This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2952849, IEEE Transactions on Knowledge and Data Engineering

Deep Variational Matrix Factorization with Knowledge Embedding for Recommendation System

Xiaoxuan Shen, Baolin Yi, Hai Liu, *Member, IEEE,* Wei Zhang, Zhaoli Zhang, *Member, IEEE,* Sannyuya Liu, and Naixue Xiong

Abstract—Automatic recommendation has become an increasingly relevant problem to industries, which allows users to discover new items that match their tastes and enables the system to target items to the right users. In this article, we have proposed a deep learning based fully Bayesian treatment recommendation framework, DVMF, which has high-quality performance and ability to integrate any kinds of side information handily and efficiently. In DVMF, the variational inference technique and the reparameterization tricks are introduced to make DVMF possible to be optimized by the stochastic gradient-based methods, in addition, two novel deep neural networks have been constructed to infer the hyper-parameters of the distributions of latent factors from the knowledge of user and item, which are represented as low-dimensional real-valued vectors retaining primary features. Experimental results on five public databases indicate that the proposed method performs better than the state-of-the-art recommendation algorithms on prediction accuracy in terms of quantitative assessments.

Index Terms—deep learning, matrix factorization, recommendation system, representation learning, variational inference

1 Introduction

The explosive growth of information available online frequently overwhelms users. Recommendation system is a useful information filtering tool for guiding users in a personalized way of discovering products or services they might be interested in from a wide range of possible options. Recommendation system has been playing a more vital and essential role in various information access systems to boost user experience and facilitate decision-making process.

In the past decades, numerous recommendation algorithms [1]–[3] have been developed. Collaborative filtering [4] is one of the most distinguished approaches. Collaborative filtering estimates the unknown ratings based on known ones subject to globally high accuracy and other requirements. One of the most popular collaborative filtering approaches is based on low-dimensional factor models. These models are also called matrix factorization or latent factor models.

To improve the recommendation effect, many kinds of information are introduced to enhance recommendation performance. For instance, social recommendation [3], [5] utilizes social relations or trust relations; content-based recommendation [2], [6], [7] employs the content of items or users, such as the brief introduction, video content and so forth. Based on this idea, a vast number of researchers focused on exploring a better feature extraction approaches [6], [8]. Meanwhile other researches tried to integrate many different kinds of information to push the model performance to the utmost limits [9].

With the considerable advancements in vision, speech and natural language processing tasks, deep learning has become a significant research tool in many fields. With the deep learning algorithm, artificial intelligence has achieved substantial breakthroughs in numerous areas. In recommendation system, a host of deep learning based models have been proposed in the last several years. The deep learning methods, including the convolutional neural network [6], [7] and the stacked denoising autoencoder [8], [10], are employed to estimate the prior of latent factors from the additional information introduced to the recommendation system. In addition, some methods, like restricted Boltzmann machine [11], [12], autoencoder [13], [14], neural autoregressive distribution estimator [15], [16] and recurrent neural network [17], [18], have been modified to novel recommendation frameworks. Compared with traditional recommendation algorithms, deep learning based recommendation algorithms have achieved extraordinary performance in the practical application scene.

To build a high-efficiency recommendation framework and able to merge all kinds of side information into the framework easily and efficaciously, we propose the deep variational matrix factorization (DVMF) model. First, knowledge embedding model (KEM) is proposed to convert the high-dimensional and sparse user and item knowledge into a low-dimensional real-valued vector retaining primary features. Furthermore, parametric inference model (PIM) is structured to produce the hyperparameters of the distribution of latent factors by two deep neural networks. Finally, the fully Bayesian treatment model, variational matrix factorization model (VMFM), can be implemented. The major

- Corresponding author: Baolin Yi.
- *X. Shen and B. Yi were with the National Engineering Research Center for E-Learning, Central China Normal University, Wuhan, 430074, P.R. China. E-mail: {shenxx, epower}@mail.ccnu.edu.cn*
- *H. Liu, W. Zhang and Z. Zhang are with the National Engineering Research Center for E-Learning, Central China Normal University, Wuhan 430079, China.*
- *S. Liu is with the National Engineering Research Center for E-Learning, the National Engineering Laboratory for Educational Big Data, Central China Normal University, Wuhan 430079, China.*
- *N. Xiong is with the National Engineering Laboratory for Educational Big Data, Central China Normal University, Wuhan 430079, China.*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2952849, IEEE Transactions on Knowledge and Data Engineering

contributions of our study can be summarized as follows.

- A new recommendation framework DVMF is proposed. As a fully Bayesian treatment model, DVMF introduced the variational inference technique and the reparameterization tricks to make DVMF possible to be optimized by the gradient-based methods. Thus, DVMF model can be easily implemented in large scale recommendation scenarios. Moreover, the hyperparameters of latent factors in DVMF are inferred by the PIM and KEM instead of sampling from an artificial distribution.
- A new deep neural network architecture, D-MLP, has been constructed in PIM. D-MLP structure ensures maximum information flow between layers in the deep neural network. And this structure could alleviate the vanishing gradient problem with effect.
- The KEM is developed to represent the implicit feedback data and other assistant information. KEM maps implicit feedback information graph of each user and item to a low-dimensional real-valued vector persisting key patterns. By utilizing KEM, the scale of parameters in DVMF is lessened and the training efficiency is raised vastly.
- Empirical study using real-world data has been set. We evaluate the proposed method DVMF on five real-world data sets. The result indicates that DVMF outperforms the state-of-the-art recommendation algorithms.

The article is organized as follows. In the next section, we introduce the related works and the motivation of our work. In Section III, we elaborate the DVMF model and three submodels. The optimization method and parameter determination are presented in Section IV. Experimental results on five public databases are provided in Section V, and Section VI concludes this article.

2 Related Work

2.1 Probabilistic Matrix Factorization

In collaborative filtering based recommendation system, historical behaviors of users are usually expressed as a user-item rating matrix. Given a user set U and an item set I, user-item rating matrix $\boldsymbol{Z}$ is a $|U| \times |I|$ matrix where each element r_{ij} is proportional to user i's preference on item j, $|U|$ and $|I|$ denote the size of user set and item set respectively. In the user-item rating matrix $\boldsymbol{Z}$, there are exceedingly few elements have been observed, the observed dataset is named $\boldsymbol{R}$. Consequently, the problem of collaborative filtering based recommendation system is constructing an estimator $\hat{r}_{ij}$ to minimize the predictive error in $\boldsymbol{R}$ and generating the prediction for each unobserved element in $\boldsymbol{Z}$.

The probabilistic matrix factorization (PMF) model builds two latent-factor vectors for specific user and item respectively. $\boldsymbol{U}_{i*}$ and $\boldsymbol{V}_{j*}$ represent the latent factor vectors for user i and item j. In PMF, the physical meanings of $\boldsymbol{U}_{i*}$ and $\boldsymbol{V}_{j*}$ are the preference of user i and item j on a set of latent factors separately. In order to estimate the values of $\boldsymbol{U}_{i*}$ and $\boldsymbol{V}_{j*}$, the PMF constructs a linear estimator to predict r_{ij}, which is given by $r_{ij} \approx \hat{r}_{ij} = \boldsymbol{U}_{i*} \times \boldsymbol{V}_{j*}^{\mathrm{T}}$. And the observation noise is assumed to be Gaussian. Then, the objective function of PMF can be derived by the maximum a posterior (MAP) rule with the observed dataset $\boldsymbol{R}$ and the standard Gaussian priors of user and item latent factors.

Based on the classic PMF model [19], many modified models have been proposed. A part of them introduce bias or implicit data to improve the PMF model, such as BiasSVD [20], SVD++ [21], etc. Moreover, plenty of researchers focus on developing superior priors for $\boldsymbol{U}_{i*}$ and $\boldsymbol{V}_{j*}$ in PMF. Instead of suppositional prior like standard Gaussian prior or standard Laplacian prior, researchers are inclined to infer the priors from the additional information of users or items. ConvMF [6] introduced the synopsises of movie to infer the priors of item latent factors by convolutional neural networks. DLTSR [22] and ReDa [14] utilized historical behaviors of user to infer the priors of latent factors by neural autoencoders. Reviews of items are adopted to infer the priors of latent factors in NARRE [7]. The empirical results of aforementioned models provide strong evidence that the additional information based prior is more logical than the suppositional prior and it is able to enhance the model performance effectively.

2.2 Fully Bayesian Matrix Factorization

In PMF, the latent factor vectors, $\boldsymbol{U}_{i*}$ and $\boldsymbol{V}_{j*}$, are estimated by the MAP rule. The MAP rule is the quintessential point estimation method. Thus, the outputs of the PMF are the most likely values of $\boldsymbol{U}_{i*}$ and $\boldsymbol{V}_{j*}$. Other than the PMF model, fully Bayesian matrix factorization (FBMF) is the fully Bayesian treatment of matrix factorization model. FBMF assumes $\boldsymbol{U}_{i*}$ and $\boldsymbol{V}_{j*}$ are random variables and the hyperparameters as well.

In recent years, a number of FBMF models have been delivered. BPMF [23] is the emblematic FBMF model. BPMF hypothesizes $\boldsymbol{U}_{i*}$ and $\boldsymbol{V}_{j*}$ are Gaussian, moreover the hyperparameters in the probability density functions (PDFs) of $\boldsymbol{U}_{i*}$ and $\boldsymbol{V}_{j*}$ are assumed to be Gaussian-Wishart distributed. SPMF [24] exploited the sparse and long-tail features of data to build the model. SPMF set $\boldsymbol{U}_{i*}$ and $\boldsymbol{V}_{j*}$ as Laplacian and the scales of Laplace distribution obey the Generalized Inverse Gaussian distribution. These studies suggest that FBMF models have achieved higher prediction accuracy than the classic PMF models, moreover the FBMF models have better robustness than the PMF models on the overfitting problem, especially when the dimension of latent factors increases. The FBMF models introduce the Markov chain Monte Carlo methods, like Gibbs sampling, for approximate inference in most cases. Unfortunately, Markov chain Monte Carlo methods are rarely used on large scale problems because they are perceived to be very slow by practitioners.

Considering the above analyses, in order to absorb all the advantages in previous researches, the proposed DVMF has the following characteristics. First, DVMF is a FBMF model, it assumes the user and item latent factors as random variables. Then, unlike the traditional FBMF models, the hyperparameters in the PDFs of $\boldsymbol{U}_{i*}$ and $\boldsymbol{V}_{j*}$ are assumed to be inferred from the additional information, rather than set factitiously. Finally, the DVMF model employes the variational inference technique and the reparameterization tricks to make DVMF can be optimized by the stochastic

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2952849, IEEE Transactions on Knowledge and Data Engineering

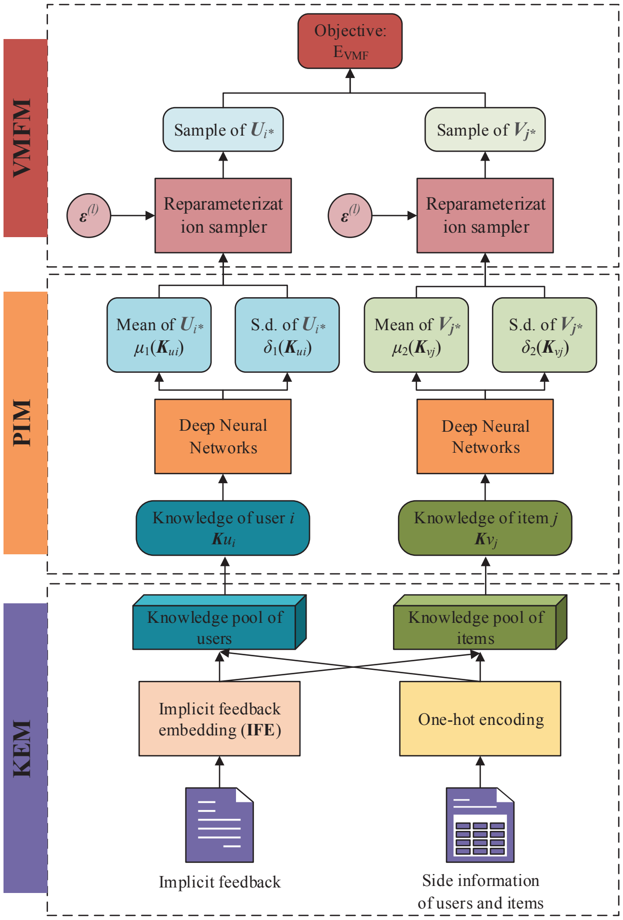


Fig. 1. The outline of DVMF

gradient-based methods. Hence, DVMF could be readily implemented on large scale recommendation problems.

