Bayesian Deep Collaborative Matrix Factorization

Teng Xiao, Shangsong Liang*, Weizhou Shen, Zaiqiao Meng
School of Data and Computer Science, Sun Yat-sen University, China
{cstengxiao, liangshangsong}@gmail.com, shenwzh3@mail2.sysu.edu.cn, zqmeng@aliyun.com

Abstract
In this paper, we propose a Bayesian Deep Collaborative Matrix Factorization (BDCMF) algorithm for collaborative filtering (CF). BDCMF is a novel Bayesian deep generative model that learns user and item latent factors from users’ social interactions, contents of items as the auxiliary information and user-item rating (feedback) matrix. It alleviates the problem of matrix sparsity by incorporating items’ auxiliary information and users’ social information into the model. It can learn more robust and dense latent representations by integrating deep learning into Bayesian probabilistic framework. As being one of deep generative models, it has both non-linearity and Bayesian nature. Additionally, in BDCMF, we derive an efficient EM-style point estimation algorithm for parameter learning. To further improve recommendation performance, we also derive a full Bayesian posterior estimation algorithm for inference. Experiments conducted on two sparse datasets show that BDCMF can significantly outperform the state-of-the-art CF methods.

Introduction
Social media, e.g., Facebook and Twitter, has greatly influenced people’s daily lives, and thus building an effective Recommender System (RS) for them has attracted many interests in recent years. The most used technique in RS is collaborative filtering (CF), which makes use of users’ ratings over items. Matrix factorization (MF) (Mnih and Salakhutdinov 2008) is one of the most commonly used CF methods due to its effectiveness and scalability. The goal of MF is to learn latent factors of both users and items from user-item feedback matrix. However, traditional MF methods suffer from the matrix sparsity problem – recommendation performance drops dramatically when the user-item feedback matrix is very sparse. In addition, traditional MF methods assume users are independent and identically distributed, and ignore the social connections among users.

To alleviate the matrix sparsity problem, many social MF methods such as those in (Park, Kim, and Choi 2013; Adams, Dahl, and Murray 2014) incorporate auxiliary information, e.g., content information of items, into MF. However, these methods incorporate auxiliary information as a linear regularization term, resulting in the fact that learning latent representations of users and items is not effective when the auxiliary information is also sparse. Aiming at learning more effective representations, many CF-based models using different types of auxiliary information, such as latent Dirichlet allocation (LDA) (Wang and Blei 2011), stacked denoising autoencoder (SDAE) (Wang, Wang, and Yeung 2015; Zhang et al. 2016), variational autoencoder(Li and She 2017) and marginalized denoising autoencoder (Li, Kawale, and Fu 2015), have been proposed to obtain items’ latent content representations. Nevertheless, all these models ignore social connections among users.

In order to jointly utilize item contents and social network information, some hybrid MF methods have been proposed, which can be roughly classified into two categories: i.e., Collaborative Topic Regression (CTR)-based methods (Wang and Blei 2011; Chen et al. 2014; Purushotham, Liu, and Kuo 2012; Wang, Chen, and Li 2013; Xu et al. 2017) and Collaborative Deep Learning (CDL)-based methods (Wang, Wang, and Yeung 2015; Zhang et al. 2016; Nguyen and Lauw 2017). Although CTR-based and CDL-based methods have achieved great performance, the full Bayesian posterior estimation is intractable in these models. All these models need to maximize a point estimation for parameter learning. However, in this paper, we point out that maximizing such a point estimation actually assumes that the variation distributions are discrete from variational Bayesian perspective (will be discussed later), which leads to a higher variance in prediction and ignores uncertainty in the model parameters (Lim and Teh 2007). This Bayesian estimation also leads to overly optimistic estimates of test-set log-likelihood (Welling, Chemudugunta, and Sutter 2008) and easily overfit the observed data, as shown in the following experiments. These drawbacks limit CTR-based and CDL-based methods to learn better latent representations from sparse datasets.

To tackle the above problems, in this paper, we aim to build a full Bayesian deep framework to jointly learn latent factors of users and items from three source data, i.e., item content, user-item feedback and user social matrices. We propose a Bayesian Deep Collaborative Matrix Factorization model, abbreviated as BDCMF, which integrates item contents and user social information into probabilistic matrix factorization (PMF). Different from CTR-
Based and CDL-based methods which utilize LDA and SDAE to capture items’ latent content vectors, our BDCMF is a full deep generative model and considers the various item content information to be generated by items’ latent content factors through a generative network. In addition, with both item content and social information, BDCMF can effectively tackle the matrix sparsity problem. In our BDCMF, to infer latent factors of users and items, we first propose a novel variational EM algorithm from a Bayesian point estimation perspective. Due to the Bayesian nature of BDCMF and drawbacks of Bayesian point estimation, we also propose Bayesian posterior estimation to infer the posterior distribution of users’ and items’ latent factors. To sum up, our main contributions are:

