International Journal of Machine Learning and Cybernetics (2019) 10:3273–3284
https://doi.org/10.1007/s13042-019-01016-2

ORIGINAL ARTICLE

Neural variational collaborative filtering with side information for top-K recommendation

Xiaoyi Deng[1,2] · **Fuzhen Zhuang**[3,4] · **Zhiguo Zhu**[5]

Received: 29 March 2019 / Accepted: 6 September 2019 / Published online: 13 September 2019

Abstract

Collaborative filtering (CF) is one of the most widely applied models for recommender systems. Despite its success, CF-based methods suffer from rating sparsity and cold-start problem, which leads to poor quality of recommendations. Previous studies have gave great attention to construct hybrid methods, by incorporating side information and user rating. Variational autoencoder (VAE) has been confirmed to be highly effective in CF task, due to its Bayesian nature and non-linearity. However, rating sparsity remains a great challenge to most VAE models, which leads to poor latent user/item representations. In addition, most existing VAE-based methods model either latent user factors or latent item factors, resulting in the incapacity to recommend items to a new user or suggest a new item to existing users. To address these problems, we design a novel deep hybrid framework for top-*k* recommendation, neural variational collaborative filtering (NVCF), and propose three NVCF-based instantiation. In generative process, the side information of user and item is incorporated to alleviate rating sparsity, for learning better latent user/item representations. In inference process, a Stochastic Gradient Variational Bayes approach is employed to approximate the unmanageable distributions of latent user/item factors. Experiments performed on four public datasets have indicated our methods significantly outperform the state-of-the-art hybrid CF models and VAE-based methods.

Keywords Neural collaborative filtering · Variational autoencoder · Top-K recommendation · Side information · Implicit feedback

1 Introduction

Recommender systems can help users to discover their potentially preferences from varieties of items on the basis of their tastes [1]. Collaborative filtering (CF) is one of the key techniques to build personalized recommender systems, due to its accuracy and scalability [2]. The essence of CF is to infer users' preferences from the behavior data of themselves and other users. Most conventional CF methods are based upon matrix factorization (MF) [3], which projects users and items into a shared latent space and uses a latent feature vector to represent either a user or an item [4]. However, MF-based methods suffer from rating sparsity, so that the accuracy of learning latent user/item representations is limited. To address rating sparsity, large numbers of works incorporate users' and items' side information into conventional MF models. For more accurate extraction of latent factors from side information, previous studies employ latent Dirichlet allocation (LDA) [5, 6], Bayesian personalized ranking [7, 8], denoising autoencoder (DAE) [9] and stacked denoising autoencoder (SDAE) [10–13] to model

Supported by the National Natural Science Foundation of China (Nos.71401058, 71672023, 61773361), and the Program for New Century Excellent Talents in Fujian Province University (NCETFJ).

✉ Xiaoyi Deng
londonbell.deng@gmail.com

1 Business School, Huaqiao University, Quanzhou 362021, China

2 Research Center for Applied Statistics and Big Data, Huaqiao University, Xiamen 361021, China

3 Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, China

4 School of Information Engineering and Research Center of Digital Medical Image Technique, Zhengzhou University, Zhengzhou 450001, China

5 School of Management Science and Engineering, Dongbei University of Finance and Economics, Dalian 116025, China

side information of users or items. Nevertheless, these methods use inner product to model interactions between users and items, which restricts their capability of capturing non-linearity [14]. To model non-linear interaction, various approaches apply deep neural networks to model these interactions and achieve promising performance, such as neural collaborative filtering (NCF) [14], deep matrix factorization (DMF) [15], neural factorization machine (NFM) [16], DeepFM [17], JRL [18], GCMC [19], DeepCoNN [20], ConvNCF [21] and IRGAN [22]. Nonetheless, these deep neural networks cannot capture the uncertainty of latent user/item representations.

Currently, several works have taken advantage of deep generative models to perform CF task, such as variational autoencoder (VAE) [23]. VAE is a non-linear probabilistic model that has the capability of capturing uncertainty, and the non-linearity enable it to explore non-linear probabilistic latent-variable models on large-scale recommendation datasets, such as collaborative variational autoencoder (CVAE) [24], CLVAE [25], VAECF [26] and VAE-HPrior [27]. Despite the effectiveness of these VAE-based methods, there are still several drawbacks. CVAE directly uses inner product to model interaction, which hinders itself to learn non-linear interactions between users and items. CLVAE and VAECF only exploit the rating information, which leads in poor performance as the sparsity of rating matrix is extremely high. VAECF and VAE-HPrior only model users' behaviors to generate prediction, which makes them unable to recommend an item to a new user. Besides, VAECF selects the same Gaussian prior for all users, resulting in poor latent user representations [28].

To solve the problems mentioned above, we devise a deep hybrid framework, neural variational collaborative filtering (NVCF), and propose three NVCF-based instantiations with side information for top-k recommendation. Different from the user/item generative processes in most existing VAE-based methods, we model the generative process of both users and items through a unified neural variational model with parallel structure, which can effectively learn non-linear latent representations of users and items for CF. The side information of users and items is incorporated into their latent factors through a deep neural network for neural CF task, which means NVCF can mitigate rating sparsity and model better latent representations of users and items. The parameters of prior neural network are learned from data, leading to the fact that it is able to embed users' better preferences and items' features into latent factors of users and items, respectively. For inferring the posterior of latent factors of users and items, we derived a Stochastic Gradient Variational Bayes (SGVB) algorithm to infer these posteriors, which makes the parameters of our model can be effectively learned by back-propagation. The rest of this paper is organized as follows: In Sect. 2, an overview of related works on CF models is provided. In Sect. 3, our models are presented, and the parameters learning process is discussed. The Sect. 4 presents experimental results and discussions, followed by conclusions and future work in Sect. 5.

2 Related work

In recent years, the deep learning methods have attained tremendous achievements in various fields [29, 30]. Due to the abilities of neural networks to discover non-linear and subtle relationships in user-item feedbacks, many works utilize neural networks to address the task of CF. To incorporate item content information into latent item factors, collaborative deep learning (CDL) [10] integrates SDAE into probabilistic matrix factorization (PMF), which can balance the influences of user ratings and side information. Collaborative deep ranking (CDR) [11] utilizes pair-wise framework with implicit feedback, which leverages deep feature representation of item content into Bayesian pair-wise ranking. Deep collaborative filtering Framework [12] utilizes deep feature learning to aid collaborative recommendation, which embeds the content information of items and users while CDL and CDR only consider the effects of item features. Recently, the additional stacked denoising autoencoder (aSDAE) [13] was presented to incorporate side information into MF, which jointly performs deep latent user/item factors learning from side information, and CF task from the user rating. GCMC [19] considers the recommendation problem as a link prediction task with graph CNNs, which can easily integrate user/item side information (such as social networks and item relationships) into the recommendation model.