3 Proposed Method

3.1 Outline of DVMF

The proposed method, DVMF, can be summed up as three primary submodels, which are variational matrix factorization model (VMFM), parametric inference model (PIM) and knowledge embedding model (KEM). In knowledge embedding model, it embeds source information into user and item knowledge pools. Then parametric inference model can be workable with the knowledge pools. Finally, the variational matrix factorization model can be operated with the hyper-parameters inferred by PIM.

In VMFM, samples of latent factors are produced by the reparameterization samplers with the hyperparameters generated by PIM. Then the objective function of DVMF could be optimized by these generated samples, meanwhile the unobserved ratings can be predicted.

In PIM, the hyper-parameters, in latent factors' PDFs, are generated by two deep neural networks. These two deep neural networks are constructed as mapping functions to extract information from encoded knowledge of each user and item.

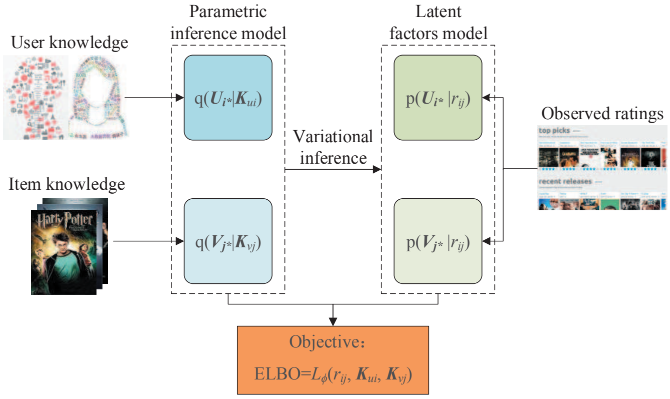


Fig. 2. Schematic of variational matrix factorization model

In KEM, implicit feedback and side information of user and item are considered as source information in this paper. Furthermore, implicit feedback information is encoded by the implicit feedback embedding (IFE) method while the side information encrypted into one-hot modality. The knowledge pools of user and item can be constituted by these two sources of encoded information.

3.2 Variational Matrix Factorization Model

The core objective of matrix factorization based recommendation algorithm is to estimate logical distributions of random variables i.e. $\boldsymbol{U}_{i*}$ and $\boldsymbol{V}_{j*}$. The most common idea is estimating $\boldsymbol{U}_{i*}$ and $\boldsymbol{V}_{j*}$ by the observed rating data r_{ij} namely $p(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}|r_{ij})$. Due to the sparsity of observed rating data, it is formidable to achieve the high-quality $p(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}|r_{ij})$ in most cases. Accordingly, in DVMF, a new method to estimate $\boldsymbol{U}_{i*}$ and $\boldsymbol{V}_{j*}$ is proposed. In the first place, the knowledge of user i and item j are represented as $\boldsymbol{K}_{ui}$ and $\boldsymbol{K}_{vj}$, then the knowledge of user and item are used to calculate the distributions of $\boldsymbol{U}_{i*}$ and $\boldsymbol{V}_{j*}$, i.e. $q(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}|\boldsymbol{K}_{ui}, \boldsymbol{K}_{vj})$, then $q(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}|\boldsymbol{K}_{ui}, \boldsymbol{K}_{vj})$ is applied to approach the $p(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}|r_{ij})$, which is shown in Fig.2. $q(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}|\boldsymbol{K}_{ui}, \boldsymbol{K}_{vj})$ is also called parametric inference model, which will be introduced in section 3.3.

In VMFM, the smaller the gap between $q(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}|\boldsymbol{K}_{ui}, \boldsymbol{K}_{vj})$ and $p(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}|r_{ij})$, the better model performance would achieve. The Kullback-Leibler (KL) divergence is introduced to measure the difference between two distributions. Specifically, the KL divergence between $q(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}|\boldsymbol{K}_{ui}, \boldsymbol{K}_{vj})$ and $p(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}|r_{ij})$ is as follows

$$\begin{aligned}
&\mathrm{KL}\left(q_\phi(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}|\boldsymbol{K}_{ui}, \boldsymbol{K}_{vj})||p(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}|r_{ij})\right)\\
=&\mathbb{E}_{q_\phi(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}|\boldsymbol{K}_{ui}, \boldsymbol{K}_{vj})} \log \frac{q_\phi(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}|\boldsymbol{K}_{ui}, \boldsymbol{K}_{vj})p(r_{ij})}{p(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}|r_{ij})p(r_{ij})}\\
=&\mathbb{E}_{q_\phi(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}|\boldsymbol{K}_{ui}, \boldsymbol{K}_{vj})} \log \frac{q_\phi(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}|\boldsymbol{K}_{ui}, \boldsymbol{K}_{vj})}{p(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}, r_{ij})}\\
&+\mathbb{E}_{q_\phi(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}|\boldsymbol{K}_{ui}, \boldsymbol{K}_{vj})} \log p(r_{ij})\\
=&\mathbb{E}_{q_\phi(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}|\boldsymbol{K}_{ui}, \boldsymbol{K}_{vj})} \log \frac{q_\phi(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}|\boldsymbol{K}_{ui}, \boldsymbol{K}_{vj})}{p(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}, r_{ij})} + \log p(r_{ij})
\end{aligned} \tag{1}$$

thus,

$$\begin{aligned}&\log p(r_{ij}) - \mathrm{KL}\left(q_\phi(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}|\boldsymbol{K}_{ui}, \boldsymbol{K}_{vj})||p(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}|r_{ij})\right)\\ =&L_\phi(r_{ij}, \boldsymbol{K}_{ui}, \boldsymbol{K}_{vj})\end{aligned} \tag{2}$$

where we use $L_\phi(r_{ij}, \boldsymbol{K}_{ui}, \boldsymbol{K}_{vj})$ to denote $-\mathbb{E}_{q_\phi(\boldsymbol{U}_{i*},\boldsymbol{V}_{j*}|\boldsymbol{K}_{ui},\boldsymbol{K}_{vj})} \log \dfrac{q_\phi(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}|\boldsymbol{K}_{ui}, \boldsymbol{K}_{vj})}{p(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}, r_{ij})}$ and ϕ means the parameters in PIM.

Because of the non-negativity of KL divergence, it is obvious that

$$\log p(r_{ij}) \geq L_\phi(r_{ij}, \boldsymbol{K}_{ui}, \boldsymbol{K}_{vj}) \tag{3}$$

thus, $L_\phi(r_{ij}, \boldsymbol{K}_{ui}, \boldsymbol{K}_{vj})$ is one of the lower bound of $\log p(r_{ij})$. $\log p(r_{ij})$ is the log-likelihood of observed data. In general, $\log p(r_{ij})$ is difficult to calculate directly, consequently we maximize the lower bound of $\log p(r_{ij})$, $L_\phi(r_{ij}, \boldsymbol{K}_{ui}, \boldsymbol{K}_{vj})$is commonly referred to as evidence lower bound(ELBO). Then,

$$\begin{aligned}&L_\phi(r_{ij}, \boldsymbol{K}_{ui}, \boldsymbol{K}_{vj})\\ =&-\mathbb{E}_{q_\phi(\boldsymbol{U}_{i*},\boldsymbol{V}_{j*}|\boldsymbol{K}_{ui},\boldsymbol{K}_{vj})} \log \frac{q_\phi(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}|\boldsymbol{K}_{ui}, \boldsymbol{K}_{vj})}{p(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}, r_{ij})}\\ =&-\mathbb{E}_{q_\phi(\boldsymbol{U}_{i*},\boldsymbol{V}_{j*}|\boldsymbol{K}_{ui},\boldsymbol{K}_{vj})} \log \frac{q_\phi(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}|\boldsymbol{K}_{ui}, \boldsymbol{K}_{vj})}{p(r_{ij}|\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*})p(\boldsymbol{U}_{i*}, V_{j*})}\\ =&\mathbb{E}_{q_\phi(\boldsymbol{U}_{i*},\boldsymbol{V}_{j*}|\boldsymbol{K}_{ui},\boldsymbol{K}_{vj})}[\log p(r_{ij}|\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}) + \log p(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*})\\ &- \log q_\phi(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}|\boldsymbol{K}_{ui}, \boldsymbol{K}_{vj})]\\ =&\mathbb{E}_{q_\phi(\boldsymbol{U}_{i*},\boldsymbol{V}_{j*}|\boldsymbol{K}_{ui},\boldsymbol{K}_{vj})} \log p(r_{ij}|\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*})\\ &- \mathrm{KL}(q_\phi(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}|\boldsymbol{K}_{ui}, \boldsymbol{K}_{vj})||p(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}))\end{aligned} \tag{4}$$

In order to simplify the ELBO in Eq. (4), some assumptions have been made in VMFM as follows

- ***Assumption 1.*** All the variables in $\boldsymbol{U}_{i*}$ and $\boldsymbol{V}_{j*}$ are independent.
- ***Assumption 2.*** All the variables in $\boldsymbol{U}_{i*}$ and $\boldsymbol{V}_{j*}$ are Gaussian.
- ***Assumption 3.*** $\mathcal{N}(0, \boldsymbol{I})$ is the prior distribution of all the variables in $\boldsymbol{U}_{i*}$ and $\boldsymbol{V}_{j*}$.
- ***Assumption 4.*** The prediction error in VMFM is Gaussian.