- We propose a novel deep generative model called BDCMF, which jointly learns latent factors of users and items from content, feedback and social matrices.
- We derive an efficient parallel variational EM-style algorithm from Bayesian point estimation perspective to infer latent factors of users and items.
- To obtain high-quality latent factors, we derive a full Bayesian variational posterior estimation algorithm that can infer posterior distribution of latent factors.
- Comprehensive experiments conducted on two large real-world datasets show BDCMF can significantly outperform state-of-the-art hybrid MF methods for CF.

### Related Work

Matrix factorization (MF) is one of the most used approaches in collaborative filtering (CF) and other applications such as rank aggregation (Liang et al. 2014; 2018a). Since the traditional MF suffers from matrix sparsity problem, some previous works explore items’ content information to improve the performance. Ma et al. (2008) proposed SocRec, which incorporates social information into probabilistic MF. Wang and Blei (2011) proposed CTR, which explores LDA to capture text topics, and integrates it into probabilistic MF. Wang, Wang, and Yeung (2015) proposed CDL which incorporates SDAE (Stack Denoising Auto-Encoder) into MF framework. To consider both user social information and item content information, many CTR-based and CDL-based methods have been proposed. Purushotham, Liu, and Kuo (2012) proposed CTR-SMF, which incorporates CTR and social matrix factorization to improve performance. In C-CTR-SMF2 (Chen et al. 2014), social content information was integrated into CTR. de Souza da Silva, Langseth, and Ramampiaro (2017) proposed PoissonMFCS, which utilizes Poisson matrix factorization to model users’ preferences and items’ contents. Ren et al. proposed sCVR (Ren et al. 2017) to predict item ratings based on user opinions and social relations. Recently, some previous works utilize neural networks to model latent factors of users and items, due to their non-linear modelling abilities. These include NeuMF (Neural Matrix factorization) (He et al. 2017) and CDAE (Collaborative Denoising Auto-Encoder) (Wu et al. 2016). However, the stacked neural network structures of them make them difficult to train and incur high computational cost. In contract, our BDCMF incorporates neural network into Bayesian generative framework, which makes it have non-linear modelling abilities and to be trained effectively.

### Problem Definition and Notations

Let $R \in \{0, 1\}^{N \times M}$ 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. We use $R_k$ to represent the $j$-th column of $R$. Let $S = \{S_{ik}\}_{1 \leq i \leq N}$ denote the social matrix of a social network graph $G$, where $S_{ik} = 1$ if user $i$ is associated with user $k$ and $S_{ik} = 0$ otherwise. Let $X = \{x_1, x_2, \ldots, x_M\} \in \mathbb{R}^{L \times M}$ represent item content matrix, where $L$ denotes the dimension of content vector $x_j$, 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-word representation of its tags. We use $U = \{u_1, u_2, \ldots, u_N\} \in \mathbb{R}^{D \times N}$ and $V = \{v_1, v_2, \ldots, v_M\} \in \mathbb{R}^{D \times M}$ to denote user and item latent matrices, respectively, where $D$ denotes the latent dimension. $G = \{g_1, g_2, \ldots, g_N\} \in \mathbb{R}^{D \times N}$ denotes the social factor latent matrix. $I_D$ represents identity matrix with dimension $D$. The problem we aim to address is: given a user-item matrix $R$, item content matrix $X$ and social matrix $S$, infer user latent matrix $U$, social factor latent matrix $G$ and item latent matrix $V$ such that missing value $R_{ij}$ in $R$ can be effectively predicted.

### Bayesian Deep Collaborative Matrix Factorization

In this section, we propose a Bayesian Deep Collaborative Matrix Factorization (BDCMF) for recommendation, the goal of which is to infer user latent matrix $U$, social factor latent matrix $G$, and item latent matrix $V$ given item content matrix $X$, user social matrix $S$, and feedback matrix $R$.

