	0.0697	0.0872
HFT	0.0483	0.0480	0.0458	0.0173	0.0347	0.0496
CVAE	0.2532	0.2126	0.1857	0.0955	0.1612	0.2113
Co-VAE	0.2946	0.2322	0.1989	0.1112	0.1745	0.2243

SVDVAE	**0.3454**	**0.2680**	**0.2244**	**0.1287**	**0.1997**	**0.2510**

Table 4. Model Performance (Lastfm)

Among the benchmark models, Co-VAE performs the best on both datasets. With proper design as done in our proposed model, the interpretability of recommendation models can be improved by using tag information, but does use more information such as tag always lead to improved recommendation accuracy? From Tables 3 and 4 we can see that PMF performs better even than some models such as CTR, HFT and CAVE that utilize tag information on the Mobile-Rec dataset and worse than any other models on the Lastfm dataset. Thus, the answer is "No". To explore the reason behind this, we will conduct further study about the benefits of utilizing tag information. We will conduct more experiments such as ablation study and parameter sensitivity analysis to study how the characteristics of the dataset and model parameters influence the performance and interpretability of the recommendation.

Interpretability

Our proposed SVDVAE model can improve the model transparency and resolve the black-box problem with the help of item tags. It connects the user vectors and the items vectors with understandable topics, and this practice can further provide interpretable user portraits, item portraits, and recommendation explanations. We illustrate the model interpretability with case study on Mobile-Rec dataset in this section.

User Portraits

Depicting users' features and preferences can help companies provide better personalized marketing and recommendation. In SVDVAE model, the learned user vectors $\boldsymbol{u}_j$ reflect users' preferences on different topics, and we can have a better understanding of the users' demands with the understandable topics. For the Mobile-Rec dataset, Table 5 shows three users' preferences reflected by the top-5 preferred topics for each user based on the corresponding values in $\boldsymbol{u}_j$. From this table, we can find that: User 1 likes mobile apps for office, travel, investment, music and foreign language learning, User 2 likes mobile apps for photo beautification, online community, shopping, and strategic games, and User 3 likes mobile apps for books, investment, life&travel, action games and spoof games. This example demonstrates that based on user vectors $\boldsymbol{u}_j$ and understandable topics, we can have a clearer understanding of users' preferences, thus improve the transparency of the model.

User	Topic#	$\boldsymbol{u}_j$	Top5 Tags
1	11	0.5162	efficiency&office, backup, documents, notes, schedule
	17	0.5116	life, travel, tourism, hotel, surrounding information
	38	0.3162	financial management, stocks, investment, stock investment, banking
	41	0.2941	music, online music, listening to songs, rhythm, radio
	2	0.2589	learning, English, words, translation, dictionary
2	18	0.5202	camera, filter, photo beautification, photo, special-effect camera
	45	0.4366	voice, chat, community, social, chat room
	14	0.3359	ACGN, social, emoticons, community, chat
	5	0.2854	life, shopping, o2o, shopping guide, group buying
	10	0.2265	strategic game, simulation game, Europe and America, story, role playing game
3	12	0.5959	books, novels, e-books, readers, listening books
	38	0.4781	financial management, stocks, investment, stock investment, banking
	17	0.2007	life, travel, tourism, hotel, surrounding information
	0	0.1683	side scroller, battle, action, adventure, Nintendo Entertainment System
	27	0.1629	hardcore, spoof, vent, casual game, alternative

Table 5. Topic Preferences of Users $\boldsymbol{u}_j$

Item Portraits

We can also infer the most related topics for items based on their item vectors $\boldsymbol{v}_i$. Table 6 lists topic distributions of 3 items and the corresponding top tags for each topic. From this table, we can find that the inferred topic distribution $\boldsymbol{\theta}_i$ for each item is highly biased, and each item is usually strongly related to a single topic while has little relations with other topics. It is also in correspondence with our intuition: while users' preferences can be relatively diversified, items usually concentrate on some particular functions.

Item	Topic#	$\boldsymbol{\theta}_i$	Top5 Tags
1	11	0.8277	efficiency&office, backup, documents, notes, schedule
	7	0.0268	life, reminder, tools, life service, life leisure
	15	0.0085	3-7 years old, children, children's education, playing games, elementary school
	34	0.0064	air combat, gun shooting, flight simulation, flight shooting, shooting
	43	0.0063	rpg online games, online games, masterpieces, fantasy, online games
2	18	0.7987	camera, filter, photo beautification, photo, special-effect camera
	5	0.0122	life, shopping, o2o, shopping guide, group buying
	45	0.0091	voice, chat, community, social, chat room
	25	0.0075	tile-matching, Match-3 Game, leisure puzzle, bubble dragon, mahjong solitaire
	7	0.0073	life, reminder, tools, life service, life leisure
3	15	0.2234	3-7 years old, children, children's education, playing games, elementary school
	4	0.0400	Shop, operation, simulation operation, construction, Cairo
	32	0.0349	Encyclopedia of life, women, makeup, tools, hairstyles
	7	0.0319	life, reminder, tools, life service, life leisure
	10	0.0284	strategic game, simulation game, Europe and America, story, role playing game

Table 6. Topic Distribution of Items $\boldsymbol{\theta}_i$

In addition, although only a proportion of items have tags and the model parameters are trained with tags from these items, for those untagged items, the proposed SVDVAE model can still infer proper item vectors that reflect their features. For example, Item 3 in Table 6 itself has no tags in the dataset. SVDVAE can still infer item vector for it based on the user-item feedback information and the tag assignment information of other items. Item 3 is a game of operating ice cream shop for children and the topic distribution verifies it.

