



Variational Deep Collaborative Matrix Factorization for Social Recommendation

Teng Xiao¹, Hui Tian², and Hong Shen^{1,3(✉)}

¹ School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China

² School of Information and Communication Technology, Griffith University,
Gold Coast, Australia

³ School of Computer Science, The University of Adelaide, Adelaide, Australia
hong@cs.adelaide.edu.au

Abstract. In this paper, we propose a Variational Deep Collaborative Matrix Factorization (VDCMF) algorithm for social recommendation that infers latent factors more effectively than existing methods by incorporating users' social trust information and items' content information into a unified generative framework. Unlike neural network-based algorithms, our model is not only effective in capturing the non-linearity among correlated variables but also powerful in predicting missing values under the robust collaborative inference. Specifically, we use variational auto-encoder to extract the latent representations of content and then incorporate them into traditional social trust factorization. We propose an efficient expectation-maximization inference algorithm to learn the model's parameters and approximate the posteriors of latent factors. Experiments on two sparse datasets show that our VDCMF significantly outperforms major state-of-the-art CF methods for recommendation accuracy on common metrics.

Keywords: Recommender System · Matrix Factorization · Deep Learning · Generative model

1 Introduction

Recommender System (RS) has been attracting great interests recently. The most commonly used technology for RS is Collaborative Filtering (CF). The goal of CF is to learn user preference from historical user-item interactions, which can be recorded by a user-item feedback matrix. Among CF-based methods, matrix factorization (MF) [17] is the most commonly used one. The purpose of MF is to find the latent factors for users and items by decomposing the user-item feedback matrix. However, the feedback matrix is usually sparse, which would result in the poor performance of MF. To track this problem, many hybrid methods such as those in [12, 21–24, 26], called *content MF* methods, incorporate auxiliary information, e.g., content of items, into MF. The content of items can be their tags and descriptions etc. These methods all first utilize some models (e.g., Latent

Dirichlet Allocation (LDA) [1], Stack Denoising AutoEncoders (SDAE) [20] or marginal Denoising AutoEncoders (mDAE) [3]) to extract items' content latent representations and then input them into probabilistic matrix factorization [17] framework. However, these methods demonstrate a number of major drawbacks: (a) They assume that users are independent and identically distributed, and neglect the social information of users, which can be used to improve recommendation performance [15,16]. (b) those methods [21,22] which are based on LDA only can handle text content information which is very limited in current multimedia scenario in real world. The learned latent representations by LDA are often not effective enough especially when the auxiliary information is very sparse [24] (c) For those methods [12,23,24,26] which utilize SDAE or aSDAE. The SDAE and mDAE are in fact not probabilistic models, which limits them to effectively combine probabilistic matrix factorization into a unified framework. They first corrupt the input content, and then use neural networks to reconstruct the original input. So those model also need manually choose various noise (masking noise, Gaussian noise, salt-and-peper noise, etc), which hinders them to expand to different datasets. Although some hybrid recommendation methods [2,10,18,25] that consider user social information have been proposed, they still suffer from problem (b) and (c) mentioned above. Recently, the deep generative model such Variational AutoEncoder (VAE) [11] has been utilized to the recommendation task and achieve promising performance due to it's full Bayesian nature and non-linearity power. Liang et al. proposed VAE-CF [14] which directly utilize VAE to the CF task. To incorporate item content information into VAE-CF, chen et al. proposed a collective VAE model [4] and Li et al. proposed Collaborative Variational Autoencoder [13]. However those methods all don't consider users' social information. To tackle the above problems, we propose a **Variational Deep Collaborative Matrix Factorization** algorithm for social recommendation, abbreviated as **VDCMF**, for social recommendation, which integrates item contents and user social information into a unified generative process, and jointly learns latent representations of users and items. Specifically, we first use VAE to extract items' latent representation and consider users' preferences effect by the personal tastes and their friends' tastes. We then combine these information into probabilistic matrix factorization framework. Unlike SDAE based methods, our model needs not to corrupt the input content, but instead to directly model the content's generative process. Due to the full Bayesian nature and non-linearity of deep neural networks, our model can learn more effective and better latent representations of users and items than LDA-based and SDAE-based methods and can capture the uncertainty of latent space [11]. In addition, with both item content and social information, VDCMF can effectively tackle the matrix sparsity problem. In our VDCMF, to infer latent factors of users and items, we propose a EM-algorithm to learn model parameters. To sum up, our main contributions are: (1) We propose a novel recommendation model called VDCMF for recommendation, which incorporates rich item content and user social information into MF. VDCMF can effectively learn latent factors of users and items in matrix sparsity cases. (2) Due to the full Bayesian

nature and non-linearity of deep neural networks, our VDCMF is able to capture more effective latent representations of users and items than state-of-the-art methods and can capture the uncertainty of latent content representation. (3) We derive an efficient parallel variational EM-style algorithm to infer latent factors of users and items. (4) Comprehensive experiments conducted on two large real-world datasets show VDCMF can significantly outperform state-of-the-art hybrid MF methods for CF.