### The Proposed Model

Similar to (Ma et al. 2008), we consider user-item and social matrices, $R$ and $S$, sharing the same user latent matrix $U$, such that $R$ is factorized by user and item latent matrices, $U$ and $V$, via probabilistic matrix factorization (PMF) (Mnih and Salakhutdinov 2008), and $S$ is factorized by user latent and social factor latent matrices, $U$ and $G$. For items’ content information, since it can be very complex and various, we don’t know its real distribution. However, we know any distribution can be generated by mapping simple Gaussian through a sufficiently complicated function (Doersch 2016). In our proposed model, we consider item contents to be generated by their latent content vector through a generative network. The generative process of BDCMF is:

1. For each item $j$, draw its latent vector $v_j \sim \mathcal{N}(0, \lambda_v I_D)$.
2. For each social factor $k$, draw its latent vector, i.e., the social latent vector $g_k \sim \mathcal{N}(0, \lambda_s I_D)$.
3. For each item $j$:
   - (a) Draw item content latent vector $z_j \sim \mathcal{N}(0, \lambda_c I_D)$.
   - (b) Draw item content vector $p_v(x_j | z_i)\).
   - (c) Draw item latent offset $k_j \sim \mathcal{N}(0, \lambda_k I_D)$ and set the item latent vector as $v_j = z_j + k_j$. 

$\end{raw}
4. For each user-item pair \((i, j)\) in \(R\), draw \(R_{ij}\):

\[
R_{ij} \sim \mathcal{N}(u_i^T v_j, c_{ij}^{-1}). 
\]

5. For each user-social factor pair \((i, k)\) in \(S\), draw \(S_{ik}\):

\[
S_{ik} \sim \mathcal{N}(u_i^T g_k, \lambda^{-1}_{ik} e_{ik}^{-1}).
\]

In the process, \(\lambda, \nu, \lambda_u, \lambda_g\), and \(\lambda_q\) are the free parameters, respectively. Specifically, \(\lambda_q\) is a trade-off between the social matrix and user-item matrix for user latent factor (will be discussed later). Similar to (Wang and Blei 2011; Wang, Wang, and Yeung 2015), \(c_{ij}\) in Eq.1 and \(e_{ik}\) in Eq.2 serves as confident parameters for \(R_{ij}\) and \(S_{ik}\), respectively:

\[
c_{ij} = \left\{ \begin{array}{ll}
\varphi_1 & \text{if } R_{ij} = 1, \\
\varphi_2 & \text{if } R_{ij} = 0,
\end{array} \right.
\]

\[
e_{ik} = \left\{ \begin{array}{ll}
\varphi_3 & \text{if } S_{ik} = 1, \\
\varphi_4 & \text{if } S_{ik} = 0,
\end{array} \right.
\]

where \(\varphi_1 > \varphi_2 > 0\) and \(\varphi_3 > \varphi_4 > 0\) are free parameters.

In our model, we follow (Wang, Wang, and Yeung 2015; Purushotham, Liu, and Kuo 2012) to set \(\varphi_1 = \varphi_3 = 1\) and \(\varphi_2 = \varphi_4 = 0.1\). \(p_0(x_j | z_j)\) represents item content information and \(x_j\) is generated from latent content vector \(z_j\) through a generative neural network parameterized by \(\theta\). It should be noted that the specific form of the probability \(p_0(x_j | z_j)\) depends on the type of the item content vector. For instance, if \(x_j\) is binary vector, \(p_0(x_j | z_j)\) can be a multivariate Bernoulli distribution \(\text{Ber}(F_0(z_j))\) with \(F_0(z_j)\) being the highly no-linear function parameterized by \(\theta\).

According to the graphic model in Fig.1, the joint probability of \(R, S, X, U, V, G, Z\) can be represented as:

\[
p(O, Z) = \prod_{i=1}^{N} \prod_{j=1}^{M} \prod_{k=1}^{N} p(O_{ijk}, Z_{ijk}) = \prod_{i=1}^{N} \prod_{j=1}^{M} \prod_{k=1}^{N} p(O_{ijk}, Z_{ijk})
\]

\[
p(R_{ij} | u_i, v_j, p(S_{ik} | u_i, g_k), p(R_{ij} | u_i, v_j, p(S_{ik} | u_i, g_k))
\]

where \(O = \{R, S, X\}\) is the set of all observed variables, \(Z = \{U, V, G, Z\}\) is the set of all latent variables needed to be inferred, and \(O_{ijk} = \{R_{ij}, S_{ik}, x_j\}\) and \(Z_{ijk} = \{u_i, v_j, g_k, z_j\}\) for short.

