Dynamic Bayesian Logistic Matrix Factorization for Recommendation with Implicit Feedback

Yong Liu[†‡], Lifan Zhao[†], Guimei Liu[†], Xinyan Lu[♯], Peng Gao[♯], Xiao-Li Li[†], Zhihui Jin[♯]
[†]Institute for Infocomm Research, A*STAR, Singapore
[‡]NTUC Link Pte. Ltd., Singapore
[♯]Tencent Social Ads, China
[†]liuysc@acm.org, zhao0145@e.ntu.edu.sg, {liug, xlli}@i2r.a-star.edu.sg,
[♯]{xinyanlu, stephengao, rickjin}@tencent.com

Abstract

Matrix factorization has been widely adopted for recommendation by learning latent embeddings of users and items from observed user-item interaction data. However, previous methods usually assume the learned embeddings are static or homogeneously evolving with the same diffusion rate. This is not valid in most scenarios, where users' preferences and item attributes heterogeneously drift over time. To remedy this issue, we have proposed a novel dynamic matrix factorization model, named Dynamic Bayesian Logistic Matrix Factorization (DBLMF), which aims to learn heterogeneous user and item embeddings that are drifting with inconsistent diffusion rates. More specifically, DBLMF extends logistic matrix factorization to model the probability a user would like to interact with an item at a given timestamp, and a diffusion process to connect latent embeddings over time. In addition, an efficient Bayesian inference algorithm has also been proposed to make DBLMF scalable on large datasets. The effectiveness of the proposed method has been demonstrated by extensive experiments on real datasets, compared with the state-of-the-art methods.

1 Introduction

Personalized recommendation systems have been extensively used in our daily lives. A lot of algorithms and real systems have been developed [Shi *et al.*, 2014]. Practically, matrix factorization models are the most successful approaches that have been widely applied to solve various recommendation problems [Hu *et al.*, 2014; Liu *et al.*, 2016; 2017]. From the data perspective, two types of data, i.e., users' explicit feedback and implicit feedback, are usually used to build recommendation models. Explicit feedback refers to users' explicit ratings to items, e.g., the 1 to 5 scale business ratings in Yelp [Hu *et al.*, 2014]. Recommendation models based on explicit feedback data mainly focus on predicting the exact rating value that a user would like to assign to an item. The implicit feedback is derived from users' interactions with items, e.g., the number of times a user has visited a location [Liu *et al.*, 2014]. Recommendation models based on implicit feedback data usually focus on predicting whether a user would interact with an item. However, explicit feedback data is often very limited, where implicit feedback may be useful for learning users' preferences. Thus, in this work, we exploit implicit feedback data to build recommendation model.

Various matrix factorization methods have been proposed to build recommendation models based on implicit feedback data [Rendle *et al.*, 2009; Pan and Chen, 2013; Johnson, 2014; Liu *et al.*, 2015; He *et al.*, 2016], where the latent embeddings of users and items are assumed to be static. In practice, this assumption is generally not valid, since users' preferences and item attributes are constantly drifting over time. To exploit the temporal dynamics for recommendation, different dynamic matrix factorization models have been proposed [Xiong *et al.*, 2010; Gultekin and Paisley, 2014; Charlin *et al.*, 2015; Lian *et al.*, 2016; Hosseini *et al.*, 2017]. The key assumptions of these methods are that the latent embeddings of users and items are homogeneously evolving with the same diffusion rate.

However, different dimensions of the user and item latent embeddings shall evolve heterogeneously with inconsistent diffusion rates. More concretely, certain dimension of embeddings that reflects users' long-term preferences (e.g., payment method) or items' stable attributes (e.g., item category) may evolve slowly with time. On the other side, the dimensions of embeddings reflecting users' short-term preferences (e.g., usage of social media) or items' dynamic attributes (e.g., monthly popularity) should quickly change over time. In this sense, it would be more desirable to simultaneously estimate the latent embeddings and their corresponding diffusion rates, where more accurate estimations of latent embeddings can be expected. This problem, however, has not yet been adequately investigated in existing literature.

In this paper, we have proposed a novel matrix factorization method, named Dynamic Bayesian Logistic Matrix Factorization (DBLMF), to bridge this gap. More specifically, DBLMF extends logistic matrix factorization [Johnson, 2014] to model the dynamic likelihood that a user would in-

teract with an item at a given timestamp. To characterize the temporal dynamics, a temporal diffusion process is developed for learning dynamic latent embeddings of users and items. In particular, the latent embeddings and the diffusion process are modeled in a probabilistic manner with model conjugacy. Therefore, the latent embeddings and the diffusion rate of each dimension can both be desirably inferred from users' implicit feedback data. To infer all the unknown parameters, we have proposed an efficient variational Bayesian inference algorithm, which adopts a sampling strategy to scale up the proposed method. In addition, we have also conducted extensive experiments on real datasets to demonstrate the effectiveness of the proposed algorithm. The experimental results indicate the proposed DBLMF method usually achieves better results than several state-of-the-art recommendation algorithms, in terms of Precision, Recall, and Normalized Discounted Cumulative Gain (NDCG).

2 Related Work

In the literature, different dynamic matrix factorization methods have been developed to explore users' dynamic preferences. For example, a set of time-aware recommendation algorithms based on users' explicit feedback have been proposed, which assigned higher weights to recent observations [Liu *et al.*, 2010], or incorporated the temporal information into latent factor models [Koren, 2009; Xiong *et al.*, 2010]. Moreover, Gultekin and Paisley proposed the collaborative Kalman filter that modeled each low-dimensional latent embedding as a multidimensional Brownian motion [Gultekin and Paisley, 2014]. Yu et al. proposed a temporal regularized matrix factorization framework for temporal learning and forecasting based on high dimensional time series data with missing values [Yu *et al.*, 2016]. For implicit feedback data, the Gaussian process had been proposed to capture users' dynamic preferences [Liu, 2015]. In addition, Charlin et al. developed a dynamic Poisson factorization model that exploited Kalman filter to model evolving latent embeddings and used Poisson distribution to model the user-item interactions [Charlin *et al.*, 2015]. Du et al. developed a convex optimization framework that connected self-exciting point processes and low-rank models to exploit recurrent user activity patterns for recommendation [Du *et al.*, 2015]. Recently, Wang et al. proposed a latent feature process method to study the co-evolving patterns of user and item features [Wang *et al.*, 2016]. Hosseini et al. proposed a recurrent Poisson factorization framework that utilized Poisson process to model users' recurrent activity patterns from their implicit feedback data for recommendation [Hosseini *et al.*, 2017].