2 Notations and Problem Definition

Let $\mathbf{R} \in \{0, 1\}^{N \times M}$ be a user-item matrix, where N and M are the number of users and items, respectively. $R_{ij} = 1$ denotes the implicit feedback from user i over item j is observed and $R_{ij} = 0$ otherwise. Let $\mathcal{G} = (\mathcal{U}, \mathcal{E})$ denote a trust network graph, where the vertex set \mathcal{U} represents users and \mathcal{E} represents the relations among them. Let $\mathbf{T} = \{T_{ik}\}^{N \times N}$ denote the trust matrix of a social network graph \mathcal{G} . We also use N_i to represent user i 's direct friends and U_{N_i} as their latent representations. Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M] \in \mathbb{R}^{L \times M}$ represent item content matrix, where L denotes the dimension of content vector \mathbf{x}_j , and x_j be the content information of item j . For example, if item j is a product or a music, the content x_j can be bag-of-words of its tags. We use $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_N] \in \mathbb{R}^{D \times N}$ and $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_M] \in \mathbb{R}^{D \times M}$ to denote user and item latent matrices, respectively, where D denotes the dimension. \mathbf{I}_D represents identity matrix with dimension D .

3 Variational Deep Collaborative Matrix Factorization

In this section, we propose a Variational Deep Collaborative Matrix Factorization for social recommendation, the goal of which is to infer user latent matrix \mathbf{U} and item latent matrix \mathbf{V} given item content matrix \mathbf{X} , user trust matrix \mathbf{T} and user-item rating matrix \mathbf{R} .

3.1 The Proposed Model

To incorporate users' social information and item content information in to probabilistic matrix, we consider a users' feedback or rating on items are a balance between item content, user's taste and friend's taste. For example, users's rating on movies is effected by the movie's content information (e.g., the genre and the actors) and his friend advices from their tastes. For items' content information, since it can be very complex and various, we do not know its real distribution. However, we know any distribution can be generated by mapping simple Gaussian through a sufficiently complicated function [7]. In our proposed model, we consider item contents to be generated by their latent content vectors through a generative network. The generative process of VDCMF is as follows:

1. For each user i , draw user latent vector $\mathbf{u}_i \sim \mathcal{N}(0, \lambda_u^{-1}) \prod_{f \in N_i} \mathcal{N}(\mathbf{u}_f, \lambda_f^{-1} T_{if}^{-1} \mathbf{I}_D)$.

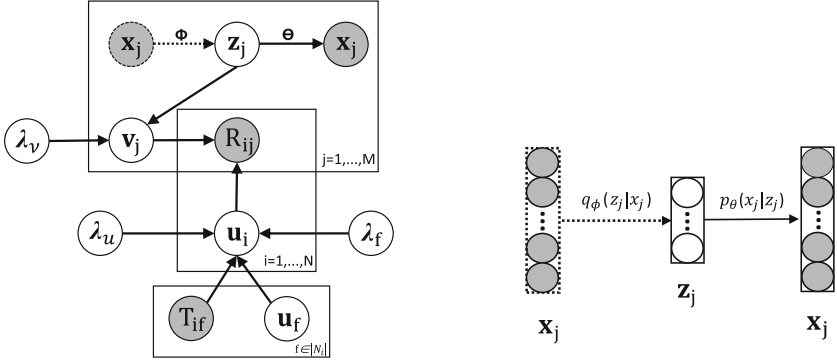


Fig. 1. Graphical model of VDCMF: generative network (left) and inference network (right), where solid and dashed lines represent generative and inference process, shaded nodes are observed variables.

2. For each item j :
 - (a) Draw item content latent vector $\mathbf{z}_j \sim \mathcal{N}(0, \mathbf{I}_D)$.
 - (b) Draw item content vector $p_\theta(\mathbf{x}_j|\mathbf{z}_j)$.
 - (c) Draw item latent offset $\mathbf{k}_j \sim \mathcal{N}(0, \lambda_v \mathbf{I}_D)$ and set the item latent vector as $\mathbf{v}_j = \mathbf{z}_j + \mathbf{k}_j$.
3. For each user-item pair (i, j) in \mathbf{R} , draw R_{ij} :

$$R_{ij} \sim \mathcal{N}(\mathbf{u}_i^\top \mathbf{v}_j, c_{ij}^{-1}). \quad (1)$$