**Bayesian Point Estimation**

Previous works (Wang, Wang, and Yeung 2015; Wang and Blei 2011) have shown that using an EM-style algorithm enables recommendation methods that integrate them to obtain high-quality latent vectors (in our case, \(\hat{U}\) and \(\hat{V}\)). Inspired by these work, in this section, we first derive an EM-style algorithm called BDCMF-1 from the view of Bayesian point estimation. The marginal log likelihood can be given by:

\[
\log p(O) = \log \int p(O, Z) dZ \geq \int q(Z) \log \frac{p(O, Z)}{q(Z)} dZ
\]

\[
= \int q(Z) \log p(O, Z) - \int q(Z) \log q(Z) = \mathcal{L}(q),
\]

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

\[
q(Z) = q(U)q(V)q(G)q(Z)
\]

\[
= \prod_{i=1}^{N} q(u_i) \prod_{j=1}^{M} q(v_j) \prod_{k=1}^{N} q(g_k) \prod_{j=1}^{M} q(z_j).
\]

For Bayesian point estimation, we assume the variational distribution of \(u_i\) is:

\[
q(u_i) = \prod_{d=1}^{D} \delta(U_{id} - \hat{U}_{id}),
\]

where \(\hat{U}_{id}\) are variational parameters and \(\delta\) is a Dirac delta function. Variational distributions of \(v_j\) and \(g_k\) are defined similarly. When \(U_{id}\) is discrete (The continuous situation will be discussed in next section), the entropy of \(u_i\) is:

\[
H(u_i) = -\int q(u_i) \log q(u_i)
\]

\[
= -\sum_{d=1}^{D} \delta(U_{id} - \hat{U}_{id}) \log \delta(U_{id} - \hat{U}_{id}) = 0.
\]

Similarly, \(H(v_j)\) and \(H(g_k)\) are 0 when the elements are discrete. Then the evidence lower bound \(\mathcal{L}(q)\) (Eq.6) can be written as:

\[
\mathcal{L}_{\text{point}}(\hat{U}, \hat{V}, \hat{G}, \theta, \phi) = \langle \log p(U)p(G)p(V|Z) \rangle_{q} - \text{KL}(q_0(Z|X)||p(Z)),
\]

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

For latent variables \(Z\), we use a generative network to represent the distribution \(p_\theta(X|Z)\). As discussed in (Kingma...
and Welling 2014), traditional variational-EM algorithm (Neal and Hinton 1998) is intractable to infer the posterior of \( Z \). Consequently, similar to VAE (Kingma and Welling 2014), we also introduce a variational distribution \( q_\phi(Z | X, R) \) to approximate the true posterior distribution \( p(Z | O) \). \( q_\phi(Z | X, R) \) is implemented by an inference neural network parameterized by \( \phi \) (see Fig.1). Specifically, for \( z_j \) we have:

\[
q(z_j) = q_\phi(z_j | x_j, R_{ij}) = \mathcal{N}(\mu_j, \text{diag}(\delta_j^2)),
\]

where the mean \( \mu_j \) and variance \( \delta_j \) are the outputs of the inference neural network.

Directly maximizing the ELBO (Eq. 10) involves solving parameters \( \tilde{\theta} \) and \( \phi \), which is intractable. To handle this problem, we approximate it by the variational E-step due to space limitations:

\[
\text{Variational E-step.} \quad \text{We first keep } \theta \text{ and } \phi \text{ fixed, then optimize evidence lower bound } \mathcal{L}_{\text{point}} \text{ with respect to } \tilde{\theta}, \tilde{\phi} \text{ and } \tilde{G}.
\]

\[
\mathcal{L}_{\text{point}}(\tilde{\theta}, \tilde{\phi}, \tilde{G}) = \sum_{j=1}^{M} \mathcal{L}(\tilde{\theta}, \tilde{\phi}, \hat{v}_j, \hat{u}_j, \hat{g}_k).
\]

\[
\text{Monte Carlo sampling as follows:}
\]

\[
\langle \log p_\theta(x_j | z_j) \rangle_{q_\phi(z_j | x_j, R_{ij})} = \frac{1}{L} \sum_{l=1}^{L} p_\theta(x_j | z^l_j),
\]

where \( L \) is the size of samplings, and \( z^l_j \) denotes the \( l \)-th sample, which is reparameterized to \( z^l_j = \hat{c}_j \odot \text{diag}(\delta_j^2) + \mu_j \).

Here \( \hat{c}_j \) is drawn from \( \mathcal{N}(0, I_D) \) and \( \odot \) is an element-wise multiplication. By using this reparameterization trick and Eq 11, \( \mathcal{L}(\theta, \phi; x_j, v_j, R_{ij}) \) in Eq 15 can be estimated by:

\[
\mathcal{L}(\theta, \phi; x_j, v_j, R_{ij}) \simeq \mathcal{L}(\tilde{\theta}, \tilde{\phi}, \hat{v}_j, \hat{u}_j, \hat{g}_k) + \text{constant}.
\]

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

\[
\mathcal{L}_{\text{point}}(\phi; \theta, X, V, R) \simeq \mathcal{L}(\tilde{\theta}, \tilde{\phi}, \hat{v}_j, \hat{u}_j, \hat{g}_k).
\]