Since the above methods apply inner product to model the user/item interactions, they are not able to capture the complex structure of the interaction data between users and items. NCF framework [14] was proposed to make use of both linearity of MF and non-linearity of MLP to capture linear and non-linear relationship between users and items. NFM [16] employs Bi-Interaction layer to incorporate both user rating and item content information. Based on factorization machines, DeepFM [17] seamlessly integrates factorization machine and MLP, and it can model the high-order feature interactions via deep neural network and low-order interactions via factorization machine. For joint representations of user and item, JRL [18] places a MLP above the element-wise product of user embedding and item embedding, where user and item side information is adopted to learn the corresponding user and item representations based on deep representation learning architectures. DeepCoNN [20] adopts two parallel CNNs to model user behaviors and item properties from review texts, which alleviates data sparsity and enhances the interpretability by exploiting

rich semantic representations of reviews with CNNs. ConvNCF [21] utilizes outer product instead of dot product to model user/item interaction patterns, and applies CNNs over the result of outer product to capture the high-order correlations among embeddings dimensions. IRGAN [22] is the first model which takes advantage of generative adversarial networks for item recommendation. Due to the power of capturing uncertainty and non-linearity of deep generative model [23], several works utilize deep generative model to address the task of CF. Such as, CVAE [24] applies VAE to incorporate item content information into MF. CLVAE [25] encompasses VAE through augmenting structures to model the auxiliary information and to model the implicit user feedbacks. VAECF [26] directly utilizes VAE for CF task, and VAE-HPrior [27] incorporates user-dependent priors in the latent VAE space to encode users' preferences as functions of item reviews. Unlike previous VAE-based recommendation methods, this paper constructs the generative processes of users and items through a unified neural variational framework, which enables our model to capture both linear and nonlinear latent representations of users and items.

3 Neural variational collaborative filtering with side information

In this section, we present the neural variational collaborative filtering framework (NVCF), as shown in Fig. 1. NVCF contains two main components: the feature extraction module and the NVCF module. In feature extraction process, the NVCF learns and extracts user/item features through a unified deep generative framework with parallel structure. Then, the latent user/item vectors are fed into NVCF module to learn the user-item relations, and finally generate the rating prediction (Table 1).

Table 1 Symbols and notations

Symbols	Description
$R(R_{ij})$	Rating latent factors
$U(u_i)$	User latent factors
$V(v_j)$	Item latent factors
$X(X_i)$	Latent profile factors
$Y(Y_j)$	Latent content factors
K	The dimension of latent space
P	The dimension of user side information
Q	The dimension of item side information

3.1 Notations

Given M users and N items, the latent factors of user and item are denoted by $U = \{u_i | i = 1, \ldots, M\} \in \mathbb{R}^{K \times M}$ and $V = \{v_j | j = 1, \ldots, N\} \in \mathbb{R}^{K \times N}$ respectively, where K denotes the dimensions of latent factors. For implicit feedback, the user rating matrix is denoted by $R \in \mathbb{R}^{M \times N}$, where $R_{ij} = 1$ indicates that the i-th user has interacted with the j-th item, otherwise $R_{ij} = 0$. The user's and item's side information is denoted by two "bag-of-items" vectors over users and items, $X = \{X_i | i = 1, \ldots, M\} \in \mathbb{R}^{P \times M}$ and $Y = \{Y_j | j = 1, \ldots, N\} \in \mathbb{R}^{Q \times N}$ respectively, where P and Q are the dimensions of user side information and item side information respectively. Here, we call X and Y latent profile representation and latent content representation, respectively. Given R, X and Y, the problem is to infer latent factors u_i and v_j, and then to predict the missing ratings $\hat{R}$.

3.2 Feature extraction

As mentioned in [26], most MF-based methods assume that the prior distributions of user and item latent factors are standard Gaussian distributions, and predict rating only through user-item feedback. Some MF methods incorporate either user's or item's side information into rating prediction

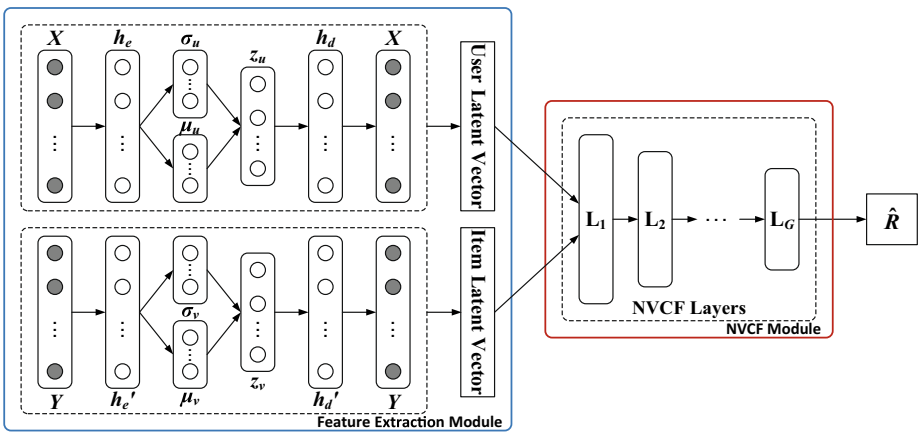

Fig. 1 NVCF framework

via linear regression, which leads to the limited accuracy of inferring latent relations between users and items. To achieve further improvement on prediction performance, our model incorporates both user's and item's side information into feature learning, which can make positive contributions to the inferring process of latent user/item factors.