In the process, λ_v , λ_u and λ_g are the free parameters, respectively. Similar to [21, 24], c_{ij} in Eq. 1 serves as confident parameters for R_{ij} and S_{ik} , respectively:

$$c_{ij} = \begin{cases} \varphi_1 & \text{if } R_{ij} = 1, \\ \varphi_2 & \text{if } R_{ij} = 0, \end{cases} \quad (2)$$

where $\varphi_1 > \varphi_2 > 0$ is the free parameters. In our model, we follow [18, 24] to set $\varphi_1 = 1$ and $\varphi_2 = 0.1$. $p_\theta(\mathbf{x}_j|\mathbf{z}_j)$ represents item content information and \mathbf{x}_j is generated from latent content vector \mathbf{z}_i through a generative neural network parameterized by θ . It should be noted that the specific form of the probability $p_\theta(\mathbf{x}_j|\mathbf{z}_j)$ depends on the type of the item content vector. For instance, if \mathbf{x}_j is binary vector, $p_\theta(\mathbf{x}_j|\mathbf{z}_j)$ can be a multivariate Bernoulli distribution $\text{Ber}(F_\theta(\mathbf{z}_j))$ with $F_\theta(\mathbf{z}_j)$ being the highly non-linear function parameterized by θ .

According to the graphic model in Fig. 1, the joint probability of \mathbf{R} , \mathbf{X} , \mathbf{U} , \mathbf{V} , \mathbf{Z} and \mathbf{T} can be represented as:

$$p(\mathcal{O}, \mathcal{Z}) = \prod_{i=1}^N \prod_{j=1}^M \prod_{k=1}^N p(\mathcal{O}_{ijk}, \mathcal{Z}_{ijk}) = \prod_{i=1}^N \prod_{j=1}^M \prod_{k=1}^N p(\mathbf{z}_j) p(\mathbf{u}_i | \mathbf{U}_{N_i}, \mathbf{T}) p_\theta(\mathbf{x}_j | \mathbf{z}_j) p(\mathbf{v}_j | \mathbf{z}_j) p(R_{ij} | \mathbf{u}_i, \mathbf{v}_j), \quad (3)$$

where $\mathcal{O} = \{\mathbf{R}, \mathbf{S}, \mathbf{X}\}$ is the set of all observed variables, $\mathcal{Z} = \{\mathbf{U}, \mathbf{V}, \mathbf{Z}\}$ is the set of all latent variables needed to be inferred, and $\mathcal{O}_{ijk} = \{R_{ij}, T_{ik}, \mathbf{x}_j\}$ and $\mathcal{Z}_{ijk} = \{\mathbf{u}_i, \mathbf{v}_j, \mathbf{z}_j\}$ for short.

3.2 Inference

Previous work [21,24] has shown that using an expectation-maximization (EM) algorithm enables recommendation methods that integrate them to obtain high-quality latent vectors (in our case, \mathbf{U} and \mathbf{V}). Inspired by these work, in this section, we derive an EM algorithm called VDCMF from the view of Bayesian point estimation. The marginal log likelihood can be given by:

$$\begin{aligned} \log p(\mathcal{O}) &= \log \int p(\mathcal{O}, \mathcal{Z}) d\mathcal{Z} \geq \int q(\mathcal{Z}) \log \frac{p(\mathcal{O}, \mathcal{Z})}{q(\mathcal{Z})} d\mathcal{Z} \\ &= \int q(\mathcal{Z}) \log p(\mathcal{O}, \mathcal{Z}) - \int q(\mathcal{Z}) \log q(\mathcal{Z}) \equiv \mathcal{L}(q), \end{aligned} \tag{4}$$

where we apply Jensen’s inequality, and $q(\mathcal{Z})$ and $\mathcal{F}(q)$ are variational distribution and the evidence lower bound (ELBO), respectively. For variational distribution $q(\mathcal{Z})$, we consider variational distributions in it to be matrix-wise independent:

$$\begin{aligned} q(\mathcal{Z}) &= q(\mathbf{U})q(\mathbf{V})q(\mathbf{Z}) \\ &= \prod_{i=1}^N q(\mathbf{u}_i) \prod_{j=1}^M q(\mathbf{v}_j) \prod_{j=1}^M q(\mathbf{z}_j). \end{aligned} \tag{5}$$

For Bayesian point estimation, we assume the variational distribution of \mathbf{u}_i is:

$$q(\mathbf{u}_i) = \prod_{d=1}^D \delta(U_{id} - \hat{U}_{id}). \tag{6}$$

$$q(\mathbf{v}_j) = \prod_{d=1}^D \delta(V_{jd} - \hat{V}_{jd}). \tag{7}$$

where $\{\hat{U}_{id}\}_{d=1}^D$ are variational parameters and δ is a Dirac delta function. Variational distributions of \mathbf{v}_j and \mathbf{g}_k are defined similarly. When U_{id} are discrete, the entropy of \mathbf{u}_i is:

$$H(\mathbf{u}_i) = - \int q(\mathbf{u}_i) \log q(\mathbf{u}_i) = \sum_{d=1}^D \sum_{U_{id}} \delta(U_{id} - \hat{U}_{id}) \log \delta(U_{id} - \hat{U}_{id}) = 0. \tag{8}$$