3 The Proposed Recommendation Model

In this work, we consider the dynamic recommendation problem based on the implicit feedback data generated by M users $\{u_i\}_{i=1}^M$ over N items $\{v_j\}_{j=1}^N$. Firstly, we use a binary matrix $\boldsymbol{Y}^t \in \mathbb{R}^{M\times N}$ to describe the interactions between users and items at time t. If a user u_i has interacted with an item v_j at time t, we set y_{ij}^t to 1; otherwise, we set y_{ij}^t to 0. Moreover, we denote the set of observed user-item interactions at time t by $\mathcal{O}^t$. We use $\mathcal{E}_i^t = \{v_j | y_{ij}^t = 1\}_{j=1}^N$ to denote the set of items that have interactions with u_i at time t. Similarly, we use $\mathcal{F}_j^t = \{u_i | y_{ij}^t = 1\}_{i=1}^M$ to denote the set of users that have interactions with v_j at time t. Following traditional matrix factorization methods, we use a vector $\boldsymbol{u}_i^t \in \mathbb{R}^{1\times d}$ to denote the embedding of u_i at time t, where d is the dimensionality of latent space and $d \ll \min(M, N)$. We denote the latent embedding of v_j at time t by $\boldsymbol{v}_j^t \in \mathbb{R}^{1\times d}$. Following [Johnson, 2014], we define the probability that u_i would like to interact with v_j at time t as follows:

$$p_{ij}^t = \sigma(\boldsymbol{u}_i^t \boldsymbol{v}_j^{t\top}), \tag{1}$$

where $\sigma(x) = 1/(1+e^{-x})$. The likelihood of the observed user-item interactions at time t can be defined as follows:

$$p(\boldsymbol{Y}^t | \boldsymbol{U}^t, \boldsymbol{V}^t) = \prod_{i=1}^{M}\prod_{j=1}^{N} \sigma(\boldsymbol{u}_i^t \boldsymbol{v}_j^t)^{w_{ij}^t \mathbb{I}[y_{ij}^t=1]} \sigma(-\boldsymbol{u}_i^t \boldsymbol{v}_j^t)^{\mathbb{I}[y_{ij}^t=0]}, \tag{2}$$

where w_{ij}^t is a dynamic weight, and $\mathbb{I}[\cdot]$ is an indicator function. Similar to [Johnson, 2014], we empirically set $w_{ij}^t > 1$. In other words, we assign larger weights to positive user-item interactions (i.e., $y_{ij}^t = 1$) than "negative" user-item interactions (i.e., $y_{ij}^t = 0$), to balance the positive and "negative" interactions for better results. More sophisticated approaches, e.g., the re-weighted probabilistic models proposed in [Wang *et al.*, 2017], can also be integrated into Eq. (2).

As discussed earlier, the user and item embeddings should evolve with time in a heterogeneous way. Thus, we develop a temporal diffusion process to model the heterogeneous evolution patterns of latent embeddings. Specifically, the following priors are imposed on the latent vectors $\boldsymbol{u}_i^{t+1}$ and $\boldsymbol{v}_j^{t+1}$,

$$\begin{aligned} p(\boldsymbol{u}_i^{t+1} | \boldsymbol{u}_i^t) &\propto \mathcal{N}(\boldsymbol{u}_i^t, \boldsymbol{\alpha}^{-1}) \mathcal{N}(\boldsymbol{0}, \sigma_0^{-1}\boldsymbol{I}), \\ p(\boldsymbol{v}_j^{t+1} | \boldsymbol{v}_j^t) &\propto \mathcal{N}(\boldsymbol{v}_j^t, \boldsymbol{\beta}^{-1}) \mathcal{N}(\boldsymbol{0}, \sigma_0^{-1}\boldsymbol{I}), \end{aligned} \tag{3}$$

where $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$ is the Gaussian distribution with mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma}$. For simplicity, we set $\boldsymbol{\alpha} \in \mathbb{R}^{d\times d}$ and $\boldsymbol{\beta} \in \mathbb{R}^{d\times d}$ to be diagonal matrices. The diagonal elements are $\{\alpha_k\}_{k=1}^d$ and $\{\beta_k\}_{k=1}^d$. Particularly, the k^{th} diagonal entries in $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ control the drifting rates for the k^{th} dimension of users' and items' embeddings, respectively. At time $t = 1$, we define $p(\boldsymbol{u}_i^1 | \boldsymbol{u}_i^0) = \mathcal{N}(\boldsymbol{0}, \sigma_0 \boldsymbol{I})$ and $p(\boldsymbol{v}_j^1 | \boldsymbol{v}_j^0) = \mathcal{N}(\boldsymbol{0}, \sigma_0 \boldsymbol{I})$. In addition, we also place the following Gamma priors on the k^{th} diagonal elements of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$,

$$\alpha_k \sim \Gamma(a_0, b_0), \quad \beta_k \sim \Gamma(a_1, b_1), \tag{4}$$

where $\Gamma(a, b)$ is the Gamma distribution with shape parameters a and inverse scale parameter b.