According to the *assumption 1*, we can get that $q_\phi(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}|\boldsymbol{K}_{ui}, \boldsymbol{K}_{vj})) = q_{\phi1}(\boldsymbol{U}_{i*}|\boldsymbol{K}_{ui}) \cdot q_{\phi2}(\boldsymbol{V}_{j*}|\boldsymbol{K}_{vj})$ and $p(\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*}) = p(\boldsymbol{U}_{i*}) \cdot p(\boldsymbol{V}_{j*})$, then the Eq. (4) can be written as

$$\begin{aligned}&L_\phi(r_{ij}, \boldsymbol{K}_{ui}, \boldsymbol{K}_{vj})\\ =&\mathbb{E}_{q_{\phi1}(\boldsymbol{U}_{i*}|\boldsymbol{K}_{ui})q_{\phi2}(\boldsymbol{V}_{j*}|\boldsymbol{K}_{vj})} \log p(r_{ij}|\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*})\\ &- \mathrm{KL}(q_{\phi1}(\boldsymbol{U}_{i*}|\boldsymbol{K}_{ui})||p(\boldsymbol{U}_{i*})) - \mathrm{KL}(q_{\phi2}(\boldsymbol{V}_{j*}|\boldsymbol{K}_{vj})||p(\boldsymbol{V}_{j*}))\end{aligned} \tag{5}$$

Furthermore, $q_{\phi1}(\boldsymbol{U}_{i*}|\boldsymbol{K}_{ui}) = \mathcal{N}(\mu_U(\boldsymbol{K}_{ui}), \delta_U(\boldsymbol{K}_{ui})^2\boldsymbol{I})$ and $q_{\phi2}(\boldsymbol{V}_{j*}|\boldsymbol{K}_{vj}) = \mathcal{N}(\mu_V(\boldsymbol{K}_{vj}), \delta_V(\boldsymbol{K}_{vj})^2\boldsymbol{I})$ are workable with the *assumption 2*, where $\mu_U(\boldsymbol{K}_{ui})$, $\delta_U(\boldsymbol{K}_{ui})\boldsymbol{I}$, $\mu_V(\boldsymbol{K}_{vj})$ and $\delta_V(\boldsymbol{K}_{vj})\boldsymbol{I}$ denote the mean value and standard deviation for user and item latent factors respectively, generated by PIM. With the *assumption 3*, $p(\boldsymbol{U}_{i*}) = \mathcal{N}(0, \boldsymbol{I})$ and $p(\boldsymbol{V}_{j*}) = \mathcal{N}(0, \boldsymbol{I})$ will be established. By that, the last two terms in Eq. (5) can be simplified as

$$\begin{aligned}&\mathrm{KL}(q_{\phi1}(\boldsymbol{U}_{i*}|\boldsymbol{K}_{ui})||p(\boldsymbol{U}_{i*}))\\ =&\mathrm{KL}(\mathcal{N}(\mu_U(\boldsymbol{K}_{ui}), \delta_U(\boldsymbol{K}_{ui})^2\boldsymbol{I})||\mathcal{N}(0, \boldsymbol{I}))\\ =&-\frac{1}{2}\sum_{s=1}^{d}\left[1 + \log\left(\delta_U(\boldsymbol{K}_{ui})_s^2\right) - \delta_U(\boldsymbol{K}_{ui})_s^2 - \mu_U(\boldsymbol{K}_{ui})_s^2\right]\end{aligned} \tag{6}$$

in the similar way,

$$\begin{aligned}&\mathrm{KL}(q_{\phi2}(\boldsymbol{V}_{j*}|\boldsymbol{K}_{vj})||p(\boldsymbol{V}_{j*}))\\ =&-\frac{1}{2}\sum_{s=1}^{d}\left[1 + \log\left(\delta_V(\boldsymbol{K}_{vj})_s^2\right) - \delta_V(\boldsymbol{K}_{vj})_s^2 - \mu_V(\boldsymbol{K}_{vj})_s^2\right]\end{aligned} \tag{7}$$

where d is the dimension of $\boldsymbol{U}_{i*}$ and $\boldsymbol{V}_{j*}$.

It is difficult to deduce the first term in Eq. (5) directly, so the Monte Carlo method is introduced to approximate it instead, which is as follows

$$\begin{aligned}&\mathbb{E}_{q_{\phi1}(\boldsymbol{U}_{i*}|\boldsymbol{K}_{ui})q_{\phi2}(\boldsymbol{V}_{j*}|\boldsymbol{K}_{vj})} \log p(r_{ij}|\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*})\\ \approx&\frac{1}{N}\sum_{l=1}^{N}\log p\left(r_{ij}|\boldsymbol{U}_{i*}^{(l)}, \boldsymbol{V}_{j*}^{(l)}\right)\end{aligned} \tag{8}$$

where $\boldsymbol{U}_{i*}^{(l)} \sim q_{\phi1}(\boldsymbol{U}_{i*}|\boldsymbol{K}_{ui})$ and $\boldsymbol{V}_{j*}^{(l)} \sim q_{\phi2}(\boldsymbol{V}_{j*}|\boldsymbol{K}_{vj})$, in addition, N denote the sampling frequency in Monte Carlo method. Further, we use the reparameterization trick to build the sampler which is given by

$$\begin{cases}\boldsymbol{U}_{i*}^{(l)} = \mu_U(\boldsymbol{K}_{ui}) + \delta_U(\boldsymbol{K}_{ui}) \cdot \varepsilon_U^{(l)},\\ \boldsymbol{V}_{j*}^{(l)} = \mu_V(\boldsymbol{K}_{vj}) + \delta_V(\boldsymbol{K}_{vj}) \cdot \varepsilon_V^{(l)},\\ \varepsilon_U^{(l)}, \varepsilon_V^{(l)} \sim \mathcal{N}(0, \boldsymbol{I}),\end{cases} \tag{9}$$

By the reparameterization sampler the random part and the parametric part in samples can be separated. It makes the model is able to optimize by the gradient-based method, such as gradient descent, Adam [25], etc. On the basis of the *assumption 4*, the log-likelihood of observed data can be represented as

$$\begin{aligned}&p\left(r_{ij}|\boldsymbol{U}_{i*}^{(l)}, \boldsymbol{V}_{j*}^{(l)}\right)\\ =&\log\frac{1}{\sqrt{2\pi}\delta}\exp\left(-\frac{\left(r_{ij} - \boldsymbol{U}_{i*}^{(l)} \cdot {\boldsymbol{V}_{j*}^{(l)}}^{\mathrm{T}} - G_{ui} - G_{vj}\right)^2}{2\delta^2}\right)\\ =&-\frac{1}{2\delta^2}\left(r_{ij} - \boldsymbol{U}_{i*}^{(l)} \cdot {\boldsymbol{V}_{j*}^{(l)}}^{\mathrm{T}} - G_{ui} - G_{vj}\right)^2 - \log\sqrt{2\pi}\delta\\ =&-\frac{1}{2\delta^2}\Bigg(r_{ij} - \left(\mu_U(\boldsymbol{K}_{ui}) + \delta_U(\boldsymbol{K}_{ui}) \cdot \varepsilon_U^{(l)}\right) \cdot (\mu_V(\boldsymbol{K}_{vj}) +\\ &\delta_V(\boldsymbol{K}_{vj}) \cdot \varepsilon_U^{(l)}\Big)^{\mathrm{T}} - G_{ui} - G_{vj}\Bigg)^2 - \log\sqrt{2\pi}\delta\end{aligned} \tag{10}$$

where G_{ui} and G_{vj} mean the global effects of user i and item j, and the Eq. (8) can be presented as

$$\begin{aligned}&\mathbb{E}_{q_{\phi1}(\boldsymbol{U}_{i*}|\boldsymbol{K}_{ui})q_{\phi2}(\boldsymbol{V}_{j*}|\boldsymbol{K}_{vj})} \log p(r_{ij}|\boldsymbol{U}_{i*}, \boldsymbol{V}_{j*})\\ \approx&-\frac{1}{2\delta^2}\Bigg(r_{ij} - \frac{1}{N}\sum_{l=1}^{N}\left(\mu_U(\boldsymbol{K}_{ui}) + \delta_U(\boldsymbol{K}_{ui}) \cdot \varepsilon_U^{(l)}\right) \cdot\\ &\left(\mu_V(\boldsymbol{K}_{vj}) + \delta_V(\boldsymbol{K}_{vj}) \cdot \varepsilon_V^{(l)}\right)^{\mathrm{T}} - G_{ui} - G_{vj}\Bigg)^2 - \log\sqrt{2\pi}\delta\end{aligned} \tag{11}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2952849, IEEE Transactions on Knowledge and Data Engineering

Combining Eq. (5), (6), (7) and (11), $L_\phi(r_{ij},\boldsymbol{K}_{ui},\boldsymbol{K}_{vj})$ can be written as

$$\begin{aligned}
&L_\phi(r_{ij},\boldsymbol{K}_{ui},\boldsymbol{K}_{vj})\\
&\approx -\frac{1}{2\delta^2}\Bigg(r_{ij}-\frac{1}{N}\sum_{l=1}^{N}\left(\mu_U(\boldsymbol{K}_{ui})+\delta_U(\boldsymbol{K}_{ui})\cdot\varepsilon_U^{(l)}\right)\cdot\Big(\mu_V(\boldsymbol{K}_{vj})\\
&\quad+\delta_V(\boldsymbol{K}_{vj})\cdot\varepsilon_V^{(l)}\Big)^{\mathrm{T}}-G_{ui}-G_{vj}\Bigg)^2-\log\sqrt{2\pi}\delta\\
&\quad+\frac{1}{2}\sum_{s=1}^{d}\left[1+\log\left(\delta_U(\boldsymbol{K}_{ui})_s^2\right)-\delta_U(\boldsymbol{K}_{ui})_s^2-\mu_U(\boldsymbol{K}_{ui})_s^2\right]\\
&\quad+\frac{1}{2}\sum_{s=1}^{d}\left[1+\log\left(\delta_U(\boldsymbol{K}_{ui})_s^2\right)-\delta_U(\boldsymbol{K}_{ui})_s^2-\mu_U(\boldsymbol{K}_{ui})_s^2\right]
\end{aligned} \tag{12}$$

The ELBO function of observed ratings over training set can be proposed as follows

$$\begin{aligned}
&L_\phi(\boldsymbol{R},\boldsymbol{K}_u,\boldsymbol{K}_v)\\
&\approx -\frac{1}{2\delta^2}\sum_i\sum_j 1_{ij}\Bigg(r_{ij}-\frac{1}{N}\sum_{l=1}^{N}\left(\mu_U(\boldsymbol{K}_{ui})+\delta_U(\boldsymbol{K}_{ui})\cdot\varepsilon_U^{(l)}\right)\\
&\quad\cdot\left(\mu_V(\boldsymbol{K}_{vj})+\delta_V(\boldsymbol{K}_{vj})\cdot\varepsilon_V^{(l)}\right)^{\mathrm{T}}-G_{ui}-G_{vj}\Bigg)^2\\
&\quad+\frac{1}{2}\sum_i\sum_{s=1}^{d}\left[1+\log\left(\delta_U(\boldsymbol{K}_{ui})_s^2\right)-\delta_U(\boldsymbol{K}_{ui})_s^2-\mu_U(\boldsymbol{K}_{ui})_s^2\right]\\
&\quad+\frac{1}{2}\sum_j\sum_{s=1}^{d}\left[1+\log\left(\delta_U(\boldsymbol{K}_{ui})_s^2\right)-\delta_U(\boldsymbol{K}_{ui})_s^2-\mu_U(\boldsymbol{K}_{ui})_s^2\right]\\
&\quad-\sum_i\sum_j 1_{ij}\log\sqrt{2\pi}\delta
\end{aligned} \tag{13}$$

where 1_{ij} is the indicator matrix, in which 1_{ij} is equal to 1 if r_{ij} is observed and equal to 0 otherwise.

$$\begin{aligned}
\mathrm{E}_{\mathrm{VMF}}&=\sum_i\sum_j 1_{ij}\Bigg(r_{ij}-\frac{1}{N}\sum_{l=1}^{N}\left(\mu_U(\boldsymbol{K}_{ui})+\delta_U(\boldsymbol{K}_{ui})\cdot\varepsilon_U^{(l)}\right)\\
&\cdot\left(\mu_V(\boldsymbol{K}_{vj})+\delta_V(\boldsymbol{K}_{vj})\cdot\varepsilon_V^{(l)}\right)^{\mathrm{T}}-G_{ui}-G_{vj}\Bigg)^2\\
&-\lambda\sum_i\sum_{s=1}^{d}\left[1+\log\left(\delta_U(\boldsymbol{K}_{ui})_s^2\right)-\delta_U(\boldsymbol{K}_{ui})_s^2-\mu_U(\boldsymbol{K}_{ui})_s^2\right]\\
&-\lambda\sum_j\sum_{s=1}^{d}\left[1+\log\left(\delta_U(\boldsymbol{K}_{ui})_s^2\right)-\delta_U(\boldsymbol{K}_{ui})_s^2-\mu_U(\boldsymbol{K}_{ui})_s^2\right]
\end{aligned} \tag{14}$$

According to the variational inference framework, $\mu_U(\boldsymbol{K}_{ui})$, $\delta_U(\boldsymbol{K}_{ui})$, $\mu_V(\boldsymbol{K}_{vj})$, $\delta_V(\boldsymbol{K}_{vj})$, G_{ui} and G_{vj} can be estimated by maximizing Eq. (13) which is equivalent to minimizing the objective function Eq. (14). In the next section, the parametric inference model will be expatiated which is denoted as $\mu_U(\boldsymbol{K}_{ui})$, $\delta_U(\boldsymbol{K}_{ui})$, $\mu_V(\boldsymbol{K}_{vj})$, $\delta_V(\boldsymbol{K}_{vj})$ in this section.