Similarly, $H(\mathbf{v}_j)$ is 0 when the elements are discrete. Then the evidence lower bound $\mathcal{L}(q)$ (Eq. 4) can be written as:

$$\begin{aligned} \mathcal{L}_{\text{point}}(\hat{\mathbf{U}}, \hat{\mathbf{V}}, \boldsymbol{\theta}, \boldsymbol{\phi}) &= \langle \log p(\mathbf{U}|\mathbf{T}, \mathbf{U}_{N_i})p(\mathbf{V}|\mathbf{Z}) \\ & p(\mathbf{X}|\mathbf{Z})p(\mathbf{R}|\mathbf{U}, \mathbf{V}) \rangle_q - \text{KL}(q_\phi(\mathbf{Z}|\mathbf{X})||p(\mathbf{Z})), \end{aligned} \tag{9}$$

where $\langle \cdot \rangle$ is the statistical expectation with respect to the corresponding variational distribution. $\hat{\mathbf{U}} = \{\hat{U}_{id}\}$ and $\hat{\mathbf{V}} = \{\hat{V}_{jd}\}$ are variational parameters corresponding to the variational distribution $q(\mathbf{U})$ and $q(\mathbf{V})$, respectively.

For latent variables \mathbf{Z} , However, it is intractable to infer \mathbf{Z} by using traditional mean-field approximation since we do not have any conjugate probability distribution in our model which requires by traditional mean-field approaches. To track this problem, we use amortized inference [6, 8], it consider a shared structure for every variational distributions, instead. Consequently, similar to VAE [11], we also introduce a variational distribution $q_\phi(\mathbf{Z}|\mathbf{X})$ to approximate the true posterior distribution $p(\mathbf{Z}|\mathcal{O})$. $q_\phi(\mathbf{Z}|\mathbf{X})$ is implemented by a inference neural network parameterized by ϕ (see Fig. 1). Specifically, for \mathbf{z}_j we have:

$$q(\mathbf{z}_j) = q_\phi(\mathbf{z}_j|\mathbf{x}_j) = \mathcal{N}(\boldsymbol{\mu}_j, \text{diag}(\boldsymbol{\delta}_j^2)), \quad (10)$$

where the mean $\boldsymbol{\mu}_j$ and variance $\boldsymbol{\delta}_j$ are the outputs of the inference neural network.

Directly maximizing the ELBO (Eq. 9) involves solving parameters $\hat{\mathbf{U}}$, $\hat{\mathbf{V}}$, $\boldsymbol{\theta}$ and ϕ , which is intractable. Thus, we derive an iterative variational-EM (VEM) algorithm to maximize $\mathcal{L}_{\text{point}}(\hat{\mathbf{U}}, \hat{\mathbf{V}}, \boldsymbol{\theta}, \phi)$, abbreviated $\mathcal{L}_{\text{point}}$.

Variational E-step. We first keep $\boldsymbol{\theta}$ and ϕ fixed, then optimize evidence lower bound $\mathcal{L}_{\text{point}}$ with respect to $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$. We take the gradient of \mathcal{L} with respect to \mathbf{u}_i and \mathbf{v}_j and set it to zero. We will get the updating rules of $\hat{\mathbf{u}}_i$ and $\hat{\mathbf{v}}_j$:

$$\hat{\mathbf{u}}_i \leftarrow (\mathbf{V}\mathbf{C}_i\mathbf{V}^\top + \lambda_u\mathbf{I}_D + \lambda_f\mathbf{T}_i\mathbf{1}_I\mathbf{I}_D)^{-1}(\lambda_f\mathbf{U}\mathbf{T}_i^\top + \mathbf{V}\mathbf{C}_i\mathbf{R}_i), \quad (11)$$

$$\hat{\mathbf{v}}_j \leftarrow (\hat{\mathbf{U}}\mathbf{C}_j\hat{\mathbf{U}}^\top + \lambda_v\mathbf{I}_D)^{-1}(\hat{\mathbf{U}}\mathbf{C}_j\mathbf{R}_j + \lambda_v\langle\mathbf{z}_j\rangle), \quad (12)$$

where $\mathbf{C}_i = \text{diag}(c_{i1}, \dots, c_{iM})$, $\mathbf{T}_i = \text{diag}(T_{i1}, \dots, T_{iM})$, $\mathbf{R}_i = [R_{i1}, \dots, R_{iM}]$. \mathbf{I}_N is a N dimensional column vector with all elements elements to 1. For item latent vector \mathbf{v}_j , \mathbf{C}_j and \mathbf{R}_j are defined similarly. $\hat{\mathbf{u}}_i = [\hat{U}_{i1}, \dots, \hat{U}_{iD}]$ and $\hat{\mathbf{v}}_j = [\hat{V}_{j1}, \dots, \hat{V}_{jD}]$. For \mathbf{z}_j , its expectation is $\langle\mathbf{z}_j\rangle = \boldsymbol{\mu}_j$, which is the output of the inference network.