3.2.1 Generative model

To learn robust features of user and item, a unified neural variational framework is built with a parallel structure. In this paper, the generative process is similar to the deep latent Gaussian model [31]. For each user u_i, the generative model starts by sampling a K-dimensional latent representation z_{u_i} from a standard Gaussian prior, i.e. $z_{u_i} \sim N(0, \mathbb{I}^K)$. The sample variable X_i is generated from its latent variable z_{u_i} through a MLP (decoder) with the generative parameter θ, i.e. $X_i \sim p_\theta(X_i|z_{u_i})$. The $p_\theta(X|z_u)$ can be generated from a multivariate Bernoulli distribution (binary) or Gaussian distribution (real-value). The generative process of user profile is defined as follows:

(1) For each layer $l \in [1, L]$ of the generative network,

a) For each column n of weight matrix W_l^d, draw
$W_{l,n}^d \sim N(0, \lambda_w^{-1}\mathbb{I}_K)$

b) Draw bias vector
$b_l^d \sim N(0, \lambda_w^{-1}\mathbb{I}_K)$

c) For each row i of h_l^d, draw

$$h_{l,i}^d \sim N(\sigma(h_{l-1,i}^d W_l^d + b_l^d), \lambda_s^{-1}\mathbb{I}_K)$$

(2) For each X_i,

a) If X_i is binary, draw
$X_i \sim B(\sigma(h_l^d W_l^d + b_{l+1}^d))$

b) If X_i is real-value, draw

$$X_i \sim N(h_l^d W_l^d + b_{l+1}^d, \lambda_X^{-1}\mathbb{I}_K)$$

where λ_w, λ_s and λ_X are hyperparameters, h_l^d represents hidden layers of decoder. Similar to SDAE, λ_s is taken to infinity for computational efficiency.

The latent representation z_{u_i} can be drawn by a Gaussian prior distribution with zero mean and identity matrix: $z_{u_i} \sim N(0, \mathbb{I}_K)$. The user's latent representation u_i consists of latent user offset and latent user profile vector:

$$u_i = \epsilon_i + z_{u_i} \tag{1}$$

The generative process of item content is similar to that of user profile, and the item latent representation v_j is composed of latent item offset and latent item content vector: $v_j = \epsilon_j + z_{v_j}$.

3.2.2 Inference model

The inference model is also a MLP network (encoder) corresponding to the one in the generative model. For user, the inference process is to approximate the intractable posterior distribution $p_\theta(z_{u_i}|X_i)$ which is determined by the generative network. Using the Stochastic Gradient Variational Bayes (SGVB) estimator, the posterior of latent user profile variable z_u can be approximated by a tractable variational distribution $q_\phi(z_{u_i}|X_i)$.

$$q_\phi(z_u|X_i) = N(\mu_\phi(X_i), diag(\sigma_\phi^2(X_i))) \tag{2}$$

where $\mu_\phi \in \mathbb{R}^K$ and $\sigma_\phi^2 \in \mathbb{R}^K$ are the mean and covariance of the approximate posterior respectively, which are outputs of the inference model (i.e. non-linear functions of X_i and the variational parameter ϕ).

Similar to [23, 26], the inference process of z_u is defined as follows:

(1) For each layer l of the inference model,

(a) For each column n of weight matrix W_l^e, draw
$W_{l,n}^e \sim N(0, \lambda_w^{-1}\mathbf{I}_K)$

(b) Draw bias vector
$b_l^e \sim N(0, \lambda_w^{-1}\mathbf{I}_K)$

(c) For each row i of h_l^e, draw

$$h_{l,i}^e \sim N(\sigma(h_{l-1,i}^e W_l^e + b_l^e), \lambda_s^{-1}\mathbf{I}_K)$$

(2) For each user u_i,

(a) Draw latent mean vector
$\mu_i \sim N(h_l^e W_\mu^e + b_\mu^e, \lambda_s^{-1}\mathbf{I}_K)$

(b) Draw latent covariance vector
$\log \sigma_i^2 \sim N(h_l^e W_\sigma^e + b_\sigma^e, \lambda_s^{-1}\mathbf{I}_K)$

(c) Draw latent content vector

$$z_{u_i} \sim N(\mu_i, diag(\sigma_i^2))$$

As explained in [26], the evidence lower bound (ELBO) for X_i can be estimated by using SGVB estimator:

$$\begin{aligned}\mathcal{L}(\theta,\phi;X_i) =& \mathbb{E}_{q_\phi(z_u|X_i)}[\log p(u_i|z_u) + \log p_\theta(X_i|z_u)] \\ & - \beta \cdot \mathbb{KL}(q_\phi(z_u|X_i)\|p(z_u)) \\ \simeq & \log p(u_i|z_{u_i,l}) + \frac{1}{L}\sum_{l=1}^{L}\log p_\theta(X_i|z_{u_i,l}) \\ & - \beta \cdot \mathbb{KL}(q_\phi(z_u|X_i)\|p(z_u)) \\ \mathbb{KL}(q_\phi(z_u|X_i)\|p(z_u)) =& \frac{1}{2}\sum_{i=1}^{M}(\mu_i^2 + \sigma_i^2 - \log\sigma_i^2 - 1) \\ & z_{u_i,l} = \mu_i + \sigma_i \odot \varepsilon_{i,l}\end{aligned} \tag{3}$$

where $\mathbb{KL}$ denotes the Kullback-Leibler divergence, $\beta \in [0, 1]$ is a parameter to control the regularization strength for addressing the posterior collapse problem [32], $\varepsilon_{i,l} \sim N(0, \mathbb{I})$, and $\odot$ represents the element-wise product.

The inference process of item content is similar to user profile inference process, and the ELBO for item network can be derived similarly:

$$\begin{aligned}\mathcal{L}(\theta,\phi;Y_j) =& \mathbb{E}_{q_\phi(z_v|Y_j)}[\log p(v_j|z_v) + \log p_\theta(Y_j|z_v)] \\ & - \beta \cdot \mathbb{KL}(q_\phi(z_v|Y_j)\|p(z_v)) \\ \simeq & \log p(v_j|z_{v_j,l}) + \frac{1}{L}\sum_{l=1}^{L}\log p_\theta(Y_j|z_{v_j,l}) \\ & - \beta \cdot \mathbb{KL}(q_\phi(z_v|Y_j)\|p(z_v)) \\ \mathbb{KL}(q_\phi(z_v|Y_j)\|p(z_v)) =& \frac{1}{2}\sum_{j=1}^{N}(\mu_j^2 + \sigma_j^2 - \log\sigma_j^2 - 1) \\ & z_{v_j,l} = \mu_j + \sigma_j \odot \varepsilon_{j,l}\end{aligned} \tag{4}$$

3.3 Side information embedded NVCF

Inspired by NCF, we propose three NVCF-based models to improve prediction performance, which are generalized MF model with side information (sGMF), MLP with side information (sMLP) and the fusion of sGMF and sMLP. The CF module of sGMF utilizes a computational method similar to the inner product of MF, which applies a linear kernel to model the latent feature interactions. The CF process of sMLP concatenates the user and item latent vectors, and then utilizes non-linear kernel to learn the interaction between user and item latent features by a MLP network. The CF part of the fused method combines sGMF and sMLP under the NVCF framework, where sGMF and sMLP share the same embedding layer, and the outputs of their interaction functions are combined. All three models integrate side information to improve prediction performance.