3.3 Parametric Inference Model

In order to construct the $\mu_U(\boldsymbol{K}_{ui})$, $\delta_U(\boldsymbol{K}_{ui})$, $\mu_V(\boldsymbol{K}_{vj})$ and $\delta_V(\boldsymbol{K}_{vj})$ for VMFM, two deep neural networks have been accomplished. These two neural networks compute both mean value and standard deviation for distributions of user and item latent factors with corresponding knowledge as input information respectively, where $\boldsymbol{K}_{ui}$ and $\boldsymbol{K}_{vj}$ are provided by the KEM.

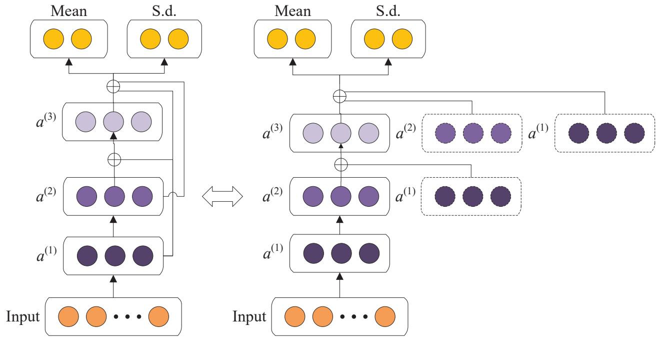


Fig. 3. Architecture of densely-connect multi-layer perceptron, where ⊕ denotes the concatenate operation and the arrow means the forward propagation between layers.

Multi-layer perceptron (MLP) is the most classic neural network architecture. It is an effective model to transform the input data into targets. As the number of layers in MLP get deeper and deeper, the model capacity is boosted accordingly [26], [27], however the vanishing-gradient problem [28] appears simultaneously. Deep MLP structure increases the difficulty of optimizing in gradient propagating process. It makes the model difficult to be trained smoothly and degrades the model performance gravely. Focusing on addressing these problems, with the inspirations of previous works [29], [30], a new neural network architecture is proposed, named densely-connect multi-layer perceptron (D-MLP). In D-MLP, each layer is connected with layers preceding it to ensure maximum information flow between layers in the network. Fig. 3 illustrates this layout schematically. D-MLP gets the utmost out of the features in each hidden layer. Furthermore, D-MLP improves flow of information and gradient throughout the network, which will result in a painless training course. Each layer has direct access to the gradients from the loss function and the original input information.

The modalities of $\mu_U(\boldsymbol{K}_{ui})$, $\delta_U(\boldsymbol{K}_{ui})$, $\mu_V(\boldsymbol{K}_{vj})$ and $\delta_V(\boldsymbol{K}_{vj})$ depend on the hidden structure of D-MLP. Therefore, we introduce the recurrence equations of them in this section. The recurrence equations of $\mu_U(\boldsymbol{K}_{ui})$ and $\delta_U(\boldsymbol{K}_{ui})$ are as follows:

$$\begin{cases}
\boldsymbol{a}_U^{(1)}=\boldsymbol{K}_{ui}\\
\boldsymbol{m}_U^{(\zeta)}\sim Bernoulli(\varphi)\\
\tilde{\boldsymbol{a}}_U^{(\zeta)}=\boldsymbol{a}_U^{(\zeta)}\circ\boldsymbol{m}_U^{(\zeta)}\\
\hat{\boldsymbol{a}}_U^{(\zeta)}=\tilde{\boldsymbol{a}}_U^{(\zeta)}\oplus\hat{\boldsymbol{a}}_U^{(\zeta-1)}\\
\qquad=\hat{\boldsymbol{a}}_U^{(\zeta)}\oplus\hat{\boldsymbol{a}}_U^{(\zeta-1)}\oplus\cdots\hat{\boldsymbol{a}}_U^{(1)}\\
\boldsymbol{a}_U^{(\zeta+1)}=\sigma\left(\hat{\boldsymbol{a}}_U^{(\zeta)}\cdot\mathbf{W}_U^{(\zeta+1)}+\boldsymbol{b}_U^{(\zeta+1)}\right),\zeta\in\{1,2,...,L-2\}\\
\mu_U(\boldsymbol{K}_{ui})=\sigma\left(\hat{\boldsymbol{a}}_U^{(L-1)}\cdot\mathbf{W}_{\mu U}^{(L)}+\boldsymbol{b}_{\mu U}^{(L)}\right)\\
\delta_U(\boldsymbol{K}_{ui})=\sigma\left(\hat{\boldsymbol{a}}_U^{(L-1)}\cdot\mathbf{W}_{\delta U}^{(L)}+\boldsymbol{b}_{\delta U}^{(L)}\right)
\end{cases} \tag{15}$$

We define $\mu_U(\boldsymbol{K}_{ui})$ and $\delta_U(\boldsymbol{K}_{ui})$ as a L-layers D-MLP. Where $\boldsymbol{a}_U^{(\zeta)}$ means the activations of the ζ-th layer and $\oplus$ is the concatenating operation. We also use $\boldsymbol{a}_U^{(1)}$ to denote the values from the input layer and $\mu_U(\boldsymbol{K}_{ui})$, $\delta_U(\boldsymbol{K}_{ui})$ denote the output of D-MLP. $\boldsymbol{W}_U^{(\zeta)}$ and $\boldsymbol{b}_U^{(\zeta)}$ are the weight and bias in layer ζ. The dropout technique [31] is adopted to enhance the generalization ability of D-MLP. $\boldsymbol{m}_U^{(\zeta)}$ represents the dropout mask in ζ-th layer and φ is the keep rate in dropout. The symbol $\circ$ means the Hadamard product between two matrices. In PIM, rectified linear unit (ReLU) [32] is introduced as the active function. In addition, it can be expressed as $\mathrm{ReLU}(x) = \max(0, x)$. Relatively, the recurrence equations of $\mu_V(\boldsymbol{K}_{vj})$ and $\delta_V(\boldsymbol{K}_{vj})$ can be described in the same manner.

In PIM, the input $\boldsymbol{K}_{ui}$ and $\boldsymbol{K}_{vj}$ may have different scales, but the output of D-MLP must have the same dimensions which is equal to the number of user and item latent factors. Accordingly, we set the same hidden structures for two D-MLPs in this article. Moreover, the method of building $\boldsymbol{K}_{ui}$ and $\boldsymbol{K}_{vj}$, namely knowledge embedding model, will be introduced in the next section.

3.4 Knowledge Embedding Model

Two kinds of source information, implicit feedback and side information, are introduced to construct $\boldsymbol{K}_{ui}$ and $\boldsymbol{K}_{vj}$. Implicit feedback indicates the user attitude to an item, but not as directly as rating behaviors. Both positive feedback (e.g., clicks, purchases) and negative feedback (e.g., blacklist, unlike) are considered as implicit feedback. Along with the user historical behaviors, There are a wealth of information which is conducive to predict the unobserved rating, such as user profile, item profile and so forth. Such information are called side information.

In our article, $\boldsymbol{K}_{ui}$ and $\boldsymbol{K}_{vj}$ can be built by combining implicit feedback and side information. The general formula of KEM is given by

$$\begin{cases} \boldsymbol{K}_{ui} = \boldsymbol{\theta}_{ui} \oplus \boldsymbol{\xi}_{ui} \oplus \cdots \\ \boldsymbol{K}_{vj} = \boldsymbol{\theta}_{vj} \oplus \boldsymbol{\xi}_{vj} \oplus \cdots \end{cases} \tag{16}$$

where $\oplus$ is the concatenation operator, $\boldsymbol{\theta}_{ui}$ and $\boldsymbol{\theta}_{vj}$ denote the embeddings of implicit feedbacks about user i and item j respectively, $\boldsymbol{\xi}_{ui}$ and $\boldsymbol{\xi}_{vj}$ are the corresponding side information. Furthermore, $\boldsymbol{K}_{ui}$ and $\boldsymbol{K}_{vj}$ are flexible, which means it can incorporate with any useful information, such as content information of items, social information of users, etc. Consequently DVMF has the capability to convey diversified information for the recommendation model.

The side information of user and item can be converted by one-hot encoding easily. So we will focus on how to represent the implicit feedback information. Normally, implicit feedback information is encoded in its adjacent matrix [14], [22], which is an extremely high-dimensional and sparse vector and it is difficult for learning algorithms to extract information effectively from it. The implicit feedback embedding (IFE) method is proposed to address this issue. By employing the IFE method, implicit feedback could be represented as a low-dimensional real-valued vector and preserves the primary features to a certain extent. In addition, it could vastly reduce the scale of model parameters and increase training efficiency. Firstly, the implicit feedback and IFE can be defined as follows.

Definition 1: Implicit feedback. Implicit feedback means the historical behaviors between user and item. Implicit feedback could be stated as an undirected graph $G = (U, I, E)$, where U and I denote the user set and item set, respectively. E is the set of edges and E is only connected between nodes in U and I.

In implicit feedback graph, whether edges are weighted or not depends on the type of historical behavior. In this article, we consider the implicit feedback graph as an unweighted and undirected graph. Only positive feedbacks are included in our model and rating behaviors are treated as a kind of positive implicit feedback.

Definition 2: Implicit feedback embedding. Given an implicit feedback graph $G = (U, I, E)$, IFE aims to learn a function f: u and $v \to \mathbb{R}^k$, $u \in U$ and $v \in I$. It projects each user and item into a vector in k-dimensional space, where $k \ll |U|$ and $k \ll |V|$.

The vectors $\boldsymbol{\theta}_{ui}$ and $\boldsymbol{\theta}_{vj}$ are introduced to denote the embeddings of user i and item j in IFE, where $\boldsymbol{\theta}_{ui} \in \mathbb{R}^k$ and $\boldsymbol{\theta}_{vj} \in \mathbb{R}^k$. The symbol k is the dimensionality of user and item embedding. IFE models the probability of incident ″whether user i has positive feedback about item j″, which is given by,

$$\begin{aligned} p\left(E_{ui,vj} \in \boldsymbol{G}\right) =& \mathrm{h}\left(\boldsymbol{\theta}_{ui} \cdot \boldsymbol{\theta}_{vj}^{\mathrm{T}}\right) \\ =& \frac{1}{1+\exp\left(-\boldsymbol{\theta}_{ui} \cdot \boldsymbol{\theta}_{vj}^{\mathrm{T}}\right)} \end{aligned} \tag{17}$$

where $\boldsymbol{G}$ is the implicit feedback graph, $E_{ui,vj}$ means the edge between the user i and the item j.