It can be observed that λ_v governs how much the latent item vector \mathbf{z}_j affects item latent vector \mathbf{v}_j . For example, if $\lambda_v = \infty$, it indicate we direct use latent item vector to represent item latent vector \mathbf{v}_j ; if $\lambda_v = 0$, it means we do not embed any item content information into item latent vector. λ_f serves as a balance parameter between social trust matrix and user-item matrix on user latent vector \mathbf{u}_i . For example, if $\lambda_f = \infty$, it means we only use the social network information to model user's preference; if $\lambda_f = 0$, we only use user-item matrix and item content information for prediction. So λ_v and λ_f are regarded as collaborative parameters for item content, user-item matrix and social matrix.

Variational M-step. Keep $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$ fixed, we optimize $\mathcal{L}_{\text{point}}$ w.r.t. ϕ and $\boldsymbol{\theta}$ (we only focus on terms containing ϕ and $\boldsymbol{\theta}$).

$$\begin{aligned} \mathcal{L}_{\text{point}} = \text{constant} + \sum_{j=1}^M \mathcal{L}(\boldsymbol{\theta}, \phi; \mathbf{x}_j, \mathbf{v}_j) = \text{constant} + \sum_{j=1}^M \quad (13) \\ - \frac{\lambda_v}{2} \langle (\mathbf{v}_j - \mathbf{z}_j)^\top (\mathbf{v}_j - \mathbf{z}_j) \rangle_{q(\mathbf{z})} + \langle \log p_\theta(\mathbf{x}_j|\mathbf{z}_j) \rangle_{q_\phi(\mathbf{z}_j|\mathbf{x}_j)} - \text{KL}(q_\phi(\mathbf{z}_j|\mathbf{x}_j)||p(\mathbf{z}_j)), \end{aligned}$$

where M is the number of items and the constant term represents terms which don't contain $\boldsymbol{\theta}$ and ϕ . For the expectation term $\langle p_\theta(\mathbf{x}_j|\mathbf{z}_j) \rangle_{q_\phi(\mathbf{z}_j|\mathbf{x}_j)}$, we can not

solve it analytically. To handle this problem, we approximate it by the Monte Carlo sampling as follows:

$$\langle \log p_{\theta}(\mathbf{x}_j | \mathbf{z}_j) \rangle_{q_{\phi}(\mathbf{z}_j | \mathbf{x}_j)} = \frac{1}{L} \sum_{l=1}^L p_{\theta}(\mathbf{x}_j | \mathbf{z}_j^l), \quad (14)$$

where L is the size of samplings, and \mathbf{z}_j^l denotes the l -th sample, which is reparameterized to $\mathbf{z}_j^l = \boldsymbol{\epsilon}_j^l \odot \text{diag}(\boldsymbol{\delta}_j^2) + \boldsymbol{\mu}_j$. Here $\boldsymbol{\epsilon}_j^l$ is drawn from $\mathcal{N}(0, \mathbf{I}_D)$ and \odot is an element-wise multiplication. By using this reparameterization trick and Eq. 10, $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}_j, \mathbf{v}_j)$ in Eq. 13 can be estimated by:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}_j, \mathbf{v}_j) &\simeq \tilde{\mathcal{L}}^j(\boldsymbol{\theta}, \boldsymbol{\phi}) = -\frac{\lambda_v}{2}(-2\boldsymbol{\mu}_j^{\top} \hat{\mathbf{v}}_j + \boldsymbol{\mu}_j^{\top} \boldsymbol{\mu}_j \\ &+ \text{tr}(\text{diag}(\boldsymbol{\delta}_j^2))) + \frac{1}{L} \sum_{l=1}^L p_{\theta}(\mathbf{x}_j | \mathbf{z}_j^l) - \text{KL}(q_{\phi}(\mathbf{z}_j | \mathbf{x}_j) || p(\mathbf{z}_j)) + \text{constant}. \end{aligned} \quad (15)$$

We can construct an estimator of $\mathcal{L}_{\text{point}}(\boldsymbol{\phi}, \boldsymbol{\theta}; \mathbf{X}, \mathbf{V})$, based on minibatches:

$$\mathcal{L}_{\text{point}}(\boldsymbol{\theta}, \boldsymbol{\phi}) \simeq \tilde{\mathcal{L}}^P(\boldsymbol{\theta}, \boldsymbol{\phi}) = \frac{M}{P} \sum_{j=1}^P \tilde{\mathcal{L}}^j(\boldsymbol{\theta}, \boldsymbol{\phi}). \quad (16)$$

As discussed in [11], the number of samplings L per item j can be set to 1 as long as the minibatch size P is large enough, e.g., $P = 128$. We can update $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ by using the gradient $\nabla_{\boldsymbol{\theta}, \boldsymbol{\phi}} \tilde{\mathcal{L}}^P(\boldsymbol{\theta}, \boldsymbol{\phi})$.