Recommendation Explanation

Based on the understandable user vectors $\boldsymbol{u}_j$ and item vectors $\boldsymbol{v}_i$, SVDVAE model can provide text explanations for the recommendation results. We use the dot multiplication between $\boldsymbol{u}_j$ and $\boldsymbol{v}_i$ to evaluate user j's rating on item i, therefore, we can sort the value of $u_{jk}v_{ik}$, $(k = 1,2,\ldots,K)$, and the topic k with the highest value is the main consideration when recommending item i to user j. By substituting the top tags of this topic into a pre-defined sentence, we may construct an explanation for the recommendation.

For example, SVDVAE will recommend Item 1 in Table 6 to User 1 in Table 5, and the topic with the largest value of $u_{jk}v_{ik}$ is Topic 11. The corresponding tags for this topic are "efficiency&office, backup, documents, notes, schedule", so the SVDVAE model can provide information like "We make the recommendation for you because you may like mobile apps for efficiency&office, backup, documents, notes, and schedule. "

Conclusion

This paper proposes a novel model named SVDVAE that applies tag information to provide interpretability for latent factor-based recommender system. SVDVAE models the item recommendation task and the item topic inference task in a unified probabilistic graphical model, and uses neural networks to infer the distributional parameters within the VAE framework. These two tasks share the same parameter space, and they mutually facilitate the parameter inference during the training process. By applying the VAE framework to a probabilistic graphical model, we can simultaneously enjoy the advantage of neural networks for modeling non-linear relations and the advantage of topic models for retaining semantic features. Tag information is widely used on the Internet and can be used to annotate different types of items, such as movies, music, games, etc. Moreover, the proposed SVDVAE model does not require all items to have tags. Therefore, our model has a wide range of application scenarios.

Preliminary experimental studies show that our proposed model has better recommendation accuracy compared with benchmark models while enhancing model interpretability. Meanwhile, we also find that utilizing side information such as tag for recommendation may not always bring benefit to accuracy. We will further study the reason behind this phenomenon in the future.

References

Amatriain, X., & Basilico, J. 2015. "Recommender systems in industry: A netflix case study," in *Recommender systems handbook*, Boston: Springer, pp. 385-419.

Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. 2013. "Recommender systems survey," *Knowledge-based systems* (*46)*, pp. 109-132.

Chen, C., Zheng, X., Wang, Y., Hong, F., & Chen, D. 2016. "Capturing semantic correlation for item recommendation in tagging systems," in *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 30), No. 1.

Chen, Y., & de Rijke, M. 2018. "A collective variational autoencoder for top-n recommendation with side information," in *Proceedings of 3rd workshop on deep learning for recommender systems*, pp. 3-9.

Cheng, H. T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., ... & Shah, H. 2016. "Wide & deep learning for recommender systems," in *Proceedings of the 1st workshop on deep learning for* recommender systems, pp. 7-10.

Da'u, A., & Salim, N. 2020. "Recommendation system based on deep learning methods: a systematic review and new directions," Artificial Intelligence Review (53:4), pp. 2709-2748.

Hennig, P., Stern, D., Herbrich, R., & Graepel, T. 2012. "Kernel topic models," in *Artificial intelligence and statistics*, pp. 511-519.

Hou, Y., Yang, N., Wu, Y., & Yu, P. S. 2019. "Explainable recommendation with fusion of aspect information," *World Wide Web* (*22*:1), pp. 221-240.

Kingma, D. P., & Welling, M. 2013. "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114.*

Koren, Y., Bell, R., & Volinsky, C. 2009. "Matrix factorization techniques for recommender systems," *Computer* (*42*:8), pp. 30-37.

Li, X., & She, J. 2017. "Collaborative variational autoencoder for recommender systems," in *Proceedings of 23rd ACM SIGKDD international conference on knowledge discovery and data mining,* pp. 305-314.

Liang, D., Krishnan, R. G., Hoffman, M. D., & Jebara, T. 2018. "Variational autoencoders for collaborative filtering," in *Proceedings of the 2018 world wide web conference*, pp. 689-698.

Lipton, Z. C. 2018. "The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery," *Queue* (*16*:3), pp. 31-57.

McAuley, J., & Leskovec, J. 2013. "Hidden factors and hidden topics: understanding rating dimensions with review text," in *Proceedings of the 7th ACM conference on Recommender systems*, pp. 165-172.

Mnih, A., & Salakhutdinov, R. R. 2007. "Probabilistic matrix factorization," *Advances in neural information processing systems* (*20)*.

Pathak, B., Garfinkel, R., Gopal, R. D., Venkatesan, R., & Yin, F. 2010. "Empirical analysis of the impact of recommender systems on sales," *Journal of Management Information Systems* (*27*:2), pp. 159-188.

Ribeiro, M. T., Singh, S., & Guestrin, C. 2016. "Model-agnostic interpretability of machine learning," *arXiv preprint arXiv:1606.05386*.

Srivastava, A., & Sutton, C. 2017. "Autoencoding variational inference for topic models," *arXiv preprint arXiv:1703.01488*.

Wang, C., & Blei, D. M. 2011. "Collaborative topic modeling for recommending scientific articles," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 448-456.

Wang, H., Chen, B., & Li, W. J. 2013. "Collaborative topic regression with social regularization for tag recommendation," in *Twenty-Third International Joint Conference on Artificial Intelligence*.

Wang, H., Wang, N., & Yeung, D. Y. 2015. "Collaborative deep learning for recommender systems," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1235-1244.

Xiao, T., & Shen, H. 2019. "Neural variational matrix factorization with side information for collaborative filtering," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 414-425.

Zhang, Y., Lai, G., Zhang, M., Zhang, Y., Liu, Y., & Ma, S. 2014. "Explicit factor models for explainable recommendation based on phrase-level sentiment analysis," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pp. 83-92.