Let $\boldsymbol{Y} \in \mathbb{R}^{T\times M\times N}$ denote the tensor of observed interactions across all times, where T is the number of timestamps. Similarly, $\boldsymbol{U} \in \mathbb{R}^{T\times M\times d}$ and $\boldsymbol{V} \in \mathbb{R}^{T\times N\times d}$ are used to denote the tensors of user and item latent embeddings across all times, respectively. Then, the joint probability distribution of the observations can be defined as follows:

$$\begin{aligned} p(\boldsymbol{Y}, \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{\alpha}, \boldsymbol{\beta}) =& p(\boldsymbol{\alpha}) p(\boldsymbol{\beta}) \prod_{t=1}^{T} p(\boldsymbol{Y}^t | \boldsymbol{U}^t, \boldsymbol{V}^t) \\ & \prod_{t=0}^{T-1} p(\boldsymbol{U}^{t+1} | \boldsymbol{U}^t, \boldsymbol{\alpha}) p(\boldsymbol{V}^{t+1} | \boldsymbol{V}^t, \boldsymbol{\beta}). \end{aligned} \tag{5}$$

Based on the above-described probabilistic model, we employ a full Bayesian treatment to infer the unknown parameters from this model.

3.1 Variational Bayesian Inference

For brevity, we denote the parameter set $\{\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{\alpha}, \boldsymbol{\beta}\}$ by $\boldsymbol{\Theta}$. In Bayesian methodology, the objective is to obtain the following full posterior,

$$p(\boldsymbol{\Theta}|\boldsymbol{Y}) = \frac{p(\boldsymbol{Y}, \boldsymbol{\Theta})}{\int p(\boldsymbol{Y}, \boldsymbol{\Theta})\, \mathrm{d}\boldsymbol{\Theta}}. \tag{6}$$

However, the normalization term is intractable due to multidimensional integral involved. We resort to variational inference (VI) [Blei *et al.*, 2017] to approximate the posterior distribution $p(\boldsymbol{\Theta}|\boldsymbol{Y})$ with factorisable assumption on the approximated posterior $q(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{\alpha}, \boldsymbol{\beta})$, which can be expressed as follows:

$$q(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = q(\boldsymbol{\alpha})q(\boldsymbol{\beta}) \prod_{t=1}^{T} q(\boldsymbol{U}^t) \prod_{t=1}^{T} q(\boldsymbol{V}^t). \tag{7}$$

The objective is to minimize the Kullback-Leibler (KL) divergence between the two distributions $q(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ and $p(\boldsymbol{\Theta}|\boldsymbol{Y})$. Based on derivation, the approximated posterior minimizing the KL divergence can be expressed as follows:

$$\log q(\Theta_i) = \mathbb{E}[\log p(\boldsymbol{Y}, \boldsymbol{\Theta})]_{j \neq i} + \text{const.} \tag{8}$$

where $\mathbb{E}[\cdot]_{j \neq i}$ represents the expectation taken with respect to (w.r.t.) all $\boldsymbol{\Theta}$ except for Θ_i [Blei *et al.*, 2017].

For probabilistic models that are conjugate, the update rules can be derived based on Eq. (8). However, in the proposed model, the likelihood function is a logistic likelihood function, which is not conjugate with Gaussian priors on the embeddings $\boldsymbol{U}$ and $\boldsymbol{V}$. To address this issue, the following Gaussian lower bound on the logistic function is employed to approximate the likelihood [Jaakkola and Jordan, 1997],

$$\sigma(x) \geq \sigma(\xi) e^{\frac{x-\xi}{2} - \lambda(\xi)(x^2 - \xi^2)}, \tag{9}$$

where $\lambda(\xi) = \frac{1}{2\xi}(\sigma(\xi) - \frac{1}{2})$, and ξ is an auxiliary variable that needs to be adjusted to make the bound tight at $x = \pm\xi$. By substituting Eq. (9) into the likelihood in Eq. (2), the lower bound of likelihood function $\log p(\boldsymbol{Y}^t|\boldsymbol{U}^t, \boldsymbol{V}^t)$ is as follows:

$$\begin{aligned}
&\log p(\boldsymbol{Y}^t|\boldsymbol{U}^t, \boldsymbol{V}^t) \geq \\
&\sum_{i=1}^{M} \sum_{v_j \in \mathcal{E}_i^t} w_{ij}^t \big[\frac{\boldsymbol{u}_i^t \boldsymbol{v}_j^{t\top}}{2} - \lambda(\xi_{ij}^t)(\boldsymbol{u}_i^t \boldsymbol{v}_j^{t\top})^2 + \varphi(\xi_{ij}^t)\big] \\
&- \sum_{i=1}^{M} \sum_{v_j \notin \mathcal{E}_i^t} \big[\frac{\boldsymbol{u}_i^t \boldsymbol{v}_j^{t\top}}{2} + \lambda(\xi_{ij}^t)(\boldsymbol{u}_i^t \boldsymbol{v}_j^{t\top})^2 - \varphi(\xi_{ij}^t)\big],
\end{aligned} \tag{10}$$

where $\varphi(\xi_{ij}^t) = \log \sigma(\xi_{ij}^t) - \frac{\xi_{ij}^t}{2} + \lambda(\xi_{ij}^t)(\xi_{ij}^t)^2$. The joint probability distribution can be approximated by replacing the likelihood function with its lower bound.

Based on the updating rule in Eq. (8), the model parameters at time t can be updated in the following iterative fashion:

Updating $\boldsymbol{U}^t$ and $\boldsymbol{V}^t$: To obtain the optimal approximated posterior, we first assume $q(\boldsymbol{U}^t)$ is fully factorized as $q(\boldsymbol{U}^t) = \prod_{i=1}^{M} \prod_{k=1}^{d} q(u_{ik}^t)$. Then, we substitute the joint probability into Eq. (8), and obtain the optimal variational distribution for the element u_{ik}^t as follows:

$$\begin{aligned}
&\log q^*(u_{ik}^t) \propto \\
&\sum_{v_j \in \mathcal{E}_i^t} w_{ij}^t u_{ik}^t \big[\frac{\mathbb{E}[v_{jk}^t]}{2} - \lambda(\xi_{ij}^t)(u_{ik}^t \mathbb{E}[(v_{jk}^t)^2] + 2\mathbb{E}[v_{jk}^t \tau_{ijk}^t])\big] \\
&- \sum_{v_j \notin \mathcal{E}_i^t} u_{ik}^t \big[\frac{\mathbb{E}[v_{jk}^t]}{2} + \lambda(\xi_{ij}^t)\big(u_{ik}^t \mathbb{E}[(v_{jk}^t)^2] + 2\mathbb{E}[v_{jk}^t \tau_{ijk}^t]\big)\big] \\
&- \frac{\alpha_k}{2}(u_{ik}^t)^2 + \alpha_k u_{ik}^t \mathbb{E}[u_{ik}^{t-1}] - \frac{\sigma_0}{2}(u_{ik}^t)^2,
\end{aligned} \tag{11}$$