Bayesian Posterior Estimation

We assume \( \tilde{\theta}, \tilde{\phi} \) and \( \tilde{G} \) are discrete variables. When these variables are continuous, as discussed in previous work (Welling, Chemudugunta, and Sutter 2008), the Bayesian point estimation will lead overly to optimistic estimates of test-set log-likelihood. This problem is exactly what pervious well-known recommendation models such CTR-based and CDL-based models (Wang, Wang, and Yeung 2015; Wang and Blei 2011; Purushotham, Liu, and Kuo 2012; Chen et al. 2014; Zhang et al. 2016) which use point estimation algorithm suffer from. In addition, point estimators leads to higher variance in prediction and ignore uncertainty in the model parameters. Unlike CTR-based and CDL-based recommendation methods, BDCMF can be easily estimated by a full Bayesian posterior estimator.

Here, we derived Bayesian posterior estimation learning algorithm called BDCMF-2. For Bayesian posterior estimation, we also assume variational distributions are matrix-wise independent:

\[
\mathcal{L}(\theta, \phi; x_j, v_j, R_{ij}) = \mathcal{L}(\theta, \phi; x_j, v_j, R_{ij}) + \mathcal{L}(\phi; \theta, x_j, v_j, R_{ij}) + \mathcal{L}(\theta; \phi, x_j, v_j, R_{ij}).
\]

\[
\mathcal{L}(\phi; \theta, x_j, v_j, R_{ij}) = \mathcal{L}(\phi; \theta, x_j, v_j, R_{ij}) + \mathcal{L}(\theta; \phi, x_j, v_j, R_{ij}).
\]

\[
\mathcal{L}(\theta; \phi, x_j, v_j, R_{ij}) = \mathcal{L}(\theta; \phi, x_j, v_j, R_{ij}) + \mathcal{L}(\phi; \theta, x_j, v_j, R_{ij}).
\]

For variational distributions \( p(z_j) \), we also introduce a variational distribution \( q_\phi(Z | X) \) to approximate the true posterior distribution \( p(Z | O) \) like Bayesian point estimation, \( q_\phi(Z | X) \) is parameterized by \( \phi \), an inference neural network (see Fig.1). For \( z_j \):

\[
q(z_j) = q_\phi(z_j | x_j, R_{ij}) = \mathcal{N}(\mu_j, \text{diag}(\delta_j^2)).
\]
It should be noted we do not assume the specific forms of variational distributions for \(q(u_i), q(v_j)\) and \(q(g_k)\) in Bayesian posterior estimation. Our goal is also to maximize the evidence lower bound \(\mathcal{L}(q)\) (Eq. 6). The algorithm also contains variational E-step and M-step.

**Variational E-step.** We fix \(\theta\) and \(\phi\), then optimize evidence lower bound (6) w.r.t. \(u_i, v_j\) and \(g_k\). Taking the derivative of lower bound (Eq. 6) w.r.t. \(q(u_i)\) and setting it to zero, then we can get:

\[
q(u_i) \propto \exp \{ \log p(O, Z) \}_{q(z \mid u_i)} = \mathcal{N}(u_i, \Lambda_i^u),
\]

where the expectation is taken with respect to the variational distributions over all latent variables excluding \(u_i\). Thus, the updating rules of \(u_i\) are:

\[
q(U) = \prod_{i=1}^{M} \mathcal{N}(u_i, \Lambda_i^u), \quad (22)
\]

\[
\Lambda_i^u \leftarrow (\langle V \rangle C_i (V)^\top + \sum_{j=1}^{M} c_{ij} \Lambda_j^u + \lambda_u \text{I}_D)^{-1},
\]

\[
\Lambda_j^u \leftarrow (\langle U \rangle C_j (U)^\top + \sum_{i=1}^{N} c_{ij} \Lambda_i^u + \lambda_u \text{I}_D)^{-1},
\]

\[
\Lambda_k^g \leftarrow (\lambda_q \langle U \rangle E_k (U)^\top + \sum_{i=1}^{N} e_{ik} \Lambda_i^u) + \lambda_g \text{I}_D)^{-1},
\]

\[
g_k \leftarrow \Lambda_k^g (\lambda_q (U) E_k S_k).
\]

Similarly, the updating rules of \(V\) and \(G\) are:

\[
q(V) = \prod_{j=1}^{N} \mathcal{N}(v_j, \Lambda_j^v), \quad (25)
\]

\[
\Lambda_j^v \leftarrow (\langle V \rangle C_j (V)^\top + \sum_{i=1}^{M} c_{ij} \Lambda_i^v + \lambda_v \text{I}_D)^{-1},
\]

\[
\Lambda_k^g \leftarrow (\lambda_g \langle U \rangle E_k (U)^\top + \sum_{i=1}^{N} e_{ik} \Lambda_i^v) + \lambda_g \text{I}_D)^{-1},
\]

\[
g_k \leftarrow \Lambda_k^g (\lambda_g (U) E_k S_k).
\]