3.3.1 sGMF

sGMF utilizes the extracted user and item features to calculate the element-wise product of the user and item latent vectors, and outputs the calculated vectors to a fully connected neural layer. The element-wise products of the user and item latent vectors in the first neural CF layer are defined in Eq. 5. Then, sGMF projects the vectors to the output layer, as shown in Eq. 6.

$$\Psi_1(u_i, v_j) = u_i \odot v_j \tag{5}$$

$$\hat{R}_{ij} = a_{out}(\hbar^\top \Psi(u_i, v_j)) = a_{out}(\hbar^\top(u_i \odot v_j)) \tag{6}$$

where a_{out} denotes the activation function, $\hbar$ denotes edge weights of the output layer, and $\hat{R}_{ij}$ denotes the predicted rating. sGMF is intuitively equivalent to MF, as a_{out} is an identity function and $\hbar$ is a uniform vector of 1.

Under the framework of NVCF, a_{out} can be a non-linear activation function and $\hbar$ can be learned from training data, so sGMF has more powerful learning capability than MF. Dissimilar to the original GMF only relying on implicit feedback, sGMF incorporates both user and item side information into latent user/item representations learning, and employs VAE to extract user's and item's latent vectors, which can lead to better performance.

3.3.2 sMLP

sMLP uses the same way with sGMF to extract user and item features from auxiliary information. However, sMLP takes a different learning strategy in the NVCF module. Instead of treating user and item latent vectors by MF, sMLP concatenates learned latent user vector u_i and latent item vector v_j, then adopts a MLP network in NVCF module to learn high-level user-item relations. The CF process of sMLP can be defined as follows:

$$\begin{aligned}\mathcal{Z}_1 &= \Psi_1(u_i, v_j) = \begin{bmatrix} u_i \\ v_j \end{bmatrix}, \\ \Psi_2(\mathcal{Z}_1) &= a_2(W_2^\top \mathcal{Z}_1 + b_2), \\ &\cdots\cdots \\ \Psi_G(\mathcal{Z}_{G-1}) &= a_G(W_G^\top \mathcal{Z}_{G-1} + b_G) \\ \hat{R}_{ij} &= \sigma(\hbar^\top \Psi_G(\mathcal{Z}_{G-1}))\end{aligned} \tag{7}$$

where W_G, b_G, and a_G denote the weights, bias vector, and activation function for the G-th layer, respectively; the $[\cdot]$ denotes the concatenating operation.

Different from the original MLP in [14] that depends only on implicit feedback, sMLP learn user-item relations through the combination of VAE and MLP neural network,

where VAE is employed to extract user and item features from auxiliary information, and MLP is used to perform CF task. Thus, sMLP is able to learn the vital relations between users and items.

3.3.3 Fusion of sGMF and sMLP

Similar to NeuMF [14], the model for combining sGMF with a single layer sMLP can be formulated as follows.

$$\hat{R}_{ij} = \sigma\left(g^{\top} a(u_i \odot v_j) + W\begin{bmatrix} u_i \\ v_j \end{bmatrix} + b\right) \tag{8}$$

However, sharing the embedding of sGMF and sMLP might limit the performance of the fused model [14]. For those datasets where the optimal embedding size of the two models varies a lot, this solution may fail to obtain the optimal ensemble. In order to provide more flexibility to the fused model, we allow sGMF and sMLP to learn separate embeddings, and combine the two models by concatenating their last hidden layer. Figure 2 illustrates our proposed method, and the formulation is given as follows.

$$\begin{aligned} \Psi^{sGMF} &= u_i^{sG} \odot v_j^{sG}\begin{bmatrix} u_i \\ v_j \end{bmatrix} \\ \Psi^{sMLP} &= a_G\left(W_G^{\top}\left(a_{G-1}\left(\cdots a_2(W_2^{\top}\begin{bmatrix} u_i^{sM} \\ v_j^{sM} \end{bmatrix} + b_2)\cdots\right)\right) + b_G\right) \\ \hat{R}_{ij} &= \sigma\left(h^{\top}\begin{bmatrix} \Psi^{sGMF} \\ \Psi^{sMLP} \end{bmatrix}\right) \end{aligned} \tag{9}$$

where u_i^{sG}, v_j^{sG} and u_i^{sM}, v_j^{sM} represent user/item embeddings for sGMF and sMLP.

As discussed in [14], ReLU is adopted as the activation function of sMLP layers. This fusion model combines the linearity of MF and non-linearity of MLP for modelling user and item latent structures with side information, so we call it side information embedded neural variational MF (sNVMF).

3.4 Optimization

Generally, the loss function consists of the reconstruction error in feature extraction and the prediction error. The reconstruction error lies on loss functions of VAEs for user and item feature extraction, which are equivalent to their ELBOs, respectively. For convenience, the ELBOs of user and item prior networks are denoted by $\mathcal{L}_u$ and $\mathcal{L}_v$, respectively.

In prediction process, NVCF outputs the predicted rating $\hat{R}_{ij}$ for each user-item pair (u_i, v_j). Due to the nature of implicit feedback, user-item ratings can be regarded as labels with binary value, i.e., if a user is relevant to an item, the implicit feedback is 1, otherwise it is 0. Therefore, the predicted $\hat{R}_{ij}$ can be regarded as the possibility that a user relevant to an item, which means the output $\hat{R}_{ij}$ has to be constrained in the range of [0, 1] by using a sigmoid activation function. Similar to [14], the loss function can be defined as follows.

$$\mathcal{L}_{\mathrm{s}} = -\sum_{(i,j)\in\mathcal{T}\cup\mathcal{T}^-} R_{ij}\log\hat{R}_{ij} + (1-R_{ij})\log(1-\hat{R}_{ij}) \tag{10}$$

where $\mathcal{T}$ denotes the set of observed instances and $\mathcal{T}^-$ denotes a set of negative instances, which can be sampled from unobserved user-item interactions.

To minimize the objective function for NVCF, the optimization can be done by performing stochastic gradient descent (SGD), which is the same as the binary cross-entropy loss. By employing a probabilistic treatment for NVCF, the recommendation with implicit feedback can be regarded as a binary classification problem. Thus, the general loss function for training NVCF is defined as,

$$\mathcal{L} = \mathcal{L}_{\mathrm{s}} + \lambda_u \cdot \mathcal{L}_u + \lambda_v \cdot \mathcal{L}_v \tag{11}$$

where λ_u and λ_v denote the hyperparameters of the loss function.