where $\tau_{ijk}^t = \sum_{\ell \neq k} u_{i\ell}^t v_{j\ell}^t$. Due to model conjugacy, it can be known from Eq. (11) that the variational distribution shall follow a Gaussian distribution $\mathcal{N}(u_{ik}^t|\tilde{u}_{ik}^t, \Sigma_{ik}^t)$, where the mean and variance are as follows:

$$\begin{aligned}
\tilde{u}_{ik}^t =& \big[\frac{1}{2} \sum_{v_j \in \mathcal{E}_i^t} w_{ij}^t \big(\tilde{v}_{jk}^t - 4\lambda(\xi_{ij}^t)\tilde{v}_{jk}^t \tilde{\tau}_{ijk}^t\big) \\
&- \frac{1}{2} \sum_{v_j \notin \mathcal{E}_i^t} \big(\tilde{v}_{jk}^t + 4\lambda(\xi_{ij}^t)\tilde{v}_{jk}^t \tilde{\tau}_{ijk}^t\big) + \tilde{u}_{ik}^{t-1}\alpha_k\big]\Sigma_{ik}^t, \\
\Sigma_{ik}^t =& \big[\alpha_k + \sigma_0 + 2 \sum_{v_j \in \mathcal{E}_i^t} w_{ij}^t \lambda(\xi_{ij}^t)((\tilde{v}_{jk}^t)^2 + \Omega_{jk}^t) \\
&+ 2 \sum_{v_j \notin \mathcal{E}_i^t} \lambda(\xi_{ij}^t)((\tilde{v}_{jk}^t)^2 + \Omega_{jk}^t)\big]^{-1},
\end{aligned} \tag{12}$$

where $\tilde{\tau}_{ijk}^t = \sum_{\ell \neq k} \tilde{u}_{i\ell}^t \tilde{v}_{j\ell}^t$. Similarly, we also assume $q(\boldsymbol{V}^t)$ is fully factorized as $q(\boldsymbol{V}^t) = \prod_{j=1}^{N} \prod_{k=1}^{d} q(v_{jk}^t)$. Then, we can obtain the variational distribution $q(v_{jk}^t)$ in a similar way. It can be seen that $q(v_{jk}^t)$ also follows a Gaussian distribution $\mathcal{N}(v_{jk}^t|\tilde{v}_{jk}^t, \Omega_{jk}^t)$ with mean and variance as follows:

$$\begin{aligned}
\tilde{v}_{jk}^t =& \big[\frac{1}{2} \sum_{u_i \in \mathcal{F}_j^t} w_{ij}^t \big(\tilde{u}_{ik}^t - 4\lambda(\xi_{ij}^t)\tilde{u}_{ik}^t \tilde{\tau}_{ijk}^t\big) \\
&- \frac{1}{2} \sum_{u_i \notin \mathcal{F}_j^t} \big(\tilde{u}_{ik}^t + 4\lambda(\xi_{ij}^t)\tilde{u}_{ik}^t \tilde{\tau}_{ijk}^t\big) + \tilde{v}_{jk}^{t-1}\alpha_k\big]\Omega_{jk}^t, \\
\Omega_{jk}^t =& \big[\beta_k + \sigma_0 + 2 \sum_{u_i \in \mathcal{F}_j^t} w_{ij}^t \lambda(\xi_{ij}^t)((\tilde{u}_{ik}^t)^2 + \Sigma_{ik}^t) \\
&+ 2 \sum_{u_i \notin \mathcal{F}_j^t} \lambda(\xi_{ij}^t)((\tilde{u}_{ik}^t)^2 + \Sigma_{ik}^t)\big]^{-1}.
\end{aligned} \tag{13}$$

Updating $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$: Based on model conjugacy, we can see that the variational distribution for the k^{th} element of $\boldsymbol{\alpha}$ shall follow a Gamma distribution: $q(\alpha_k) \sim \Gamma(\tilde{a}_{0k}, \tilde{b}_{0k})$, where

$$\begin{aligned}
\tilde{a}_{0k} &= a_0 + \frac{M}{2}, \\
\tilde{b}_{0k} &= b_0 + \frac{1}{2} \sum_{i=1}^{M} [(\tilde{u}_{ik}^t - \tilde{u}_{ik}^{t-1})^2 + \Sigma_{ik}^t + \Sigma_{ik}^{t-1}].
\end{aligned} \tag{14}$$

The parameter α_k can be updated by using the mean as $\tilde{a}_{0k}/\tilde{b}_{0k}$. Similarly, the variational distribution of β_k also follows a Gamma distribution as: $q(\beta_k) \sim \Gamma(\tilde{a}_{1k}, \tilde{b}_{1k})$, where

$$\tilde{a}_{1k} = a_1 + \frac{N}{2},$$
$$\tilde{b}_{1k} = b_1 + \frac{1}{2}\sum_{j=1}^{N}[(\tilde{v}_{jk}^t - \tilde{v}_{jk}^{t-1})^2 + \Omega_{jk}^t + \Omega_{jk}^{t-1}]. \quad (15)$$

Then, β_k can be updated using the mean as $\tilde{a}_{1k}/\tilde{b}_{1k}$.