**Variational M-step.** We only focus parameters \(\theta\) and \(\phi\). The M-step is as same as Bayesian point estimation. By using the reparameterization trick as we did in Bayesian point estimation, the objection function w.r.t \(\theta\) and \(\phi\) is:

\[
\mathcal{L}(q) = \text{constant} + \sum_{j=1}^{M} \mathcal{L}(\theta, \phi; x_j, v_j, R_{-j}) = \text{constant} + \sum_{j=1}^{M} \frac{\lambda_u}{2} \langle (v_j - z_j)^\top (v_j - z_j) \rangle_{q(z_j)} + 
\]

\[
\langle \log p(x_j | z_j) \rangle_{q(x_j | z_j, R_{-j})} - \text{KL}(q(\phi(z_j | x_j, R_{-j})) || p(z_j)).
\]

We can optimize it as the same as we did in Bayesian point estimation.

**Discussion**

According to Algorithm 1, the time complexity of updating \(g_k\) is \(O(ND^2 + D^3)\), where \(N\) is the number of users and \(D\) is the dimension of latent space. The time complexity of updating \(v_j\) is \(O(ND^2 + D^3 + LP^2)\), where \(L\) is the number of layers in neural network and \(P\) is the average dimension of these layers. The time complexity of updating \(u_i\) is \(O(MD^2 + ND^2 + D^3)\), where \(M\) is the number of items. Because \(\mathcal{V}C_i \mathcal{V}^\top\) in Eq. 12 can be rewritten as \(\mathcal{V}(C_i - \varphi_i I)\mathcal{V}^\top + \varphi_i C_i\mathcal{V}^\top\), the complexity of updating \(g_k\) decreases from \(O(ND^2 + D^3)\) to \(O(N_0 D^2 + D^3)\), where \(N_0\) is observed rating numbers with respect to item \(j\) which is relatively small in sparse matrix. \(\mathcal{U}C_j \mathcal{U}^\top\) and \(\mathcal{U}E_k \mathcal{U}^\top\) can be rewritten, similarly. In fact, the latent dimensions \(D\) is also relatively small (<200). Therefore, our BDCMF can be very efficient for large datasets. Additionally, our model is flexible to handle different types of contents in different scenarios by replacing the architecture of neural networks for inferring items’ latent content representations to the other neural networks such as recurrent neural network.

**Prediction**

After Algorithm 1 is converged, we predict the missing value \(R_{ij}\) in \(R\) by using the learned latent features \(u_i\) and \(v_j\):

\[
\hat{R}_{ij} = \langle R_{ij} \rangle = \langle (z_j) + (k_j) \rangle_{q(u_i)} = \langle v_j \rangle_{q(u_i)}^\top.
\]

For a new item that is not rated by any other users, the offset \(\epsilon_j\) is zero, and we can predict \(\hat{R}_{ij}\) by:

\[
\hat{R}_{ij}^* = \langle R_{ij} \rangle = \langle z_j \rangle_{q(u_i)}^\top.
\]

**Experiments**

**Experimental Setup**

**Research Questions.** The research questions guiding the remainder of the paper are: (RQ1) How does our proposed BDCMF-1 compare to state-of-the-art MF methods for CF on sparsity matrix? (RQ2) How do different parameter settings (e.g., the social parameter \(\lambda_s\) and content parameter \(\lambda_c\)) affect BDCMF-1? (RQ3) Does the Bayesian posterior estimation outperform Bayesian point estimation?

**Datasets.** In order to answer our research questions, we conduct experiments on two real-world datasets from...
Lastfm\(^1\) (lastfm-2k) and Delicious\(^2\) (delicious-2k) collected by Brusilovsky et al. (2010). Both of the datasets contain user-item, user-user, and user-tag-item relations. We first transform datasets 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. Similarly, for the Delicious dataset, if the user has bookmarked a URL (item), the user-item feedback is 1; otherwise, the feedback is 0. After doing so, Lastfm and Delicious datasets only contain 0.27% and 0.08% observed feedbacks, respectively, and are extremely sparse. We use items bag-of-word tag representations as their items content information.

**Baselines.** For fair comparisons, like that in our BDCMF-1 and BDCMF-2 (in what follows, we use BDCMF to refer to these two models), most baselines we used also incorporate user social information or item content information into matrix factorization.