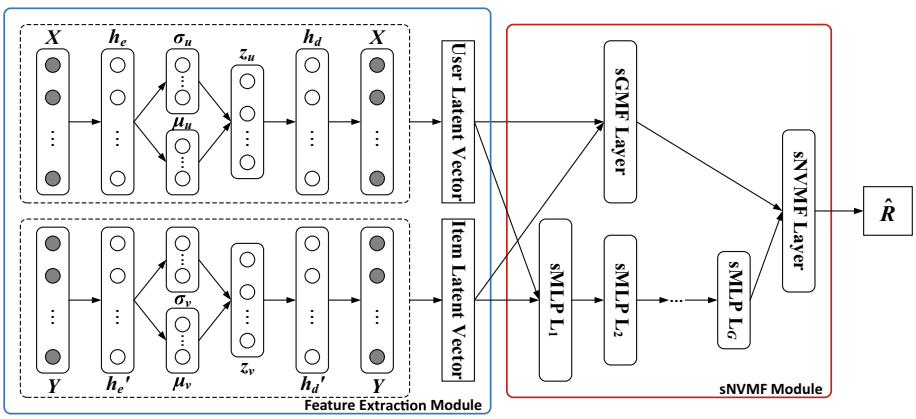

Fig. 2 Side Information embedded neural variational MF (sNVMF) Model

3.5 Prediction

After model training and parameters learning, we can predict the probability that user will rate an item for a user-item pair (u_i, v_i). Given a trained model, for a user-item pair (u_i, v_i) without any observed relation, the predicted rating can be written as follow:

$$\hat{R}_{ij} = \sigma\left(\hbar^{\top}\begin{bmatrix}\Psi^{sGMF}\\ \Psi^{sMLP}\end{bmatrix}\right),\ \hbar \leftarrow \begin{bmatrix}\alpha h^{sGMF}\\ (1-\alpha)h^{sMLP}\end{bmatrix} \tag{12}$$

where h^{sGMF} and h^{sMLP} denote the h vector of the pretrained sGMF and sMLP models, respectively; α is a hyper-parameter determining the tradeoff between the two pretrained models.

3.6 Computational complexity

The computational complexity of NVCF is $max(\mathrm{O}_{\mathrm{VAE}_u}, \mathrm{O}_{\mathrm{VAE}_v}) + \mathrm{O}_{\mathrm{NCF}}$. The first term $max(\cdot)$ is the computational complexity of the VAE part (feature extraction), $\mathrm{O}_{\mathrm{VAE}_u} = \mathrm{O}(MP^2 + LH^2)$ and $\mathrm{O}_{\mathrm{VAE}_v} = \mathrm{O}(NQ^2 + LH^2)$, where M and N are the number of users and items respectively; P and Q are the dimensions of user and item side information, respectively; L is the number of layers of MLP networks in VAE; H is the average hidden layer size. Since M, N, P and Q are all linear and $L, H \ll min(P, Q)$, the complexity of feature extraction has a squared term. The second term is the computational complexity of NCF. As we know, the complexity of NCF is linear to the size of matrix and the layers of neural network, i.e. $\mathrm{O(NCF)} = \mathrm{O}(MN + MNG)$, where G is the layers of neural network. Because $G \ll min(M, N)$, the complexity of NCF has a squared term. Thus, the computational complexity of NVCF is of a squared term.

4 Experiments and results

4.1 Experimental settings

4.1.1 Datasets

In this section, four public datasets from GroupLens, Yelp and Epinions are collected to evaluate our model, which are MovieLens-100K (ML100K), MovieLens-1M (ML1M), Yelp Challenge Dataset (Yelp) and Extended Epinions dataset (EPext). Table 2 summarizes the characteristics of four datasets.

The ML100K and ML1M have been widely utilized to evaluate CF-based recommendation algorithms. The former one contains 943 users and 1682 movies with 100,000 ratings, while the latter one includes 6040 users and 3706 movies with 1,000,209 ratings. Each rating value is on a scale of 1–5, and each user has rated at least 20 movies. These two datasets are explicit feedback data, while our goal is to investigate the performance of learning from the implicit feedback. Therefore, these two datasets are transformed into implicit data, where each entry is marked as 1 if the corresponding rating is no less than 4, otherwise marked as 0. For side information, user demographics including age, occupation and gender are regarded as collaborative information, and movie descriptions (genres) are taken as auxiliary item information.

The Yelp contains customer reviews of local businesses. Each review is associated with a rating ranging from 1 to 5, and the user-item matrix is binarized using value 3 as a threshold. The reviews are filtered out, whose text is not written in English and businesses other than restaurants. To reduce sparsity, users with less than 5 reviews and businesses that have been rated by less than 30 users are deleted. Moreover, we merge the repetitive ratings at different time-stamps to the earliest one, so as to study the performance of recommending new items to a user. The final dataset obtains 25,815 users, 25,677 items, and 730,791 ratings.

The EPext contains ratings on articles/reviews users on products on Epinions.com. This dataset is extremely sparse (its density is about 0.015%), and it includes over 13,000,000 ratings on a 5-star scale of 120,492 users on 755,760 items (articles/reviews). The corresponding rating is also assigned to a value of 1 as implicit feedback. The EPext also contains 717,667 trust relations and 123,705 distrust statements, which can be considered as user side information. For item side information, each article/review have a topic(subject) associated with it, which can be regarded as auxiliary information.

Table 2 Statistics of MovieLens, Yelp and Extended Epinions datasets

Dataset	Users	Items	Ratings	Sparsity (%)	User features	Item features
ML100K	943	1682	100,000	93.70	Demographics	Genres
ML1M	6040	3706	1,000,209	95.53	Demographics	Genres
Yelp	25,815	25,677	730,791	99.89	Social relations	Categories
EPext	120,492	755,760	13,668,319	99.98	Trust/distrust	Topics

4.1.2 Baselines and evaluation metrics

To evaluate the proposed NVCF framework and its three instantiations, six representative CF models are selected as baselines.

BPR [7] optimizes the MF model with a pairwise ranking loss, to learn from implicit feedback. It is a highly competitive baseline for item recommendation.

mDCF [12] employs SDAE to extract features from user and item auxiliary information and uses MF to determine user-item latent relations.

NeuMF [14] is a model proposed within the NCF framework, which combines hidden layer of GMF and MLP to learn the user-item interaction function.

NFM [16] generalizes factorization machine for CF, and combines the factorization machine and neural network to incorporate both feedback and content.

CVAE [24] is a Bayesian generative model that jointly models CTR and deep generative model to bridge auxiliary information together with deep architecture.

VAECF [26] is a state-of-the-art method that directly apply VAE to CF to for implicit feedback.