Updating ξ_{ij}^t: Since there is no prior assigned to the auxiliary parameter ξ_{ij}^t, the optimal value of ξ_{ij}^t can be obtained by maximizing the expected log-likelihood function, which can be expressed as follows:

$$\mathcal{L}(\xi_{ij}^t) = \mathbb{E}[\log p(y_{ij}^t | \boldsymbol{U}_i^t, \boldsymbol{V}_j^t, \xi_{ij}^t)]. \quad (16)$$

Taking derivative of the expected log-likelihood w.r.t. ξ_{ij}^t and setting it to zero, the optimal value of ξ_{ij}^t can be obtained as

$$\xi_{ij}^t = \big[\sum_{k=1}^{d}\big((\Sigma_{ik}^t)^2 + (\tilde{u}_{ik}^t)^2\big)\big((\Omega_{jk}^t)^2 + (\tilde{v}_{jk}^t)^2\big) + (\sum_{k=1}^{d}\tilde{u}_{ik}^t\tilde{v}_{jk}^t)^2 - \sum_{k=1}^{d}(\tilde{u}_{ik}^t\tilde{v}_{jk}^t)^2\big]^{\frac{1}{2}}. \quad (17)$$

The details of the proposed optimization algorithm are summarized in Algorithm 1. In this algorithm, the time complexity of each iteration is $\mathcal{O}(M \cdot N \cdot d + d \cdot (M + N))$. Thus, it is not scalable to large datasets. To remedy this issue, we propose the following sampling based inference algorithm.

3.2 Inference by Uniform Sampling

The updating rule in Eq. (11) can be rewritten as follows:

$$\log q^*(u_{ik}^t) \propto u_{ik}^t\Big[\alpha_k\mathbb{E}[u_{ik}^{t-1}] + \sum_{v_j\in\mathcal{E}_i^t}\mathbb{E}\big[\frac{(w_{ij}^t+1)v_{jk}^t}{2} - (w_{ij}^t-1)\rho_{ijk}^t\big] - \underbrace{\sum_{j=1}^{N}\mathbb{E}\big[\frac{v_{jk}^t}{2} + \rho_{ijk}^t\big]}_{\psi_{ik}^t} - \frac{(\alpha_k+\sigma_0)u_{ik}^t}{2}\Big], \quad (18)$$

where $\rho_{ijk}^t = \lambda(\xi_{ij}^t)[u_{ik}^t(v_{jk}^t)^2 + 2v_{jk}^t\tau_{ijk}^t]$. In practice, a user u_i usually only interacts with a small fraction of items, i.e., $|\mathcal{E}_i^t| \ll N$, where $|\cdot|$ is the cardinality of a set. Therefore, the main computation portion in Eq. (18) is the expectation ψ_{ik}^t over all items. To improve computation efficiency, we propose to approximate ψ_{ik}^t by uniformly sampling over all items as $\psi_{ik}^t \approx \frac{N}{|\mathcal{S}_{ik}^t|}\sum_{v_j\in\mathcal{S}_{ik}^t}\mathbb{E}\big[\frac{v_{jk}^t}{2} + \rho_{ijk}^t\big]$, where $\mathcal{S}_{ik}^t$ denotes the set of sampled items used for optimizing $q(u_{ik}^t)$.

The **approximations** of mean and variance of the variational Gaussian distribution for u_{ik}^t in Eq. (12) are as follows:

$$\phi(\tilde{u}_{ik}^t) = [\frac{1}{2}\sum_{v_j\in\mathcal{E}_i^t}\tilde{v}_{jk}^t\big(w_{ij}^t + 1 - 4(w_{ij}^t-1)\lambda(\xi_{ij}^t)\tilde{\tau}_{ijk}^t\big) - \frac{N}{2|\mathcal{S}_{ik}^t|}\sum_{v_j\in\mathcal{S}_{ik}^t}\big(\tilde{v}_{jk}^t + 4\lambda(\xi_{ij}^t)\tilde{v}_{jk}^t\tilde{\tau}_{ijk}^t\big) + \tilde{u}_{ik}^{t-1}\alpha_k\big]\phi(\Sigma_{ik}^t),$$

Algorithm 1: Variational Inference for DBLMF

Input : $\{\mathcal{O}^t\}_{t=1}^T$, d, σ_0, a_0, b_0, a_1, b_1
Output: $\{q^*(\boldsymbol{U}^t)\}_{t=1}^T$, $\{q^*(\boldsymbol{V}^t)\}_{t=1}^T$

```
Set α_k = 0, β_k = 0, ∀1 ≤ k ≤ d;
for t = 1, 2, ..., T do
    Initialize ũ_ik^t, ṽ_jk^t using N(0, 1/d), and set Σ_ik^t = 1,
      Ω_jk^t = 1, ∀1 ≤ k ≤ d, 1 ≤ i ≤ M, 1 ≤ j ≤ N;
    for iter = 1, 2, ..., max_iter do
        Compute auxiliary variables ξ_ij^t, 1 ≤ i ≤ M,
          1 ≤ j ≤ N using Eq. (17);
        for i = 1, 2, ..., M do
            for k = 1, 2, ..., d do
                Update ũ_ik^t and Σ_ik^t using Eq. (12);
        for j = 1, 2, ..., N do
            for k = 1, 2, ..., d do
                Update ṽ_jk^t and Ω_jk^t using Eq. (13);
        Update the Gamma distributions for α and β using
          Eq. (14) and Eq. (15), resepectively;
```

$$\phi(\Sigma_{ik}^t) = \big[\alpha_k + \sigma_0 + 2\sum_{v_j\in\mathcal{E}_i^t}(w_{ij}^t-1)\lambda(\xi_{ij}^t)((\tilde{v}_{jk}^t)^2 + \Omega_{jk}^t) + \frac{2N}{|\mathcal{S}_{ik}^t|}\sum_{v_j\in\mathcal{S}_{ik}^t}\lambda(\xi_{ij}^t)((\tilde{v}_{jk}^t)^2 + \Omega_{jk}^t)\big]^{-1}. \quad (19)$$