Negative feedback is an essential part in estimating user and item embeddings. Because of the difficulty of collecting negative feedback data, noise contrastive estimation [33] and negative sampling [34] is introduced to generate and estimate the probability of negative feedback for IFE. The probability of negative term is given as follows

$$p\left(E_{ui,vj} \in \boldsymbol{S}_{neg}\right) = 1 - \mathrm{h}\left(\boldsymbol{\theta}_{ui} \cdot \boldsymbol{\theta}_{vj}^{\mathrm{T}}\right) \tag{18}$$

where $\boldsymbol{S}_{neg}$ means the negative feedback graph, which is generated by negative sampling. The probability of the user and item sampled in negative sampling is proportional to its frequency in training sets. The likelihood function for training sets is proposed as

$$p\left(\boldsymbol{G}, \boldsymbol{S}_{neg} | \boldsymbol{\theta}_u, \boldsymbol{\theta}_v\right) = \prod_{E_{ui,vj} \in \boldsymbol{G}} \mathrm{h}\left(\boldsymbol{\theta}_{ui} \cdot \boldsymbol{\theta}_{vj}^{\mathrm{T}}\right) \cdot \prod_{E_{ui,vj} \in \boldsymbol{S}_{neg}} \left(1 - \mathrm{h}\left(\boldsymbol{\theta}_{ui} \cdot \boldsymbol{\theta}_{vj}^{\mathrm{T}}\right)\right) \tag{19}$$

where $\boldsymbol{\theta}_u$ and $\boldsymbol{\theta}_v$ indicate the embeddings of all user and item. The logarithmic form of Eq. (19) can be given

$$\begin{aligned} \mathrm{E}_{\mathrm{IFE}} =& \log p\left(\boldsymbol{G}, \boldsymbol{S}_{neg} | \boldsymbol{\theta}_u, \boldsymbol{\theta}_v\right) \\ =& \sum_{E_{ui,vj} \in \boldsymbol{G}} \log \mathrm{h}\left(\boldsymbol{\theta}_{ui} \cdot \boldsymbol{\theta}_{vj}^{\mathrm{T}}\right) + \sum_{E_{ui,vj} \in \boldsymbol{S}_{neg}} \log\left(1 - \mathrm{h}\left(\boldsymbol{\theta}_{ui} \cdot \boldsymbol{\theta}_{vj}^{\mathrm{T}}\right)\right) \end{aligned} \tag{20}$$

According to the maximum likelihood estimation rule, embeddings of user and item could be estimated by maximizing $\mathrm{E}_{\mathrm{IFE}}$, which is illustrated in Eq. (20).

4 Optimization and Parameter Determination

4.1 Optimize the VMFM and PIM

To address the large-scale representation learning problem, we adopt mini-batch gradient descent (MBGD) algorithm to optimize the objective function in Eq. (14). According to the MBGD strategy, we update $\mu_U(\boldsymbol{K}_{ui})$, $\delta_U(\boldsymbol{K}_{ui})$, $\mu_V(\boldsymbol{K}_{vj})$, $\delta_V(\boldsymbol{K}_{vj})$, G_{ui} and G_{vj} respectively, by the update rule in Eq. (21).

$$\mathop{\arg\max}_{\mu_U(\boldsymbol{K}_{ui}),\delta_U(\boldsymbol{K}_{ui}),\mu_V(\boldsymbol{K}_{vj}),\delta_V(\boldsymbol{K}_{vj}),G_{ui},G_{vj}} \quad \mathrm{E}_{\mathrm{VMF}}$$
$$\Rightarrow \begin{cases} \begin{aligned} G_{ui} \leftarrow & G_{ui} - \alpha \cdot \frac{\partial \mathrm{E}_{\mathrm{VMF}}}{\partial G_{ui}} \\ = & G_{ui} - \alpha \cdot \sum_{i} 1_{ij}(r_{ij} - \hat{r}_{ij}) \\ G_{vj} \leftarrow & G_{vj} - \alpha \cdot \frac{\partial \mathrm{E}_{\mathrm{VMF}}}{\partial G_{vj}} \\ = & G_{vj} - \alpha \cdot \sum_{j} 1_{ij}(r_{ij} - \hat{r}_{ij}) \\ \mu_U(\boldsymbol{K}_{ui}) \leftarrow & \mu_U(\boldsymbol{K}_{ui}) - \alpha \cdot \frac{\partial \mathrm{E}_{\mathrm{VMF}}}{\partial \mu_U(\boldsymbol{K}_{ui})} \\ \delta_U(\boldsymbol{K}_{ui}) \leftarrow & \delta_U(\boldsymbol{K}_{ui}) - \alpha \cdot \frac{\partial \mathrm{E}_{\mathrm{VMF}}}{\partial \delta_U(\boldsymbol{K}_{ui})} \\ \mu_V(\boldsymbol{K}_{vj}) \leftarrow & \mu_V(\boldsymbol{K}_{vj}) - \alpha \cdot \frac{\partial \mathrm{E}_{\mathrm{VMF}}}{\partial \mu_V(\boldsymbol{K}_{vj})} \\ \delta_V(\boldsymbol{K}_{vj}) \leftarrow & \delta_V(\boldsymbol{K}_{vj}) - \alpha \cdot \frac{\partial \mathrm{E}_{\mathrm{VMF}}}{\partial \delta_V(\boldsymbol{K}_{vj})} \end{aligned} \end{cases} \tag{21}$$

where α denotes the learning rate in MBGD and $\hat{r}_{ij} = \frac{1}{N}\sum_{l=1}^{N}\left(\mu_U(\boldsymbol{K}_{ui}) + \delta_U(\boldsymbol{K}_{ui}) \cdot \varepsilon_U^{(l)}\right) \cdot \left(\mu_V(\boldsymbol{K}_{vj}) + \delta_V(\boldsymbol{K}_{vj}) \cdot \varepsilon_V^{(l)}\right)^{\mathrm{T}} - G_{ui} - G_{vj}$. The partial derivatives of $\mathrm{E}_{\mathrm{VMF}}$ with respect to $\mu_U(\boldsymbol{K}_{ui})$, $\delta_U(\boldsymbol{K}_{ui})$,$\mu_V(\boldsymbol{K}_{vj})$, $\delta_V(\boldsymbol{K}_{vj})$ is given by

$$\begin{cases} \begin{aligned} \frac{\partial \mathrm{E}_{\mathrm{VMF}}}{\partial \mu_U(\boldsymbol{K}_{ui})} = & 2\sum_i\sum_j 1_{ij}(\hat{r}_{ij} - r_{ij}) \cdot \frac{1}{N}\sum_{l=1}^{N}\Big(\mu_V(\boldsymbol{K}_{vj}) \\ & + \delta_V(\boldsymbol{K}_{vj}) \cdot \varepsilon_V^{(l)}\Big) + 2\lambda \cdot \sum_i \mu_U(\boldsymbol{K}_{ui}) \\ \frac{\partial \mathrm{E}_{\mathrm{VMF}}}{\partial \delta_U(\boldsymbol{K}_{ui})} = & 2\sum_i\sum_j 1_{ij}(\hat{r}_{ij} - r_{ij}) \cdot \frac{1}{N}\sum_{l=1}^{N}\Big(\mu_V(\boldsymbol{K}_{vj}) \\ & + \delta_V(\boldsymbol{K}_{vj}) \cdot \varepsilon_V^{(l)}\Big) \cdot \varepsilon_U^{(l)} + 2\lambda \cdot \sum_i \left[\delta_U(\boldsymbol{K}_{ui}) - \frac{1}{\delta_U(\boldsymbol{K}_{ui})}\right] \\ \frac{\partial \mathrm{E}_{\mathrm{VMF}}}{\partial \mu_V(\boldsymbol{K}_{vj})} = & 2\sum_i\sum_j 1_{ij}(\hat{r}_{ij} - r_{ij}) \cdot \frac{1}{N}\sum_{l=1}^{N}\Big(\mu_U(\boldsymbol{K}_{ui}) \\ & + \delta_U(\boldsymbol{K}_{ui}) \cdot \varepsilon_U^{(l)}\Big) + 2\lambda \cdot \sum_i \mu_V(\boldsymbol{K}_{vj}) \\ \frac{\partial \mathrm{E}_{\mathrm{VMF}}}{\partial \delta_V(\boldsymbol{K}_{vj})} = & 2\sum_i\sum_j 1_{ij}(\hat{r}_{ij} - r_{ij}) \cdot \frac{1}{N}\sum_{l=1}^{N}\Big(\mu_U(\boldsymbol{K}_{ui}) \\ & + \delta_U(\boldsymbol{K}_{ui}) \cdot \varepsilon_U^{(l)}\Big) \cdot \varepsilon_V^{(l)} + 2\lambda \cdot \sum_i \left[\delta_V(\boldsymbol{K}_{vj}) - \frac{1}{\delta_V(\boldsymbol{K}_{vj})}\right] \end{aligned} \end{cases} \tag{22}$$

The parameters in $\mu_U(\boldsymbol{K}_{ui})$, $\delta_U(\boldsymbol{K}_{ui})$, $\mu_V(\boldsymbol{K}_{vj})$, $\delta_V(\boldsymbol{K}_{vj})$ can be optimized by back-propagation (BP) algorithm [35] with Eq. (22) straightforward.

4.2 Optimize the KEM

We can optimize the objective function of IFE, shown in Eq. (20), by mini-batch gradient ascent (MBGA) algorithm efficiently which is quite similar to MBGD. The update rule for $\boldsymbol{\theta}_u$ and $\boldsymbol{\theta}_v$ in IFE is given by as follows,

$$\mathop{\arg\max}_{\boldsymbol{\theta}_u,\boldsymbol{\theta}_v} \ \mathrm{E}_{\mathrm{IFE}}$$
$$\Rightarrow \begin{cases} \begin{aligned} \boldsymbol{\theta}_u \leftarrow \boldsymbol{\theta}_u + \alpha \cdot \frac{\partial \mathrm{E}_{\mathrm{IFE}}}{\partial \boldsymbol{\theta}_u} & \\ = \boldsymbol{\theta}_u + \alpha \cdot & \sum_{E_{ui,vj}\in \boldsymbol{G}} \left(1 - h\left(\boldsymbol{\theta}_{ui} \cdot \boldsymbol{\theta}_{vj}^{\mathrm{T}}\right)\right) \cdot \boldsymbol{\theta}_{vj} \\ + \alpha \cdot & \sum_{E_{ui,vj}\in \boldsymbol{S}_{neg}} -h\left(\boldsymbol{\theta}_{ui} \cdot \boldsymbol{\theta}_{vj}^{\mathrm{T}}\right) \cdot \boldsymbol{\theta}_{vj} \\ \boldsymbol{\theta}_v \leftarrow \boldsymbol{\theta}_v + \alpha \cdot \frac{\partial \mathrm{E}_{\mathrm{IFE}}}{\partial \boldsymbol{\theta}_v} & \\ = \boldsymbol{\theta}_v + \alpha \cdot & \sum_{E_{ui,vj}\in \boldsymbol{G}} \left(1 - h\left(\boldsymbol{\theta}_{ui} \cdot \boldsymbol{\theta}_{vj}^{\mathrm{T}}\right)\right) \cdot \boldsymbol{\theta}_{ui} \\ + \alpha \cdot & \sum_{E_{ui,vj}\in \boldsymbol{S}_{neg}} -h\left(\boldsymbol{\theta}_{ui} \cdot \boldsymbol{\theta}_{vj}^{\mathrm{T}}\right) \cdot \boldsymbol{\theta}_{ui} \end{aligned} \end{cases} \tag{23}$$

where α is the learning rate in the gradient-based optimization algorithm, $\{ui, vj\}$ means a mini-batch specimen sampled from $\boldsymbol{G}$ or $\boldsymbol{S}_{neg}$. Combining the one-hot encoded side information, the construction process of KEM can be proposed as