To evaluate the performance of our models, two common evaluation metrics for top-k recommendation are adopted: Hit Radio (HR) [14] and Normalized Discounted Cumulative Gain NDCG [33]. HR@k is a recall-based metric, measuring whether the testing item is in the top-k position. NDCG@k assigns the higher scores to the items within the top-k positions of the ranking list.

4.1.3 Parameter settings

For training set, four negative instances are sampled for each positive instance. The model parameters are randomly initialized by using a Gaussian distribution with mean of 0 and standard deviation of 0.01. Similar to [34], a mini-batch Adam method is employed to optimize model, and the learning rate and the batch size are set to 0.001 and 128 respectively. In the feature extraction step, K is set to 128. The two generative networks both are two latent layers with ReLU activation, and the two prior networks are one latent layer. The last layer of generative network is sigmoid activation, and the parameter β is set to be 0.2 to achieve the best performance of VAE, according to [26]. In NVCF step, α is set to 0.5, allowing sGMF and sMLP to initialize sNVMF equally, and the dimension of latent vectors is defined as the number of neurons in the last NVCF layer. Specifically, the layer number of sMLP are set to 4 for the best performance [14].

4.2 Experimental results

4.2.1 Overall performance

In our experiments, each dataset is split into two parts: training datasets and testing datasets. For the training set, experiments are carried out with a setting of 80% random sample of each rating, and the rest (20%) are used for testing. Tables 2 and 3 list the top-k recommendation results of all methods on four datasets, in term of HR@5/NDCG@5 and HR@10/NDCG@10, respectively.

From Tables 3 and 4, we can find that most neural network-based methods (NeuMF, NFM, VAECF, sGMP, sMLP and sNVMF) outperform linear baselines, which

Table 3 Performance comparison between all methods on HR@5 and NDCG@5

Dataset	Metrics	BPR	mDCF	NeuMF	NFM	CVAE	VAECF	sGMF	sMLP	sNVMF
ML100K	HR@5	0.4789	0.4801	0.4944	0.5053	0.4720	0.5039	0.5082	0.5045	0.5138
	IMP	9.38%	9.10%	5.95%	3.66%	10.97%	3.95%	3.07%	3.83%	–
	NDCG@5	0.3185	0.3238	0.3356	0.3392	0.3181	0.3407	0.3489	0.3388	0.3524
	IMP	10.64%	8.83%	5.01%	3.89%	10.78%	3.43%	1.00%	4.01%	–
ML1M	HR@5	0.5305	0.5319	0.5489	0.5624	0.5390	0.5646	0.5709	0.5627	0.5736
	IMP	8.12%	7.84%	4.50%	1.99%	6.42%	1.59%	0.47%	1.94%	–
	NDCG@5	0.3642	0.3688	0.3866	0.3887	0.3764	0.3924	0.4003	0.3905	0.4031
	IMP	10.68%	9.30%	4.27%	3.70%	7.09%	2.73%	0.70%	3.23%	–
Yelp	HR@5	0.1757	0.1820	0.1879	0.1931	0.1818	0.1942	0.1966	0.1928	0.1985
	IMP	12.98%	9.07%	5.64%	2.80%	9.19%	2.21%	0.97%	2.96%	–
	NDCG@5	0.1105	0.1143	0.1186	0.1222	0.1147	0.1228	0.1242	0.1220	0.1253
	IMP	13.39%	9.62%	5.65%	2.54%	9.24%	2.04%	0.89%	2.70%	–
EPext	HR@5	0.6491	0.6598	0.6776	0.6809	0.6596	0.6813	0.6915	0.6806	0.7019
	IMP	8.13%	6.38%	3.59%	3.08%	6.41%	3.02%	1.50%	3.13%	–
	NDCG@5	0.5446	0.5523	0.5679	0.5728	0.5622	0.5721	0.5817	0.5688	0.5898
	IMP	8.30%	6.79%	3.86%	2.97%	4.91%	3.09%	1.39%	3.69%	–

Table 4 Performance comparison between all methods on HR@10 and NDCG@10

Dataset	Metrics	BPR	mDCF	NeuMF	NFM	CVAE	VAECF	sGMF	sMLP	sNVMF
ML100K	HR@10	0.6233	0.6227	0.6416	0.6554	0.6248	0.6542	0.6610	0.6559	0.6816
	IMP	9.35%	9.46%	6.23%	4.00%	9.09%	4.19%	3.12%	3.92%	–
	NDCG@10	0.3502	0.3619	0.3851	0.3990	0.3596	0.4005	0.4093	0.4017	**0.4144**
	IMP	18.33%	14.51%	7.61%	3.86%	15.24%	3.47%	1.25%	3.16%	–
ML1M	HR@10	0.6819	0.6962	0.7003	0.7095	0.6971	0.7178	0.7239	0.7204	0.7326
	IMP	7.44%	5.23%	4.61%	3.26%	5.09%	2.06%	1.20%	1.69%	–
	NDCG@10	0.4117	0.4206	0.4368	0.4428	0.4234	0.4423	0.4495	0.4449	0.4590
	IMP	11.49%	9.13%	5.08%	3.66%	8.41%	3.78%	2.11%	3.17%	–
Yelp	HR@10	0.2794	0.2907	0.2956	0.3012	0.2899	0.3028	0.3061	0.3033	0.3105
	IMP	11.13%	6.81%	5.04%	3.09%	7.11%	2.54%	1.44%	2.37%	–
	NDCG@10	0.1445	0.1501	0.1536	0.1562	0.1509	0.1570	0.1588	0.1573	0.1612
	IMP	11.56%	7.40%	4.95%	3.20%	6.83%	2.68%	1.51%	2.48%	–
EPext	HR@10	0.8169	0.8292	0.8466	0.8578	0.8280	0.8509	0.8712	0.8526	0.8891
	IMP	8.99%	7.37%	5.16%	3.79%	7.52%	4.63%	2.19%	4.42%	–
	NDCG@10	0.6283	0.6367	0.6545	0.6619	0.6444	0.6587	0.6691	0.6618	0.6859
	IMP	9.17%	7.73%	4.80%	3.63%	6.44%	4.13%	2.51%	3.64%	–

demonstrates that deep neural network can help to achieve more subtle and better latent user and item representations. We also find that although sMLP is a little less robust than NFM in terms of HR@5 and NDCG@5 on ML100K, Yelp and EPext, it has better performance than other non-NVCF models in terms of HR@10 and NDCG@10 on all datasets. It is clear that sGMP, sMLP and sNVMF outperform other baselines in most cases, and sNVMF achieves the best performance on all datasets with two metrics, which indicates the effectiveness of NVCF framework to perform CF task. It is also found most VAE-based non-linear models (VAECF, sGMP, sMLP and sNVMF) achieve promising performance, which means the Bayesian nature and non-linearity of neural network can facilitate inferring better latent preferences of users and items. Among VAE models, our three NVCF-based methods outperform VAECF and CVAE in terms of all metrics on all datasets, which shows the advantages of our VAE-boosted NVCF framework.