Similarly, the **approximations** of mean and variance of the variational Gaussian distribution for v_{jk}^t are as follows:

$$\phi(\tilde{v}_{jk}^t) = [\frac{1}{2}\sum_{u_i\in\mathcal{F}_j^t}\tilde{u}_{ik}^t\big(w_{ij}^t + 1 - 4(w_{ij}^t-1)\lambda(\xi_{ij}^t)\tilde{\tau}_{ijk}^t\big) - \frac{M}{2|\mathcal{T}_{jk}^t|}\sum_{u_i\in\mathcal{T}_{jk}^t}\big(\tilde{u}_{ik}^t + 4\lambda(\xi_{ij}^t)\tilde{u}_{ik}^t\tilde{\tau}_{ijk}^t\big) + \tilde{v}_{jk}^{t-1}\alpha_k\big]\phi(\Omega_{jk}^t),$$

$$\phi(\Omega_{jk}^t) = \big[\beta_k + \sigma_0 + 2\sum_{u_i\in\mathcal{F}_j^t}(w_{ij}^t-1)\lambda(\xi_{ij}^t)((\tilde{u}_{ik}^t)^2 + \Sigma_{ik}^t) + \frac{2M}{|\mathcal{T}_{jk}^t|}\sum_{u_i\in\mathcal{T}_{jk}^t}\lambda(\xi_{ij}^t)((\tilde{u}_{ik}^t)^2 + \Sigma_{ik}^t)\big]^{-1}, \quad (20)$$

where $\mathcal{T}_{jk}^t$ denotes the set of sampled users used for optimizing $q(v_{jk}^t)$. Considering model stability, we set $\mathcal{S}_{ik}^t = \mathcal{S}_i^t$, $\mathcal{T}_{jk}^t = \mathcal{T}_j^t$, $1 \le k \le d$. The final updating rules for the variational distributions of elements in $\boldsymbol{U}^t$ and $\boldsymbol{V}^t$ are as follows:

$$\theta \leftarrow (1-\gamma)\theta + \gamma\phi(\theta), \quad (21)$$

where θ is the mean or variance of a variational Gaussian distribution, $\phi(\theta)$ is the approximation based on uniform sampling, and $\gamma \in (0, 1]$ is a constant which is empirically set at 0.95. Algorithm 2 summarizes the details of the inference algorithm based on uniform sampling. In the experiments, we empirically set $|\mathcal{S}_i^t| = |\mathcal{E}_i^t|$ and $|\mathcal{T}_j^t| = |\mathcal{F}_j^t|$. Then, the time complexity of each iteration in Algorithm 2 is $\mathcal{O}(|\mathcal{O}^t| \cdot d)$.

Algorithm 2: Variational Inference for DBLMF by Uniform Sampling

Input : $\{\mathcal{O}^t\}_{t=1}^T$, d, γ, σ_0, a_0, b_0, a_1, b_1
Output: $\{q^*(\boldsymbol{U}^t)\}_{t=1}^T$, $\{q^*(\boldsymbol{V}^t)\}_{t=1}^T$

Set $\alpha_k = 0$, $\beta_k = 0$, $\forall 1 \le k \le d$;
for $t = 1, 2, \ldots, T$ **do**
  Initialize $\tilde{u}_{ik}^t$, $\tilde{v}_{jk}^t$ using $\mathcal{N}(0, 1/d)$, set $\Sigma_{ik}^t = 1$, $\Omega_{jk}^t = 1, \forall 1 \le k \le d, 1 \le i \le M, 1 \le j \le N$;
  for $iter = 1, 2, \cdots, max\_iter$ **do**
    for $i = 1, 2, \ldots, M$ **do**
      Uniformly sampled items to construct $\mathcal{S}_i^t$;
      Compute $\xi_{ij}^t, \forall v_j \in \mathcal{E}_i^t \cup \mathcal{S}_i^t$ using Eq. (17);
      for $k = 1, 2, \ldots, d$ **do**
        Compute $\phi(\tilde{u}_{ik}^t)$, $\phi(\Sigma_{ik}^t)$ using Eq. (19);
        $\tilde{u}_{ik}^t \leftarrow (1-\gamma)\tilde{u}_{ik}^t + \gamma\phi(\tilde{u}_{ik}^t)$;
        $\Sigma_{ik}^t \leftarrow (1-\gamma)\Sigma_{ik}^t + \gamma\phi(\Sigma_{ik}^t)$;
    for $j = 1, 2, \ldots, N$ **do**
      Uniformly sampled users to construct $\mathcal{T}_j^t$;
      Compute $\xi_{ij}^t, \forall u_i \in \mathcal{F}_j^t \cup \mathcal{T}_j^t$ using Eq. (17);
      for $k = 1, 2, \ldots, d$ **do**
        Compute $\phi(\tilde{v}_{jk}^t)$, $\phi(\Omega_{jk}^t)$ using Eq. (20);
        $\tilde{v}_{jk}^t \leftarrow (1-\gamma)\tilde{v}_{jk}^t + \gamma\phi(\tilde{v}_{jk}^t)$;
        $\Omega_{jk}^t \leftarrow (1-\gamma)\Omega_{jk}^t + \gamma\phi(\Omega_{jk}^t)$;
    Update the Gamma distributions for $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ using Eq. (14) and Eq. (15), resepectively;

4 Experiments

4.1 Experimental Setting

Datasets and Setups: We extracted four datasets from Movielens-20M[1] for evaluation. Following [Pan and Chen, 2013; Liu *et al.*, 2015], we kept ratings larger than 3 as implicit feedback. Dataset 2005-3Y was extracted using the following conditions: 1) only ratings between 2005-01-01 and 2010-12-31 were included; 2) ratings before and on 2007-12-31 were used for training, and ratings after 2007-12-31 were used for testing; and 3) only users with at least 5 ratings in training data and at least one rating in testing data were included. Dataset 2002-6Y was generated in a similar way by using the rating data between 2002-01-01 and 2007-12-31 as training data, and the rating data between 2008-01-01 to 2010-12-31 as testing data. As the proposed algorithms focus on modeling users' preferences over time, it is better to have users' activities over a long period. To this end, we further restrict that users must have at least one rating before 2005-12-31 on dataset 2005-3Y, and at least one rating before 2002-12-31 on dataset 2002-6Y. This leads to two more datasets: 2005-3Y-S and 2002-6Y-S. Table 1 summarizes the statistics of these datasets.