We iteratively update $\mathbf{U}, \mathbf{V}, \mathbf{G}, \boldsymbol{\theta}$, and $\boldsymbol{\phi}$ until it converges.

3.3 Prediction

After we get the approximate posteriors of \mathbf{u}_i and \mathbf{v}_j . We predict the missing value R_{ij} in \mathbf{R} by using the learned latent features \mathbf{u}_i and \mathbf{v}_j :

$$R_{ij}^* = \langle R_{ij} \rangle = (\langle \mathbf{z}_j \rangle + \langle \mathbf{k}_j \rangle)^{\top} \langle \mathbf{u}_i \rangle = \langle \mathbf{v}_j \rangle^{\top} \langle \mathbf{u}_i \rangle \quad (17)$$

For a new item that is not rated by any other users, the offset $\boldsymbol{\epsilon}_j$ is zero, and we can predict R_{ij} by:

$$R_{ij}^* = \langle R_{ij} \rangle = \langle \mathbf{z}_j \rangle^{\top} \langle \mathbf{u}_i \rangle \quad (18)$$

4 Experiments

4.1 Experimental Setup

Datasets. In order to evaluate performance of our model, we conduct experiments on two real-world datasets from Lastfm¹ (*lastfm-2k*) and Epinions² (*Epinions*) datasets:

¹ <http://www.lastfm.com>.

² <http://www.Epinions.com>.

Lastfm. This dataset contains user-item, user-user, and user-tag-item relations. We first transform this dataset as implicit feedback. For Lastfm dataset, we consider the user-item feedback is 1 if the user has listened to the artist (item); otherwise, it is 0. Lastfm only contains 0.27% observed feedbacks. We use items bag-of-word tag representations as their items content information. We directly use user social matrix as trust matrix.

Epinions. This dataset contains rating, user trust and review information. We transform this dataset as implicit feedback. For those >3 ratings, we transform it as ‘1’; otherwise, it is 0. We use item’s review as its content information. Epinions contains 0.08% observed feedbacks.

Baselines. For fair comparisons, like that in our VDCMF, the baselines we used also incorporate user social information or item content information into matrix factorization. **(1) PMF.** This model [17] is a famous MF method, and only uses user-item feedback matrix. **(2) SoRec.** This model [15] jointly decomposes user-user social matrix and user-item feedback matrix to learn user and item latent representations. **(3) Collaborative topic regression (CTR).** This model [21] utilizes topic model and matrix factorization to learn latent representations of users and items. **(4) Collaborative deep learning (CDL).** This model [24] utilizes stack denoising autoencoder to learn latent items’ content representations, and incorporates them into probabilistic matrix factorization. **(5) CTR-SMF.** This model [18] incorporates topic modeling and probabilistic MF of social networks. **(6) PoissonMF-CS.** This model [19], jointly models user social trust, item content and users preference using Poisson matrix factorization framework. It is a state-of-the-art MF method for Top-N recommendation on the Lastfm dataset. **(7) Neural Matrix Factorization (NeuMF).** This model [9] is a state-of-the-art collaborative filtering method, which utilizes neural network to model the interaction between user and item.

Settings. For fair comparisons, We first set the parameters for PMF, SoRec, CTR, CTR-SMF, CDL, NeuMF via five-fold cross validation. For our model, we set $\lambda_u = 0.1$, $D = 25$ for Lastfm and $D = 50$ for Epinions. Without special mention, we set $\lambda_v = 0.1$ and $\lambda_f = 0.1$. We will further study the impact of the key hyper-parameters for the recommendation performance.

Evaluation Metrics. The metrics we used are Recall@K, NDCG@K and MAP@K [5] which are common metrics for recommendation.

4.2 Experimental Results and Discussions

Overall Performance. To evaluate our model in top- K recommendation task, we evaluate our model and baselines in two datasets in terms of Recall@20, Recall@50, NDCG@20 and MAP@20. Table 1 shows the performance of our VDCMF and the baselines using the two datasets. According to Table 1, we have following findings: (a) VDCMF outperforms the baselines in terms of all

Table 1. Recommendation performance of VDCMF and baselines. The best baseline method is highlighted with underline.