4.2.2 Performance in cold-start scenarios

To evaluate our models in different cold-start scenarios, we form evaluation sets in different cold ratios. The datasets are split into training set (80%), validation set (10%) and test set (10%). For 30% cold users, we random choose 30% samples in the test sets and give each sample a specific user id only for the sample. We evaluate our models in 30% user cold (Cold-U) and 30% item cold (Cold-V) scenarios on all datasets in term of NDCG@5 and NDCG@10. BPR, NeuMF and VAECF only use rating information and are not able to manage cold-start scenarios well, so they are not compared with our methods under NVCF framework.

Tables 5 and 6 show the performance of our methods and other hybrid methods in different cold-start scenarios. Because CVAE cannot handle cold user problem, we do not conduct experiments in cold user scenario. It is obvious that our methods significantly outperform other methods in the scenarios of both cold items and cold users, and achieve more remarkable improvements against other methods than the case of testing all users/items. These results indicate that using neural network to model interactions between users and items works better than those of simply using inner product, and demonstrate our methods have the ability to provide high quality recommendations to cold start scenarios.

4.2.3 Sensitivity Analysis

In this section, we investigate the influence of the dimension of latent space on four datasets in term of HR@10 and NDCG@10, with sampling 80% data for training. Because sNVMF has better performance than sGMF and sMLP, we focus on the performance of sNVMF for a different dimension of the latent vector. The dimension of latent space K is set to be 8, 16, 32, 64 and 128, respectively. According to Fig. 3, it is crystal clear that larger dimension leads to better performance, and the optimal K of sNVMF for ML100K, ML1M, Yelp and EPext is 128. Thus, we set $K = 128$ as default for all datasets.

5 Conclusion

In this paper, we proposed a new hybrid deep framework, NVCF, for top-k recommendation, with three instantiation sGMF, sMLP and sNVMF, which incorporates a unified

Table 5 Performance comparison between selected methods in cold-start on NDCG@5

Dataset	Scenario	mDCF	NFM	CVAE	sGMF	sMLP	sNVMF
ML100K	Cold-U	0.1651	0.1820	–	0.2014	0.1843	0.2148
	IMP	30.10%	18.02%	–	6.65%	16.55%	–
	Cold-V	0.1439	0.1778	0.1382	0.1906	0.1797	0.1989
	IMP	38.28%	11.91%	43.98%	4.40%	10.73%	–
ML1M	Cold-U	0.1918	0.2269	–	0.2412	0.2321	0.2534
	IMP	32.12%	11.68%	–	5.06%	9.18%	–
	Cold-V	0.1879	0.1951	0.1567	0.2133	0.1995	0.2206
	IMP	17.40%	13.07%	40.78%	3.42%	10.58%	–
Yelp	Cold-U	0.0533	0.0546	–	0.0576	0.0563	0.0590
	IMP	10.69%	8.06%	–	2.43%	4.80%	–
	Cold-V	0.0609	0.0642	0.0529	0.0698	0.0642	0.0731
	IMP	20.03%	13.86%	38.19%	4.73%	13.86%	–
EPext	Cold-U	0.2976	0.3393	–	0.3594	0.3431	0.3773
	IMP	26.80%	11.20%	–	4.98%	9.95%	–
	Cold-V	0.2571	0.2962	0.2406	0.3185	0.2972	0.3274
	IMP	27.33%	10.52%	36.09%	2.81%	10.16%	–

Table 6 Performance comparison between selected methods in cold-start on NDCG@10

Dataset	Scenario	mDCF	NFM	CVAE	sGMF	sMLP	sNVMF
ML100K	Cold-U	0.1828	0.2141	–	0.2362	0.2185	0.2
	IMP	38.18%	17.98%	–	6.94%	15.61%	–
	Cold-V	0.1593	0.2086	0.1562	0.2234	0.2127	0.2333
	IMP	46.45%	11.84%	49.36%	4.43%	9.69%	–
ML1M	Cold-U	0.2187	0.2585	–	0.2708	0.2644	0.2881
	IMP	31.73%	11.45%	–	6.39%	8.96%	–
	Cold-V	0.2143	0.2219	0.1762	0.2391	0.2268	0.2509
	IMP	17.08%	13.07%	42.40%	4.94%	10.63%	–
Yelp	Cold-U	0.0699	0.0721	–	0.0737	0.0725	0.0760
	IMP	8.73%	5.41%	–	3.12%	4.83%	–
	Cold-V	0.0793	0.0827	0.0696	0.0892	0.0833	0
	IMP	18.28%	13.42%	34.77%	5.16%	12.61%	–
EPext	Cold-U	0.3351	0.3748	–	0.4201	0.3991	0.4387
	IMP	30.93%	17.03%	–	4.42%	9.92%	–
	Cold-V	0.2946	0.3230	0.2882	0.3523	0.3395	0.3799
	IMP	28.96%	17.63%	31.81%	7.84%	11.91%	–

deep generative model for hybrid deep collaborative filtering. The NVCF framework models both users' and items' generative processes, which enables it to generate recommendation under different cold-start scenarios. Our methods incorporate users' and items' side information through two parallel VAE networks, in order to mitigate rating sparsity and facilitate modeling user/item features. For inference purpose, we proposed a SGVB approach to approximate posteriors of latent user and item variables. Due to Bayesian nature and non-linearity, NVCF can learn better latent user/item factors and deal with the cold-start problem via a full Bayesian probabilistic view. Experimental results show that our methods achieve the best performance and can effectively handles user/item cold-start problem.