The recommendation accuracies of a model are measured by: Precision@K, Recall@K, and NDCG@K, which have been widely applied to evaluate the performances of recommendation algorithms based on implicit feedback [Shi *et al.*, 2014]. In the experiments, we set K=5 and 10. For each metric, we first compute the accuracy for each user, and then report the averaged accuracy over all users.

[1]https://grouplens.org/datasets/movielens/

Dataset	# users	# items	# train pairs	# test pairs
2005-3Y	4,579	9,039	964,918	276,218
2002-6Y	4,737	9,164	1,451,515	285,290
2005-3Y-S	1,900	8,723	387,609	98,350
2002-6Y-S	1,046	8,918	410,843	49,872

Table 1: Statistics of the experimental datasets.

Evaluated Recommendation Models: We compare the following recommendation methods: (1) ***BPR*** [Rendle *et al.*, 2009]: This is a pairwise learning-to-rank method developed for recommendation with implicit feedback. (2) ***LMF*** [Johnson, 2014]: This is the logistic matrix factorization method that employs MAP (i.e., maximum a posterior) to learn user and item embeddings. (3) ***eALS*** [He *et al.*, 2016]: This matrix factorization method weights the missing data based on item popularity and uses element-wise alternating least squares technique to learn latent embeddings. (4) ***dPF*** [Charlin *et al.*, 2015]: This method uses Poisson matrix factorization to model users' implicit feedback and utilizes Kalman filter to learn time evolving embeddings. (5) ***DBLMF***: This is the proposed method that uses Algorithm 1 to learn dynamic embeddings. (6) ***sDBLMF***: This is the proposed method that uses Algorithm 2 to learn dynamic embeddings.

For matrix factorization methods, we set the dimensionality of the latent space d at 32. In DBLMF and sDBLMF, we set $\sigma_0 = 10$, $a_0 = a_1 = 10^{-4}$, and $b_0 = b_1 = 10^{-4}$. Moreover, we define the dynamic weight as $w_{ij}^t = 1 + \delta r_{ij}^t$, where r_{ij}^t denotes the rating that u_i assigned to v_j at time t, and we empirically set $\delta = 1$ in the experiments.

4.2 Experimental Results

We have tested two different settings for DBLMF and sDBLMF. The first setting is to have only one time period over the whole dataset (i.e., DBLMF-full and sDBLMF-full). The second setting is to set the time granularity to be 3 months for the proposed methods (i.e., DBLMF-3m and sDBLMF-3m) and dPF. Table 2 summarizes the performances on different datasets. We make the following observations:

- Without considering temporal dynamics, the proposed methods (i.e., DBLMF-full and sDBLMF-full) achieve comparable or even better results than existing recommendation algorithms (i.e., BPR, LMF, and eALS) that are based on implicit feedback data, in terms of all metrics. Moreover, DBLMF-full usually outperforms LMF. This indicates that Bayesian inference can more accurately estimate latent embeddings than MAP by considering higher statistics such as covariance.
- The dynamic matrix factorization methods (i.e., dPF, DBLMF-3m, and sDBLMF-3m) usually achieve better results than BPR, LMF, and eALS, which do not consider temporal information. For instance, on 2002-6Y-S dataset, DBLMF-3m and sDBLMF-3m outperform eALS by 139.16% and 145.07% respectively, in terms of NDCG@5. This indicates that the recommendation accuracy can be significantly improved by considering temporal dynamics. Additionally, the accuracy improvements obtained by dynamic matrix factorization methods

Datasets	Metrics	BPR	LMF	eALS	DBLMF-full	sDBLMF-full	dPF	DBLMF-3m	sDBLMF-3m
2005-3Y	Precision@5	0.1821	0.2136	0.2126	0.2221	0.2190	**0.2308**	0.2140	0.2134
	Precision@10	0.1684	0.1989	0.1921	0.2022	0.1996	**0.2081**	0.2026	0.2027
	Recall@5	0.0226	0.0277	0.0279	0.0299	0.0280	0.0286	**0.0318**	0.0315
	Recall@10	0.0404	0.0493	0.0477	0.0490	0.0485	0.0518	**0.0624**	0.0612
	NDCG@5	0.1866	0.2222	0.2224	0.2320	0.2275	**0.2350**	0.2264	0.2253
	NDCG@10	0.1782	0.2124	0.2080	0.2178	0.2145	**0.2201**	0.2199	0.2194
2002-6Y	Precision@5	0.1932	0.2247	0.2169	**0.2312**	0.2237	0.2308	0.2038	0.2092
	Precision@10	0.1785	0.2030	0.1989	**0.2090**	0.2029	0.2039	0.1967	0.1996
	Recall@5	0.0234	0.0284	0.0283	0.0296	0.0290	**0.0351**	0.0296	0.0308
	Recall@10	0.0421	0.0497	0.0485	0.0507	0.0490	0.0573	0.0599	**0.0607**
	NDCG@5	0.2007	0.2350	0.2264	**0.2416**	0.2343	0.2356	0.2175	0.2215
	NDCG@10	0.1905	0.2198	0.2138	**0.2259**	0.2195	0.2197	0.2125	0.2162
2005-3Y-S	Precision@5	0.1179	0.1355	0.1541	0.1493	0.1523	0.2204	**0.2539**	0.2538
	Precision@10	0.1122	0.1247	0.1404	0.1355	0.1371	0.1963	0.2252	**0.2261**
	Recall@5	0.0146	0.0200	0.0208	0.0200	0.0197	0.0348	**0.0424**	0.0422
	Recall@10	0.0273	0.0335	0.0379	0.0343	0.0346	0.0600	0.0769	**0.0771**
	NDCG@5	0.1200	0.1405	0.1609	0.1573	0.1598	0.2232	**0.2607**	**0.2607**
	NDCG@10	0.1169	0.1335	0.1514	0.1471	0.1487	0.2088	0.2458	**0.2465**
2002-6Y-S	Precision@5	0.0912	0.1057	0.1120	0.1208	0.1235	0.2298	0.2795	**0.2809**
	Precision@10	0.0858	0.0941	0.0999	0.1047	0.1067	0.1904	**0.2163**	0.2153
	Recall@5	0.0110	0.0163	0.0148	0.0166	0.0161	0.0401	0.0518	**0.0526**
	Recall@10	0.0200	0.0267	0.0259	0.0292	0.0274	0.0656	0.0802	**0.0808**
	NDCG@5	0.0955	0.1092	0.1167	0.1262	0.1310	0.2354	0.2791	**0.2860**
	NDCG@10	0.0914	0.1017	0.1082	0.1151	0.1184	0.2115	0.2425	**0.2459**