Algorithm 1. Knowledge embedding model

```
Input: implicit feedback graph $\boldsymbol{G}$, side information of user and item
Set: batch size $b$, learning rate $\alpha$, dimensionality $k$
1:Initialize $\boldsymbol{\theta}_u$ and $\boldsymbol{\theta}_v$ randomly
2:while not $\mathrm{E}_{\mathrm{IFE}}$ is converged do:
      sample a mini batch samples from $\boldsymbol{G}$ in size $b$,
      generate a mini batch samples by negative sampling,
      update $\boldsymbol{\theta}_u$ and $\boldsymbol{\theta}_v$ via Eq. (23) with the mini batch
  end while
3:Generate $\boldsymbol{\xi}_{ui}$ and $\boldsymbol{\xi}_{vj}$ by one-hot encoding with the side information of user and item
4:Built $\boldsymbol{K}_{ui}$ and $\boldsymbol{K}_{vj}$ via Eq. (16)
Output: $\boldsymbol{K}_{ui}$ and $\boldsymbol{K}_{vj}$
```

Further, the training process of DVMF algorithm is given by

Algorithm 2. Deep variational matrix factorization

```
Input: implicit feedback graph $\boldsymbol{G}$, rating set $\boldsymbol{R}$, side information $\boldsymbol{S}$
Set: batch size $b$, learning rate $\alpha$, dimensionality $k$
1:Initialize $\mu_U(\boldsymbol{K}_{ui})$, $\delta_U(\boldsymbol{K}_{ui})$, $\mu_V(\boldsymbol{K}_{vj})$, $\delta_V(\boldsymbol{K}_{vj})$, $G_{ui}$ and $G_{vj}$ randomly
2:Built knowledge pool $\boldsymbol{K}_{ui}$ and $\boldsymbol{K}_{vj}$ with $\boldsymbol{G}$ and $\boldsymbol{S}$ by Algorithm 1
3:while not $\mathrm{E}_{\mathrm{VMF}}$ is converged do:
      sample a mini batch from $\boldsymbol{R}$ in size $b$,
      update $G_{ui}$ and $G_{vj}$ via (21) with mini batch,
      parameters in $\mu_U(\boldsymbol{K}_{ui})$, $\delta_U(\boldsymbol{K}_{ui})$, $\mu_V(\boldsymbol{K}_{vj})$, $\delta_V(\boldsymbol{K}_{vj})$
via Eq. (21)-(22) and BP algorithm with mini batch,
  end while
Output: DVMF model
```

4.3 Parameter determination

In VMFM, λ in Eq. (14) is the regularization parameters, which can balance the fidelity term (first term) and regularization terms (last two terms). Furthermore N is sampling frequency in Monte Carlo approximation and it can be set as 1 handily. The deducing is as follows

$$\begin{aligned}&\frac{1}{N}\sum_{l=1}^{N}\left(\mu_U(\boldsymbol{K}_{ui})+\delta_U(\boldsymbol{K}_{ui})\cdot\varepsilon_U^{(l)}\right)=\mu_U(\boldsymbol{K}_{ui})+\delta_U(\boldsymbol{K}_{ui})\cdot\frac{1}{N}\sum_{l=1}^{N}\varepsilon_U^{(l)}\\&p\left(\frac{1}{N}\sum_{l=1}^{N}\varepsilon^{(l)}\right)=p\left(\varepsilon^{(l)}\right)=\mathcal{N}(0,\boldsymbol{I})\end{aligned}\tag{24}$$

The hyper-parameter φ, in PIM, is the parameter in dropout. The value of φ affects the generalization of PIM. The structure and number of neurons in D-MLP control the capacity of PIM. The deeper or wider the structure is, the bigger the capacity PIM will have [26], [27]. In KEM, k denotes the dimensionality of user and item embedding. It controls the capacity of the representation model.

Some approaches have been developed to determine these parameters automatically, such as the discrepancy principle [36], generalized cross-validation [37], and the L-curve method [38]. In this article, the parameters are determined heuristically. We validate a large range of the parameters with the method given in [37]. For the different scale levels of datasets, there are small changes for the optimal parameters. We find that promising performance can be achieved with the parameters $\varphi = 0.5$, $\lambda = 0.001$ and the five layers MLP structure $[k+, a\times k, a\times k, a\times k, 0.5\times a\times k]$, where $k+$ means the dimensionality of input with k-dimensional embedding and one-hot encoded side information, $a \in [2, 4]$.

5 Experiments and Discussion

5.1 General Settings

1) Evaluation Metrics: There are numerous aspects to evaluate a recommendation algorithm, such as the prediction accuracy, coverage, serendipity, and so on. In this article, we mainly consider the error between predictions and the actual ratings, because it can directly demonstrate whether the model is in a position to capture the essential features of training data or not. In our experiments, two popular error metrics are used to measure the prediction accuracy: the mean absolute error (MAE) and the root mean square error (RMSE). The smaller the MAE or RMSE value becomes, the higher the accuracy. MAE and RMSE are defined as

$$RMSE=\sqrt{\frac{1}{|R_{test}|}\sum_{r_{ij}\in R_{test}}\left(r_{ij}-\hat{r}_{ij}\right)^2}\tag{25}$$

and

$$MAE=\frac{1}{|R_{test}|}\sum_{r_{ij}\in R_{test}}|r_{ij}-\hat{r}_{ij}|_{abs}\tag{26}$$

where $|R_{test}|$ denotes the size of the test set, and $|\bullet|_{abs}$ means the absolute value.

2) Tested Models: Eight models are included in our experiment as follows

a) Mean: Each rating is predicted by the average value of the ratings on the training set.

b) PMF: Probabilistic matrix factorization is a baseline matrix factorization model proposed by Salakhutdinov and Minh et al. in [19]. PMF is the most widely used recommendation model.

c) BPMF: Bayesian probabilistic matrix factorization [23] is the baseline of the fully Bayesian matrix factorization model published by Salakhutdinov and Minh.

d) AutoRec: AutoRec is an autoencoder-based recommendation framework, designed by Sedhai and Menon et al. in [13].In this article, we use I-AutoRec as the test model.

e) NADE: Neural autoregressive distribution estimation is submitted by Uria B, Marc-Alexandre, Gregor K, et al. in [15]. And NADE is used to address collaborative filtering problem by Zheng and Tang et al. in [16].

f) DLTSR: Deep learning for long-tail web service recommendations is proposed by Bai B, Fan Y, Tan W, et al. in [22].

g) ReDa: Representation learning via dual-autoencoder for recommendation is a dual-autoencoder based recommendation algorithm designed by Zhuang F, Zhang Z, Qian M, et al. in [14].

h) DVMF: Deep variational matrix factorization is the model proposed in this article. To obtain objective results, we only use rating data to train the model, as the above mentioned models.

i) DVMF+: The model proposed in this article. DVMF+ adopts side information and extra implicit feedback to enhance the model performance.

3) Datasets: Five datasets are employed in our experiments as listed below: MovieLens-100K and MovieLens-1M datasets were collected under the GroupLens Research Project at the University of Minnesota. Douban-Book, Douban-Movie and Douban-Music datasets are the subsets of Douban-50000 on the corresponding area. Douban-50000 dataset is collected and shared by Zhong [39] from Douban which is one of the Chinese online social network sites providing reviews and recommendations services in books, movies, and music domains. The statistics information of five datasets summarizes in Table 1.

Rating scale on all five datasets is [1, 5]. For gaining objective and unbiased results, we have employed the 80%-20% train-test settings and 5-fold cross-validation technique. The experiments have been implemented for five times with different random seeds and the average scores are reported.

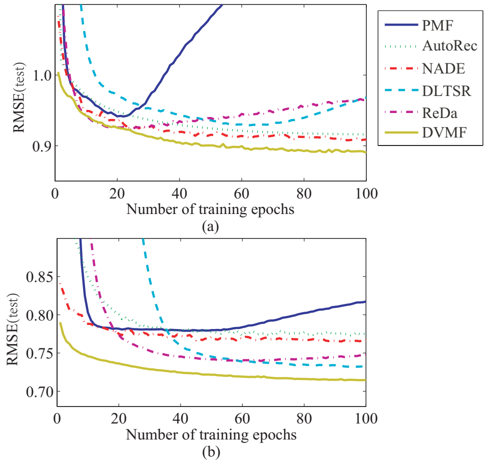
5.2 Experimental Process

The involved eight models are trained and compared on all five datasets. For PMF, we set $\lambda = 0.005$ in all datasets, and $k = 50$ in MovieLens-100K, $k = 200$ in the rest of the datasets. In BPMF, the k is set as 50 in MovieLens-100K, 200 in the other datasets, we also set μ_0 = 0, α = 2, and W_0 to the identity matrix, for both user and movie hyper priors. As to AutoRec, we choose $\lambda = 1$ in all datasets, and 200 hidden neurons for MovieLens-100K, 500 hidden neurons for the other datasets. We set the hidden size equal to 200 in MovieLens-100K and 500 for others in NADE. In DLTSR, we set $a = 100$, $\lambda_n = 1$, $\lambda_v = 10$, $\lambda_w = 0.0001$, and $c_H = 0.1$ in all datasets as the author recommended, the hidden structure is built as $[200, 50, 200]$ in MovieLens-100K, $[500, 200, 500]$ in the rest of the datasets. For ReDa, α

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2952849, IEEE Transactions on Knowledge and Data Engineering

TABLE 1
Statistics of MovieLens-100K, MovieLens-1M, Douban-Book, Douban-Movie and Douban-Music datasets

	User	Item	Rating	Sparsity	User side information	Item side information	Implicit feedback
MovieLens-100K	943	1 682	10 000	93.70%	Age, gender, occupation	Release data , genre	-
MovieLens-1M	3 900	6 040	1 000 209	95.75%	Age, gender, occupation	Release data , genre	-
Douban-Book	9 671	8 330	543 432	99.32%	-	-	422 783
Douban-Movie	13 363	13 530	2 530 679	98.60%	-	-	1 116 269
Douban-Music	8 334	11 073	809 000	99.12%	-	-	333 673



(a)

(b)

Fig. 4. Training processes of all models compared with RMSE measuring the generalized error on (a) MovieLens-100K and (b) Douban-Book.

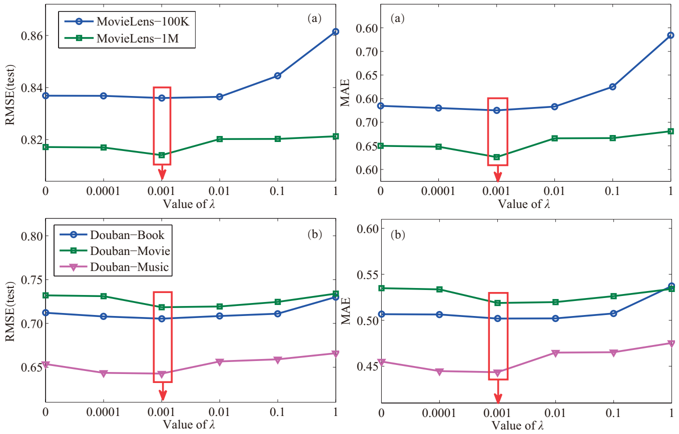


Fig. 5. Performance in test sets of DVMF+ measured by RMSE and MAE with λ increasing from 0 to 1 on (a) MovieLens-100K, MovieLens-1M, (b) Douban-Book, Douban-Movie, Douban-Music

and β are set at 0.5, γ is set as 1, $k = 50$ in all datasets. In DVMF, φ is set at 0.5 and λ is equal to 0.001 in all datasets, the hidden structure is $[50+, 200, 200, 200, 100]$ in MovieLens-100K dataset, $[300+, 600, 600, 600, 300]$ in MovieLens-1M, and $[200, 400, 400, 400, 200]$ in three Douban datasets. The first value in the hidden structure represents the embedding dimensionality in IFE, namely k. DVMF+ has the same parameters setting as DVMF. All eight models are optimized by mini-batch gradient descent algorithm with learning rate $\alpha = 0.02$ and batch size 128.