In the future, we plan to incorporate more structural auxiliary information to further improve the recommendation precision, such as employing knowledge graph to integrating item knowledge graph and social network with user-item graph, for establishing knowledge-aware connectivities between user and item. In addition, NVCF framework is not limited to textual auxiliary information, and can be extended to multimedia auxiliary information, such as videos and images. Thus, building recommender systems with multimedia that contain more abundant visual semantics can

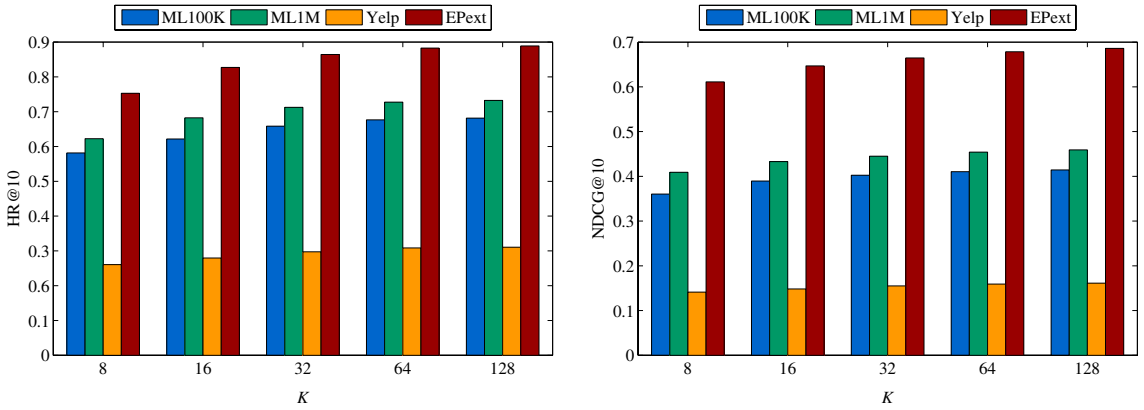

Fig. 3 sNVMF performance on HR@10 and NDCG@10 with K on four Datasets

better understand users' preferences and provide more efficient recommendation.

References

1. Bobadilla J, Ortega F, Hernando A, Gutierrez A (2013) Recommender systems survey. Knowl Based Syst 46(1):109–132
2. Shi Y, Larson M, Hanjalic A (2014) Collaborative filtering beyond the user-item matrix: a survey of the state of the art and future challenges. ACM Comput Surv 47(1):3
3. Mnih A, Salakhutdinov RR (2008) Probabilistic matrix factorization. In: Advances in neural information processing systems, pp 1257–1264
4. Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. IEEE Comput 42(8):30–37
5. Zhong J, Li X (2010) Unified collaborative filtering model based on combination of latent features. Expert Syst Appl 37(8):5666–5672
6. Wang C, Blei DM (2011) Collaborative topic modeling for recommending scientific articles. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 448–56
7. Rendle S, Freudenthaler C, Gantner Z, Schmidt-Thieme L (2009) BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings of the 25th conference on uncertainty in artificial intelligence, pp 452–461
8. Pan W, Zhong H, Xu C, Ming Z (2015) Adaptive Bayesian personalized ranking for heterogeneous implicit feedbacks. Knowl Based Syst 73:173–180
9. Strub F, Gaudel R, Mary J (2016) Hybrid Recommender System based on Autoencoders. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, pp 11–16
10. Wang H, Wang N, Yeung DY (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
11. Ying H, Chen L, Xiong Y, Wu J (2016) Collaborative deep ranking: a hybrid pair-wise recommendation algorithm with implicit feedback. In: Proceedings of the 20th Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp 555–567
12. Li S, Kawale J, Fu Y (2015) Deep collaborative filtering via marginalized denoising auto-encoder. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp 811–820
13. Dong X, Yu L, Wu Z, Sun Y, Yuan L, Zhang F (2017) A hybrid collaborative filtering model with deep structure for recommender systems. In: Proceedings of 31st AAAI Conference on Artificial Intelligence, pp 1309–1315
14. He X, Liao L, Zhang H, Nie L, Hu X, Chua TS (2017) Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web, pp 173–182
15. Xue HJ, Dai XY, Zhang J, Huang S, Chen J (2017) Deep matrix factorization models for recommender systems. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence, pp 3203–3209
16. He X, Chua TS (2017) Neural factorization machines for sparse predictive analytics. In: Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval, pp 355–364
17. Guo H, Tang R, Ye Y, Li Z, He X (2017) DeepFM: a factorization-machine based neural network for CTR prediction. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence, pp 1725–1731
18. Zhang Y, Ai Q, Chen X, Croft WB (2017) Joint representation learning for top-n recommendation with heterogeneous information sources. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp 1449–1458
19. Berg RVD, Kipf TN, Welling M (2018) Graph convolutional matrix completion. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 1–7
20. Zheng L, Noroozi V, Yu PS (2017) Joint deep modeling of users and items using reviews for recommendation. In: Proceedings of the 10th ACM International Conference on Web Search and Data Mining, pp 425–434
21. He X, Du X, Wang X, Tian F, Tang J, Chua TS (2018) Outer product-based neural collaborative filtering. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence, pp 2227–2233
22. Wang J, Yu L, Zhang W, Gong Y, Xu Y, Wang B, Zhang P, Zhang D (2017) Irgan: A minimax game for unifying generative and discriminative information retrieval models. In: Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval, pp 515–524
23. Kingma DP, Welling M (2013) Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114

24. Li X, She J (2017) Collaborative variational autoencoder for recommender systems. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 305–314
25. Lee W, Song K, Moon IC (2017) Augmented variational autoencoders for collaborative filtering with auxiliary information. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp 1139–1148
26. Liang D, Krishnan RG, Hoffman MD, Jebara T (2018) Variational autoencoders for collaborative filtering. In: Proceedings of the 2018 World Wide Web Conference, pp 689–698
27. Karamanolakis G, Cherian KR, Narayan AR, Yuan J, Tang D, Jebara T (2018) Item Recommendation with Variational Autoencoders and Heterogeneous Priors. In: Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems, pp 10–14
28. Hoffman MD, Johnson MJ (2016) Elbo surgery: yet another way to carve up the variational evidence lower bound. In: Workshop in Advances in Approximate Bayesian Inference, pp 1–4
29. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436
30. Zhang S, Yao L, Sun A (2019) Deep learning based recommender system: a survey and new perspectives. ACM Comput Surv 52(1):5
31. Rezende DJ, Mohamed S, Wierstra D (2014) Stochastic backpropagation and approximate inference in deep generative models. In: Proceedings of the 31st International Conference on International Conference on Machine Learning, pp 1278–1286
32. Bowman SR, Vilnis L, Vinyals O, Dai AM, Jozefowicz R, Bengio S (2016) Generating sentences from a continuous space. In: Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, pp 10–21
33. He X, Chen T, Kan MY, Chen X (2015) Trirank: Review-aware explainable recommendation by modeling aspects. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp 1661–1670
34. Shi S, Zhang M, Liu Y, Ma S (2018) Attention-based adaptive model to unify warm and cold starts recommendation. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp 127–136