	Lastfm dataset				Epinions dataset			
	Recall@20	Recall@50	NDCG@20	MAP@20	Recall@20	Recall@50	NDCG@20	MAP@20
PMF	0.0923	0.1328	0.0703	0.1083	0.4012	0.5121	0.3019	0.3365
SoRec	0.1088	0.1524	0.0721	0.1128	0.4341	0.5547	0.3254	0.3621
CTR	0.1192	0.1624	0.0799	0.1334	0.5024	0.6125	0.3786	0.4197
CTR-SMF	0.1232	0.1832	0.0823	0.1386	0.5213	0.6217	0.3942	0.4437
CDL	0.1346	0.2287	0.0928	0.1553	0.5978	0.6597	0.4502	0.4792
NeuMF	<u>0.1517</u>	0.2584	0.1036	<u>0.1678</u>	0.6043	<u>0.6732</u>	0.4611	<u>0.4987</u>
PoissonMF-CS	0.1482	<u>0.2730</u>	<u>0.1089</u>	0.1621	0.5876	0.6533	<u>0.4628</u>	0.4876
VDCMF (ours)	0.1613	0.3006	0.1114	0.1695	0.6212	0.6875	0.4782	0.5123

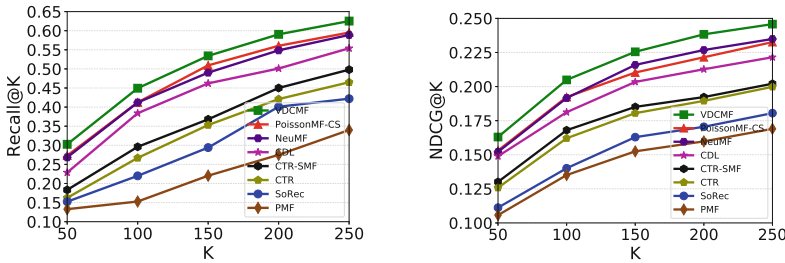


Fig. 2. Evaluation of Top-K item recommendation where K ranges from 50 to 250 on Lastfm

matrices on Lastfm and Epinions, which demonstrates the effectiveness of our method of inferring the latent factors of users and items, and leading to better recommendation performance. (b) For more sparse dataset, Epinions, VDCMF also achieves the best performance, which demonstrates our model can effectively handle matrix sparsity problem. We attribute this improvement to the incorporated item content and social trust information. (c) We can see methods which both utilizes content and social information (VDCMF, NeuMF and PoissonMF-CS) outperform others (CDL,CTR, CTR-SMF, SoRec and PMF), which demonstrates incorporating content and social information can effectively alleviate matrix sparse problem. (d) Our VDCMF outperforms the strong baseline PoissonMF-CS, though they are both Bayesian generative model. The reason VDCMF is that our VDCMF incorporates neural network into Bayesian generative model, which makes it have powerful non-linearity to model item content’s latent representation. To further evaluate our VDCMF robustness, we evaluate the empirical performance of large recommendation list for our VDCMF on Lastfm and report results in Fig. 2. We can find our VDCMF significantly and consistently outperforms other baselines. This, again, demonstrates the effectiveness of our model. All of these findings demonstrates that our VDCMF is robust and it is able to achieve significant improvements of top-k recommendation over the state-of-the-art.

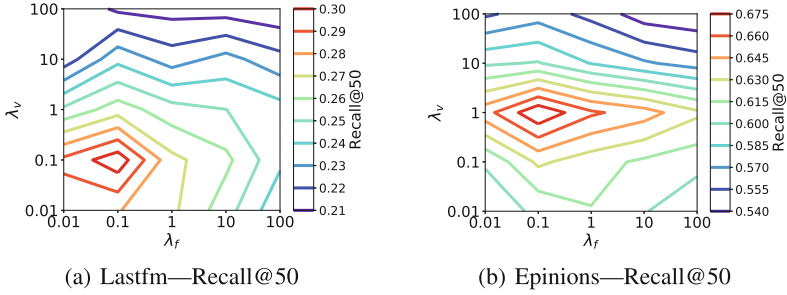


Fig. 3. The effect of λ_v and λ_f of the proposed VDCMF with Recall@50 on Lastfm and Epinions.

Impact of Parameters. In this section, we study the effect of the key hyperparameters of the proposed model. We first study the parameters of λ_f and λ_v . We use Recall@50 as an example, the plot the contours on Lastfm and Epinions datasets. Figure 3(a) and (b) show the contour of Recall@50. As we can see, VDCMF achieves the best recommendation performance when $\lambda_v = 0.1$ and $\lambda_f = 0.1$ on Lastfm, and $\lambda_v = 1$ and $\lambda_f = 0.1$ on Epinions. From Fig. 3(a) and (b), we can find our model is sensitive to λ_v and λ_f . The reason is that λ_v can control how much item content information is incorporated into item latent vector, λ_q can control how much social information is incorporated into user latent vector. Figure 3(a) and (b) show that we can balance the content information and social information by varying λ_v and λ_q , leading to better recommendation performance.