The experiment was performed on a PC Server equipped with an Intel(R) Core(TM) i7-7700K CPU@4.20GHz, NVIDIA GeForce GTX 1080 Ti GPU, and 32 GB RAM. We implemented the DVMF with software library TensorFlow [40].

5.3 Results and Discussion

1) Accuracy analysis of DVMF: For accuracy comparison of Mean, PMF, BPMF, AutoRec, NADE, DLTSR, ReDa, and DVMF on all five datasets, the lowest RMSE and MAE of each model are summarized in Table 2. The RMSE at training process is depicted in Fig. 4. In a general manner, deep learning based methods have a better performance than the traditional methods, as manifested in Table 2 and Fig. 4. AutoRec, NADE, DLTSR, ReDa, and DVMF have prodigious improvement relative to the baseline method PMF, BPMF and Mean. Compared with PMF, DVMF has 6.0%-10.8% improvement in RMSE and 12.4%-21.9% in MAE. Moreover DVMF achieves the best performance in both MAE and RMSE metrics even compared with the state-of-the-art recommendation algorithms. During the training process, DVMF is proved on its high efficiency which is revealed in Fig. 4. Furthermore, DVMF+ has better result than DVMF as it utilizes side information and extra implicit feedback. Results are improved 0.3%-0.4% by using side information in MovieLens-100K and MovieLens-1M datasets, 0.5%-1.4% by using extra implicit feedback in three Douban datasets, respectively. It proves that the DVMF+ is a high-performance recommendation framework and it has the capability to enhance model effectiveness by extracting the features from additional input information as well. And we believe the model performance can be further improved by merging meaningful information of user and item.

2) Sensitivity analysis of the parameter λ: In VMFM, λ is the regularization parameter, which controls the effectiveness of the regularization terms. In order to investigate the validity of the regularization terms and the optimum value of λ, some correlative experiments have been set. We implemented the DVMF+ model on all five datasets with the parameter λ sampled from 0 to 1 and the results are displayed in Fig. 5. From Fig. 5, we can observe that the model achieves better performance when λ is equal to 0.001 than 0. It demonstrates that the *Assumption 3* namely the prior of latent factors is logical and resultful. Model effectiveness could be improved by introduce the prior of latent factors with appropriate regularization parameter. Furthermore, improper regularization parameter could be unfavorable to the model which can be observed in Fig. 5. Thus there is a need to choose the most applicable value of regularization parameter λ. The DVMF+ model attains the highest forecast accuracy when λ equals 0.001 in all five

TABLE 2
Performance comparison in terms of RMSE, MAE on Movielens-100K, Movielens-1M, Douban-Book, Douban-Movie and Douban-Music

	MovieLens-100K		MovieLens-1M		Douban-Book		Douban-Movie		Douban-Music	
Methods	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Mean	1.1537	0.9680	1.1169	0.9335	0.8477	0.6516	0.9368	0.7570	0.7874	0.6455
PMF [19]	0.9701	0.7823	0.8891	0.6971	0.7791	0.6131	0.7909	0.6233	0.7192	0.5663
BPMF [23]	0.9480	0.7168	0.8578	0.6403	0.7439	0.6016	0.7438	0.5620	0.6884	0.5800
AutoRec [13]	0.9019	0.6771	0.8401	0.6214	0.7832	0.5788	0.8036	0.5900	0.7466	0.5616
NADE [15]	0.8984	0.6578	0.8457	0.6122	0.7656	0.5603	0.7458	0.5381	0.6776	0.4790
DLTSR [22]	0.9304	0.7375	0.8637	0.6709	0.7304	0.5267	0.7327	0.5160	0.6609	0.4605
ReDa [14]	0.9190	0.7203	0.8485	0.6646	0.7390	0.5386	0.7362	0.5277	0.6699	0.4712
DVMF	0.8892	0.6557	0.8347	0.6111	0.7059	0.5023	0.7094	0.5092	0.6408	0.4412
DVMF+	**0.8867**	**0.6534**	**0.8317**	**0.6081**	**0.7021**	**0.4992**	**0.7017**	**0.5010**	**0.6381**	**0.4405**

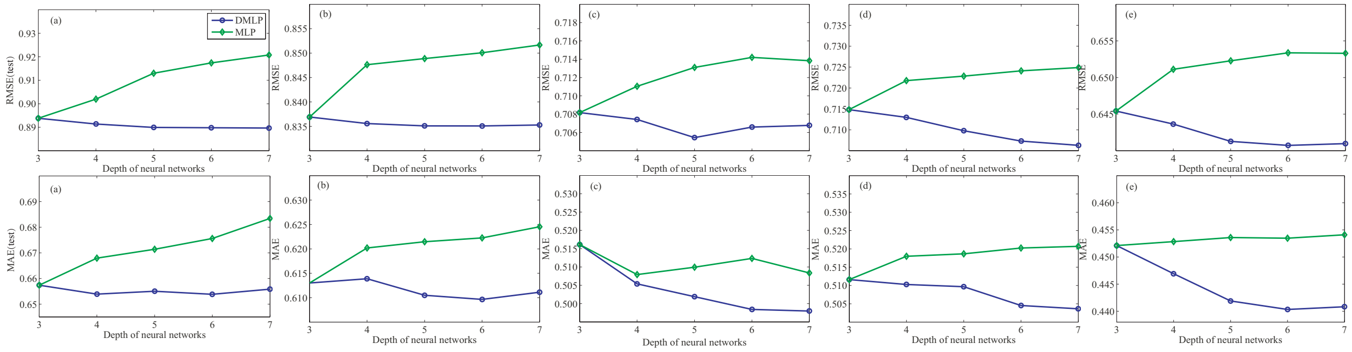


Fig. 6. Performance in test sets of DVMF+ measured by RMSE and MAE with different MLP architecture and depth on (a) MovieLens-100K, (b) MovieLens-1M, (c) Douban-Book, (d) Douban-Movie, (e) Douban-Music.

datasets. Hence, we believe that the value of λ can be set as 0.001 as a general rule.

3) Effect analysis of the D-MLP architecture: For exploring the availability of the proposed MLP architecture, we implemented DVMF+ model on all datasets with different MLP architectures and depth. Related experimental results are described in Fig. 6. It is interesting to observe that as the depth grows, the performance of the MLP based model are getting worse in all five datasets. Relatively the predictive accuracy of the D-MLP based model steadily improves as the depth grows. We believe the reason behind this phenomenon is that MLP based model is suffering from the vanishing gradient problem [28]. Deep MLP structure increases the difficulty of gradient propagating. However, the results in Fig. 6 indicate that the D-MLP architecture alleviates this problem with effect. Fig. 6 shows that these models not only significantly outperform their MLP counterparts, but also outperform DVMF+ models that have fewer hidden layers. Theoretically, in D-MLP architecture, each node is connected with the output layer directly, it means the gradient of each node has no decay in the back propagation process. Meanwhile, the model obtains high level neural features. So, comparing with traditional MLP, D-MLP is easier to train without losing any features. In practice, however, we will be limited by the available computer resources to choose befitting depth of D-MLP.

4) Stability analysis of the reparameterization sampler: we collected samples over the user and movie latent factors generated by the reparameterization sampler in MovieLens-100K datasets during the training course. The first 100 training epochs were discarded as pre-training phase. Fig. 7 shows these samples for two users and two movies projected onto the two dimensions of the highest variance. Users A and B were chosen by randomly picking among rare users who have fewer than 10 ratings and more frequent users who have more than 100 ratings in the dataset respectively. Movies X and Y were chosen in the same way. Compare with the numerous-rating user, the user A has almost the same variance with it, although user A only has 4 ratings in training data, and the movie X has similar qualities. We

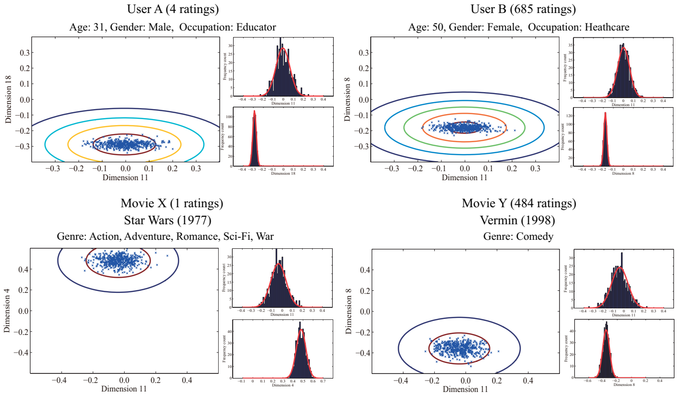


Fig. 7. Samples from the PIM generated by the reparameterization sampler. The two dimensions with the highest variance are shown for two users and two movies.

TABLE 3
Performance comparison in terms of nDCG on GPCR, Ion Channels(IC), Nuclear Receptors(NR) and Enzymes(E)

nDCG	GPCR	IC	NR	E
BLM-NII	0.887	0.928	0.919	0.872
WNN-GIP	0.890	0.895	0.920	0.860
NetLapRLS	0.917	0.930	0.941	0.871
CMF	0.910	0.954	0.910	0.873
BRDTI	0.929	0.953	0.948	0.897
DVMF	**0.956**	**0.980**	**0.989**	**0.952**

may conclude that DVMF is enabled to predict the latent factors steadily, even though the user or item has insufficient training data.

5.4 Extended application

In general, matrix factorization technique can be used in many tasks, like image restoration [41]–[43], drug-target interaction (DTI) prediction [44], [45], etc. In order to explore the extended applications of the proposed method DVMF, the experiments on DTI prediction are implemented. For adapting the DTI task, the training process and the objective function are modified as the same as the Bayesian ranking approach [45], moreover the experiment datasets and protocol are the same as [45]. We evaluate ranked lists for each drug separately and use normalized discounted cumulative gain (nDCG) as evaluation metric, which was shown to be the best graded relevance ranking metric with respect to the stability and sensitivity. The experiment results are presented in Table 3. Which can be observed from Table 3 that DVMF outperforms the advanced DTI algorithms. It proves that DVMF can be used as a powerful general matrix factorization method. Although the application considered here is focused on rating prediction task in recommendation, the method is more generally applicable to ranking or other matrix factorization tasks.