**MF-based methods:**

1. **PMF.** This model (Mnih and Salakhutdinov 2008) is a famous MF method, and only uses user-item feedback matrix.

2. **SoRec.** This model (Ma et al. 2008) jointly decomposes user-user social matrix and user-item feedback matrix to learn user and item latent representations.

**Content-based methods:**

3. **Collaborative topic regression (CTR).** This model (Wang and Blei 2011) utilizes topic model and matrix factorization to learn latent representations of users and items.

4. **Collaborative deep learning (CDL).** This model (Wang, Wang, and Yeung 2015) utilizes stack denoising autoencoder to learn latent items’ content representations, and incorporates them into probabilistic matrix factorization.

**Hybrid methods:**

5. **CTR-SMF.** This model (Purushotham, Liu, and Kuo 2012) incorporates topic modeling and probabilistic MF of social networks.

6. **PoissonMF-CS** This model (de Souza da Silva, Langseth, and Ramampiaro 2017), jointly models use social trust, item content and user’s preference using Poisson matrix factorization framework. It is a state-of-the-art MF method for Top-N recommendation on the Lastfm dataset.

**Neural network-based method:**

7. **Neural Matrix Factorization (NeuMF).** This model (He et al. 2017) is a state-of-the-art CF method, which utilizes neural network to model the interaction between user and item features.

**Settings.** For fair comparisons, We first set the parameters for PMF, SoRec, CTR, CTR-SMF, CDL, NeuMF via five-fold cross validation. For PMF, we set \(D, \lambda_u\) and \(\lambda_v\) as 50, 0.01 and 0.001, respectively. The parameter settings for SoRec are \(\lambda_u=10, \lambda_a=\lambda_v=\lambda_c=0.001\). For CTR, we find it achieve best performance when \(D=50, \lambda_u=0.1, \lambda_a=100, a=1\) and \(b=0.01\). CTR-SMF yields the best performance when \(D=75, \lambda_a=0.1, \lambda_v=100, \lambda_c=100, a=1\) and \(b=0.01\). For CDL, it achieves the best performance when we set \(a=1, b=0.01, D=50, \lambda_u=1, \lambda_v=10, \lambda_a=1000, \lambda_c=0.0001\). For NeuMF, we set \(D=50\), and the last hidden layer is 16. For PoissonMF-CS, we set \(\lambda_u=0.1\) and \(\lambda_v=0.1\). For our model BDCMF, we set \(D=50\). We set \(\lambda_u=1, \lambda_v=10\) for dataset Lastfm, and set \(\lambda_u=0.1, \lambda_v=10\) for dataset Delicious. To evaluate our model performance on extreme sparse matrix, we use 90% of dataset to train our model and the remainder for testing. The dimensions of the layers of the network for BDCMF are \((L+N)-200-100(z_j)-100-1(L+N)\), where \(L\) and \(N\) are the dimension of \(x_j\) and \(R_j\). The “-“ denotes the next layer in neural networks.

**Evaluation Metrics.** The metrics we used are Recall@K, NDCG@K and MAP@K which are common metrics for recommendation and their definitions can be found at (Croft, Metzler, and Strohman 2015; Liang and de Rijke 2016). We report the average performance over all users on the metrics.

**Experimental Results and Discussions**

**Overall Performance (RQ1).** Table 2 shows the performance of our BDCMF-1 and the baselines using the two datasets. According to Table 2, we have following findings: a) BDCMF-1 outperforms the baselines in terms of all matrices on Lastfm, which demonstrates the effectiveness of our method of inferring the latent factors of users and items. b) For more sparse dataset, Delicious, BDCMF-1 also achieves the best performance, which demonstrates our model can effectively handle matrix sparsity problem. c) We can see both methods that utilize content and social information (BDCMF-1 and PoissonMF-CS) outperform the others (CDL, CTR, SoRec and PMF), which demonstrates incorporating content and social information can effectively alleviate matrix sparse problem. d) Our BDCMF-1 outperforms state-of-the-art PoissonMF-CS, though they are both Bayesian generative model. The reason is that our BDCMF-1 incorporates neural network into Bayesian generative model, which makes it have powerful non-linearity to model item content’s latent representation.