5 Conclusion

In this paper, we studied the problem of inferring effective latent factors of users and items for social recommendation. We have proposed a novel Variational Deep Collaborative Matrix Factorization algorithm, VDCMF, which incorporates rich item content and user social trust information into a full Bayesian deep generative framework. Due to the full Bayesian nature and non-linearity of deep neural networks, our proposed model is able to learn more effective latent representations of users and items than those generated by state-of-the-art neural networks based recommendation algorithms. To effectively infer latent factors of users and items, we derived an efficient expectation-maximization algorithm. We have conducted experiments on two publicly available datasets. We evaluated the performance of our VDCMF and baselines methods based on Recall, NDCG and MAP metrics. Experimental results demonstrate that our VDCMF can effectively infer latent factors of users and items.

Acknowledgement. This work is supported by the National Key Research and Development Program of China (No. #2017YFB0203201) and Australian Research Council Discovery Project DP150104871.

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. *J. Mach. Learn. Res.* **3**(Jan), 993–1022 (2003)
2. Chen, C., Zheng, X., Wang, Y., Hong, F., Lin, Z., et al.: Context-aware collaborative topic regression with social matrix factorization for recommender systems. In: *AAAI*, pp. 9–15 (2014)
3. Chen, M., Weinberger, K., Sha, F., Bengio, Y.: Marginalized denoising autoencoders for nonlinear representations. In: *International Conference on Machine Learning*, pp. 1476–1484 (2014)
4. Chen, Y., de Rijke, M.: A collective variational autoencoder for top-n recommendation with side information. In: *Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems*, pp. 3–9. ACM (2018)
5. Croft, W.B., Metzler, D., Strohman, T.: *Search Engines: Information Retrieval in Practice*. Addison-Wesley, Reading (2015)
6. Dayan, P., Hinton, G.E., Neal, R.M., Zemel, R.S.: The Helmholtz machine. *Neural Comput.* **7**(5), 889–904 (1995)
7. Doersch, C.: Tutorial on variational autoencoders. CoRR abs/1606.05908 (2016)
8. Gershman, S., Goodman, N.: Amortized inference in probabilistic reasoning. In: *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 36 (2014)
9. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: *WWW*, pp. 173–182 (2017)
10. Hu, G.N., et al.: Collaborative filtering with topic and social latent factors incorporating implicit feedback. *ACM Trans. Knowl. Discov. Data (TKDD)* **12**(2), 23 (2018)
11. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: *ICLR* (2014)
12. Li, S., Kawale, J., Fu, Y.: Deep collaborative filtering via marginalized denoising auto-encoder. In: *CIKM*, pp. 811–820 (2015)
13. Li, X., She, J.: Collaborative variational autoencoder for recommender systems. In: *KDD*, pp. 305–314 (2017)
14. Liang, D., Krishnan, R.G., Hoffman, M.D., Jebara, T.: Variational autoencoders for collaborative filtering. In: *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pp. 689–698. International World Wide Web Conferences Steering Committee (2018)
15. Ma, H., Yang, H., Lyu, M.R., King, I.: Sorec: social recommendation using probabilistic matrix factorization. In: *CIKM*, pp. 931–940 (2008)
16. Ma, H., Zhou, D., Liu, C., Lyu, M.R., King, I.: Recommender systems with social regularization. In: *WSDM*, pp. 287–296 (2011)
17. Mnih, A., Salakhutdinov, R.R.: Probabilistic matrix factorization. In: *NIPS*, pp. 1257–1264 (2008)
18. Purushotham, S., Liu, Y., Kuo, C.C.J.: Collaborative topic regression with social matrix factorization for recommendation systems. In: *ICML*, pp. 691–698 (2012)
19. da Silva, E.S., Langseth, H., Ramampiaro, H.: Content-based social recommendation with poisson matrix factorization. In: Ceci, M., Hollmén, J., Todorovski, L., Vens, C., Džeroski, S. (eds.) *ECML PKDD 2017. LNCS (LNAI)*, vol. 10534, pp. 530–546. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-71249-9_32
20. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A.: Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.* **11**(Dec), 3371–3408 (2010)

21. Wang, C., Blei, D.M.: Collaborative topic modeling for recommending scientific articles. In: KDD, pp. 448–456 (2011)
22. Wang, H., Chen, B., Li, W.J.: Collaborative topic regression with social regularization for tag recommendation. In: IJCAI, pp. 2719–2725 (2013)
23. Wang, H., Shi, X., Yeung, D.Y.: Relational stacked denoising autoencoder for tag recommendation. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
24. Wang, H., Wang, N., Yeung, D.Y.: Collaborative deep learning for recommender systems. In: KDD, pp. 1235–1244 (2015)
25. Wu, H., Yue, K., Pei, Y., Li, B., Zhao, Y., Dong, F.: Collaborative topic regression with social trust ensemble for recommendation in social media systems. *Knowl.-Based Syst.* **97**, 111–122 (2016)
26. Zhang, F., Yuan, N.J., Lian, D., Xie, X., Ma, W.Y.: Collaborative knowledge base embedding for recommender systems. In: KDD, pp. 353–362 (2016)