6 Conclusion

In this article, we have proposed a deep learning-based fully Bayesian treatment recommendation framework, DVMF, which has high-quality performance and ability to integrate any kinds of side information handily and efficiently. The main idea of DVMF is to build a fully Bayesian treatment recommendation model, named variational matrix factorization model (VMFM), which is supported by parametric inference model (PIM). The PIM structures two deep neural networks to produce the hyperparameters of the distributions of latent factor in VMFM with the knowledge embedding model (KEM) as the source information. Moreover the KEM is able to convert the high-dimensional and sparse knowledge of user and item into a low-dimensional real-valued vector retaining primary features. More precisely, in VMFM, the variational inference technique and the reparameterization tricks are introduced to make DVMF possible to be optimized by the stochastic gradient-based methods which is much more applicative in the big data scenario. Meanwhile, a new deep neural network architecture, D-MLP, has been constructed in PIM to combat the predicament caused by traditional MLP model, such as the vanishing gradient problem, and the KEM is capable of lessening the parameter scale and raising the training efficiency. Experimental results on five manifold real-world datasets well demonstrate that DVMF can obtain advantage in prediction accuracy even compared to the state-of-the-art models. Furthermore, DVMF is proficient to enhance recommendation effect by merging side information. Whereas the tuning process of deep learning based model is time-consuming and tedious. Thus, automatic adjustment of hyper-parameters is an important issue in our future research. Moreover it is necessary to explore new model assumptions and the latest deep learning models, such as attention model, in our future work.

Acknowledgments

This work was financially supported by the National Key R&D Program of China (2017YFB1401300, 2017YFB1401303). The authors would like to thank colleagues and the anonymous reviewers who have provided valuable feedback to help improve the paper.

References

[1] B. Yi, X. Shen, H. Liu, Z. Zhang, W. Zhang, S. Liu, and N. Xiong, "Deep matrix factorization with implicit feedback embedding for recommendation system," *IEEE Transactions on Industrial Informatics*, vol. 15, pp. 4591 – 4601, 2019.

[2] D. Lian, Y. Ge, F. Zhang, N. J. Yuan, X. Xie, T. Zhou, and Y. Rui, "Scalable content-aware collaborative filtering for location recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 6, pp. 1122–1135, 2018.

[3] S. Yan, K.-J. Lin, X. Zheng, W. Zhang, and X. Feng, "An approach for building efficient and accurate social recommender systems using individual relationship networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 10, pp. 2086–2099, 2017.

[4] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *International Conference on World Wide Web*, 2001, pp. 285–295.

[5] G. Guo, J. Zhang, and N. Yorke-Smith, "A novel recommendation model regularized with user trust and item ratings," *IEEE Transactions on Knowledge & Data Engineering*, vol. 28, no. 7, pp. 1607–1620, 2016.

[6] D. Kim, C. Park, J. Oh, and H. Yu, "Deep hybrid recommender systems via exploiting document context and statistics of items," *Information Sciences*, vol. 417, pp. 72–87, 2017.

[7] C. Chen, M. Zhang, Y. Liu, and S. Ma, "Neural attentional rating regression with review-level explanations," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2018, pp. 1583–1592.

[8] J. Shu, X. Shen, H. Liu, B. Yi, and Z. Zhang, "A content-based recommendation algorithm for learning resources," *Multimedia Systems*, vol. 24, no. 2, pp. 163–173, 2018.

[9] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W. Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 353–362.

[10] H. Wang and D. Y. Yeung, "Towards bayesian deep learning: A framework and some existing methods," *IEEE Transactions on Knowledge & Data Engineering*, vol. 28, no. 12, pp. 3395–3408, 2016.

[11] K. Georgiev and P. Nakov, "A non-iid framework for collaborative filtering with restricted boltzmann machines," in *International Conference on International Conference on Machine Learning*, 2013, pp. 1148–1156.

[12] N. Hazrati, B. Shams, and S. Haratizadeh, "Entity representation for pairwise collaborative ranking using restricted boltzmann machine," *Expert Systems with Applications*, vol. 116, pp. 161–171, 2019.

[13] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "Autorec:autoencoders meet collaborative filtering," in *International Conference on World Wide Web*, 2015, pp. 111–112.

[14] F. Zhuang, Z. Zhang, M. Qian, C. Shi, X. Xie, and Q. He, "Representation learning via dual-autoencoder for recommendation," *Neural Networks*, vol. 90, pp. 83–89, 2017.

[15] B. Uria, Marc-Alexandre, K. Gregor, I. Murray, and H. Larochelle, "Neural autoregressive distribution estimation," *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 7184–7220, 2016.

[16] Y. Zheng, B. Tang, W. Ding, and H. Zhou, "A neural autoregressive approach to collaborative filtering," in *International Conference on Machine Learning*, 2016, pp. 764–773.

[17] D. Jannach and M. Ludewig, "When recurrent neural networks meet the neighborhood for session-based recommendation," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, 2017, pp. 306–310.

[18] J. Liu, C. Wu, and J. Wang, "Gated recurrent units based neural network for time heterogeneous feedback recommendation," *Information Sciences*, vol. 423, pp. 50–65, 2018.

[19] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *International Conference on Neural Information Processing Systems*, 2007, pp. 1257–1264.

[20] Y. Koren, "Factor in the neighbors: Scalable and accurate collaborative filtering," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 4, no. 1, p. 1, 2010.

[21] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. Ieee, 2008, pp. 263–272.

[22] B. Bai, Y. Fan, W. Tan, and J. Zhang, "Dltsr: A deep learning framework for recommendation of long-tail web services," *IEEE Transactions on Services Computing*, vol. DOI:10.1109/TSC.2017.2681666, 2018.

[23] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using markov chain monte carlo," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 880–887.

[24] L. Jing, P. Wang, and L. Yang, "Sparse probabilistic matrix factorization by laplace distribution for collaborative filtering." in *IJCAI*, 2015, pp. 1771–1777.

[25] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *International Conference on Learning Representations*, 2015, p. 1C13.

[26] Hornik and Kurt, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991.

[27] M. Raghu, B. Poole, J. Kleinberg, S. Ganguli, and J. Sohl-Dickstein, "On the expressive power of deep neural networks," in *International Conference on Machine Learning*, 2017, pp. 2847–2854.

[28] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 06, no. 02, pp. 107–116, 1998.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[30] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[31] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[32] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *International Conference on International Conference on Machine Learning*, 2010, pp. 807–814.

[33] M. U. Gutmann and H. A., "Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics," *Journal of Machine Learning Research*, vol. 13, pp. 307–361, 2012.

[34] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[35] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, p. 533, 1986.

[36] H. W. Engl, "Discrepancy principles for tikhonov regularization of ill-posed problems leading to optimal convergence rates," *Journal of Optimization Theory & Applications*, vol. 52, no. 2, pp. 209–215, 1987.

[37] G. Golub, MichaelHeath, and GraceWahba, "Generalized cross-validation as a method for choosing a good ridge parameter," *Technometrics*, vol. 21, no. 2, pp. 215–223, 1979.

[38] P. C. Hansen, "Analysis of discrete ill-posed problems by means of the l-curve," *Siam Review*, vol. 34, no. 4, pp. 561–580, 1992.

[39] E. Zhong, Y. Li, Y. Li, Y. Li, and Y. Li, "Comsoc: adaptive transfer of user behaviors over composite social network," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012, pp. 696–704.

[40] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, and M. Isard, "Tensorflow: a system for large-scale machine learning," in *OSDI*, vol. 16, 2016, pp. 265–283.

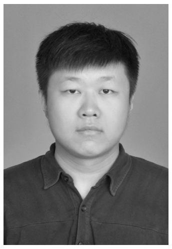
[41] H. Liu, Y. Li, Z. Zhang, S. Liu, and T. Liu, "Blind poissonian reconstruction algorithm via curvelet regularization for an ftir spectrometer," *Optics express*, vol. 26, no. 18, pp. 22 837–22 856, 2018.

[42] T. Liu, H. Liu, Y. Li, Z. Zhang, and S. Liu, "Efficient blind signal reconstruction with wavelet transforms regularization for educational robot infrared vision sensing," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 1, pp. 384–394, 2018.

[43] T. Liu, H. Liu, Z. Chen, and A. M. Lesgold, "Fast blind instrument function estimation method for industrial infrared spectrometers," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 12, pp. 5268–5277, 2018.

[44] Y. Wang and J. Zeng, "Predicting drug-target interactions using restricted boltzmann machines," *Bioinformatics*, vol. 29, no. 13, pp. i126–i134, 2013.

[45] L. Peska, K. Buza, and J. Koller, "Drug-target interaction prediction: A bayesian ranking approach," *Computer methods and programs in biomedicine*, vol. 152, pp. 15–21, 2017.



Xiaoxuan Shen is currently a PhD candidate in the National Engineering Research Center for E-Learning, Central China Normal University. His research interests include deep learning, representation learning, and their applications in recommendation system and intelligent e-learning environment.

Baolin Yi received the BE and MS degree in mathematics from the Wu Han University, China, in 1992 and 1997, respectively, and the PhD degree in computer science from the Huazhong University of Science and Technology in 2003. He is professor and PhD supervisor of National Engineering Research Center for E-Learning in Central China Normal University since 2010. He is awarded as the expert in the field of Education Information by the Department of Education of Hubei province. His research interest includes database and data mining, education information technology, education cloud computing and education big data analysis.

Hai Liu (S'12-M'14) received the M.S. degree in applied mathematics from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2010, and the Ph.D. degree in pattern recognition and artificial intelligence from the same university, in 2014.

Since June 2017, he has been an Assistant Professor with the National Engineering Research Center for E-Learning, Central China Normal University, Wuhan. Currently, he is a ”Hong Kong Scholar” postdoctoral fellow with the Department of Mechanical Engineering, City University of Hong Kong, Kowloon, Hong Kong, where he is hosted by the Porfessor Youfu Li; he will hold the position till March 2019. He has authored more than 30 peer-reviewed articles in international journals from multiple domains such as pattern recognition, image processing. His current research interests include big data processing, artificial intelligence, recommendation system, deep learning, signal processing and pattern recognition.

Dr. Liu has been frequently serving as a reviewer for more than six international journals including the *IEEE/ASME Transactions on Mechatronics*, *IEEE Translations on Industrial Informatics*, *IEEE Translations on Cybernetics*, *IEEE Translations on Instrumentation and Measurement*, *Digital Signal Processing*, *Measurement Science & Technology*, and *Applied Optics*. He is also a Communication Evaluation Expert for the National Natural Science Foundation of China.

Zhaoli Zhang (M'18) received the M.S. degree in Computer Science from Central China Normal University, Wuhan, China, in 2004, and the Ph.D. degree in Computer Science from Huazhong University of Science and Technology in 2008. He is currently a professor in the National Engineering Research Center for E-Learning, Central China Normal University. His research interests include signal processing, knowledge services and software engineering. He is a member of IEEE and CCF (China Computer Federation).

Wei Zhang is currently an associate professor in the National Engineering Research Center for E-Learning (http://nercel.ccnu.edu.cn/) and National Engineering Laboratory for Educational Big Data at the Central China Normal University. He holds a Ph.D. degree from Huazhong University of Science and Technology. His research interests include computer applications, big data analysis, data mining, and application of information technology in education. He published more than 40 papers in the academic journals, including 20 papers indexed by SSCI, SCI, EI, ISTP.

Sannyuya Liu received the B.E. and M.E. degrees in 1996 and 1999, and received the Ph.D. degree in 2003 from (HUST). He devoted himself to his postdoctoral research in Xiamen University from 2003 to 2005, and worked for the field of enterprise information, business intelligence, and distributed computing. Currently, he is a professor in NERCEL, CCNU. His research interests include artificial intelligence, computer application, and educational data mining.

Naixue Xiong received the Ph.D. degrees in sensor system engineering and in dependable sensor networks from Wuhan University and the Japan Advanced Institute of Science and Technology, respectively. Before he attended Tianjin University, he was with Northeastern State University, Georgia State University, the Wentworth Technology Institution, and Colorado Technical University (Full Professor for about five years) for about 10 years. He is currently a Professor with the College of Intelligence and Computing, Tianjin University, China. His research interests include cloud computing, security and dependability, parallel and distributed computing, networks, and optimization theory.