**Impact of Parameters (RQ2).** Fig. 2(a) and 2(b) show the contour of Recall@50 by varying content parameters \(\lambda_v\) and social parameters \(\lambda_u\) on Lastfm and Delicious datasets. As we can see, BDCMF-1 achieves the best recommendation performance when \(\lambda_u=0.1\) and \(\lambda_v=10\) in Lastfm. We also can see that the performance is worse when \(\lambda_v > 10\), as \(\lambda_v\) can control how much item content information is incorporated into item latent vector, and Lastfm is a social media dataset where people have similar preferences with

<table>
<thead>
<tr>
<th>Table 1: Statistics of the datasets.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datasets</td>
</tr>
<tr>
<td>users</td>
</tr>
<tr>
<td>items</td>
</tr>
<tr>
<td>tags</td>
</tr>
<tr>
<td>user-items relations</td>
</tr>
<tr>
<td>user-user relations</td>
</tr>
<tr>
<td>user-tags-items</td>
</tr>
</tbody>
</table>

\(^1\)http://www.lastfm.com.  
<table>
<thead>
<tr>
<th></th>
<th>Lastfm Dataset</th>
<th>Delicious Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Recall@20</td>
<td>Recall@50</td>
</tr>
<tr>
<td>PMF</td>
<td>0.0923</td>
<td>0.1328</td>
</tr>
<tr>
<td>SoRec</td>
<td>0.1088</td>
<td>0.1524</td>
</tr>
<tr>
<td>CTR</td>
<td>0.1192</td>
<td>0.1624</td>
</tr>
<tr>
<td>CTR-SMF</td>
<td>0.1232</td>
<td>0.1832</td>
</tr>
<tr>
<td>CDL</td>
<td>0.1346</td>
<td>0.2287</td>
</tr>
<tr>
<td>NeuMF</td>
<td>0.1517</td>
<td>0.2584</td>
</tr>
<tr>
<td>PoissonMF-CS</td>
<td>0.1482</td>
<td>0.2730</td>
</tr>
<tr>
<td>BDCMF-1(ours)</td>
<td><strong>0.1625</strong></td>
<td><strong>0.3026</strong></td>
</tr>
</tbody>
</table>

Figure 2: The contour of Recall@50 by varying $\lambda_v$ and $\lambda_q$ on datasets Lastfm and Delicious.

Their friends. $\lambda_v$ can control how much social information is incorporated into user latent factor. Thus, our BDCMF-1 is very sensitive to $\lambda_v$ and $\lambda_q$. For Delicious dataset, Fig. 2(b) shows BDCMF-1 achieves the best performance on Recall@50 when $\lambda_v = 1$ and $\lambda_q = 10$. Fig. 2(a) and 2(b) show that we can balance the content information and social information by varying $\lambda_v$ and $\lambda_q$, leading to better recommendation performance.

To figure out the impact of the latent dimension $D$, we vary $D$ and see the performance. As shown in Fig. 3, BDCMF-1 outperforms all the baselines with all dimensions, which demonstrates BDCMF-1 can learn better latent representations of users and items than the baselines regardless of their dimensional size.

The Effectiveness of Bayesian posterior estimation (RQ3). To figure out the effectiveness of Bayesian posterior estimation, we make comparison between BDCMF-1 and BDCMF-2 on Lastfm dataset. Results on Delicious show the same trend and thus we omit it here. Performance of BDCMF-1 and BDCMF-2 over iterations on Lastfm is observed as: a) With more iterations, the ELBO of BDCMF-1 and BDCMF-2 gradually increases. BDCMF-2 achieves bigger ELBO than BDCMF-1, which illustrates BDCMF-2 can fit data better (the margin likelihood of data is more higher). b) For recommendation performance, BDCMF-2 outperforms BDCMF-1 in terms of all metrics. Specifically, BDCMF-2 betters with a 10.4% relative improvement on Recall@20. c) When the number of iterations $> 30$, BDCMF-1 suffers severe overfitting problem (Recall@20 scores begin to decrease, although the ELBO keeps increasing). In contrast, BDCMF-2 does not suffer this overfitting problem. These observations demonstrate BDCMF-2 can learn better latent representations than BDCMF-1.
Conclusions

In this paper we studied the problem of inferring latent factors of users and items in the social recommender system. We have proposed a novel Bayesian deep collaborative matrix factorization model, BDCMF, which incorporates various item content information and user social relationship into a full Bayesian framework. To model item’s latent content vector, we use a generative network to model the generative process of item content. To effectively infer latent factors of users and items, we first derived an EM algorithm from a Bayesian point estimation perspective. Due to the full Bayesian nature of our proposed model and the drawbacks from a Bayesian point estimation perspective. Due to the full posterior distribution of the latent factors of users and items, we have conducted several experiments on two public datasets. Experimental results show that our BDCMF can effective infer latent factors of users and items, and the Bayesian posterior estimation is more robust than point estimation. In the future, we will utilize the proposed model to deal with other information retrieval task such as user profiling (Liang 2018; Liang et al. 2018b; Liang, Yilmaz, and Kanoulas 2018) and social network analysis.

References