Table 2: Performances of different recommendation algorithms. The best results are in **bold faces** and the second best results are underlined.

on 2005-3Y-S and 2002-6Y-S datasets are much more significant than that obtained on other two datasets. The potential reason is that users' activities on 2005-3Y-S and 2002-6Y-S datasets are over a longer period. Thus, these two datasets contain more sufficient temporal information to learn users' dynamic preferences.

- Compared with dPF, DBLMF-3m and sDBLMF-3m achieve better performances on 2005-3Y-S and 2002-6Y-S datasets. For example, in terms of NDCG@5, DBLMF-3m and sDBLMF-3m outperform dPF by 18.56% and 21.50%, on 2002-6Y-S dataset. This is because dPF assumes different dimensions of user/item embeddings are homogeneously evolving with the same diffusion rate. However, the proposed DBLMF methods enable different dimensions of latent embeddings to heterogeneously evolve with inconsistent diffusion rates, so that better performances can be achieved.
- On all datasets, sDBLMF methods (i.e., sDBLMF-full and sDBLMF-3m) achieve comparable results with DBLMF methods (i.e., DBLMF-full and DBLMF-3m). However, sDBLMF methods are tens of times faster than DBLMF methods. This demonstrate the effectiveness of the sampling inference algorithm.

In addition, we have also evaluated the sensitivity to parameters for sDBLMF, because it usually achieves comparable results with DBLMF and is scalable to large datasets. The time granularity is varied in $\{1, 3, 6, 9, 12, 18, 24, 36, 72\}$ months. The dimensionality of latent space d is adjusted in $\{8, 16, 32, 64, 128\}$. The parameter of prior distributions σ_0 is varied in $\{1, 10, 50, 100, 150, 200, 250\}$. Moreover, we also adjust the number of iteration times from 1 to 200. The results on 2002-6Y-S dataset are summarized in Figure 1. Observed that smaller time granularity generally achieves better results. When the time granularity is set to 3 months, sDBLMF achieves the best results. We also notice that sDBLMF is relatively stable with respect to different settings of d and σ_0. The recommendation accuracy generally increases with the increase of the number of iterations, particularly when the number of iterations is fewer than 10. When the number of iteration times is larger than 10, further increases of iteration times does not improve the performance much. Due to space limitation, we do not report the performance trends regarding with other model parameters in this paper.

5 Conclusions

In this paper, we propose a novel recommendation method, namely Dynamic Bayesian Logistic Matrix Factorization (DBLMF). The proposed model is capable of inferring the dynamic embeddings of users and items, and adapting the diffusion rates of latent embeddings in a data-driven manner. Remarkably, the proposed Bayesian inference framework can obtain the advantages of model flexibility and statistical information incorporation. In addition, a scalable sampling based inference approach has also been developed, which is particularly useful in presence of large datasets. The superiority of the proposed approaches has been validated by comparing with the state-of-the-art baselines on Movielens-20M dataset. As for the future work, we intend to develop different sampling strategies to improve the recommendation accuracy.

References

[Blei *et al.*, 2017] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 2017.

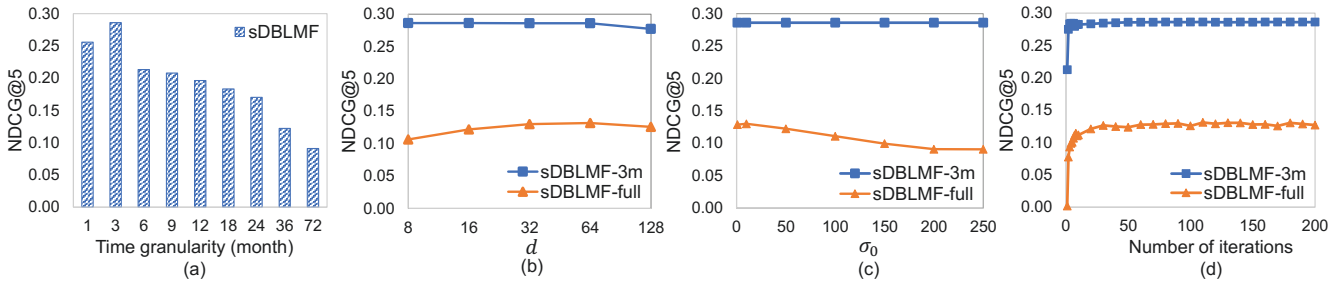

Figure 1: Performance trend of sDBLMF on dataset 2002-6Y-S measured by NDCG@5 with different parameter settings.

[Charlin *et al.*, 2015] Laurent Charlin, Rajesh Ranganath, James McInerney, and David M Blei. Dynamic poisson factorization. In *RecSyS'15*, pages 155–162, 2015.

[Du *et al.*, 2015] Nan Du, Yichen Wang, Niao He, Jimeng Sun, and Le Song. Time-sensitive recommendation from recurrent user activities. In *NIPS'15*, pages 3492–3500, 2015.

[Gultekin and Paisley, 2014] San Gultekin and John Paisley. A collaborative kalman filter for time-evolving dyadic processes. In *ICDM'14*, pages 140–149, 2014.

[He *et al.*, 2016] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. Fast matrix factorization for online recommendation with implicit feedback. In *SIGIR'16*, pages 549–558, 2016.

[Hosseini *et al.*, 2017] Seyed Abbas Hosseini, Keivan Alizadeh, Ali Khodadadi, Ali Arabzadeh, Mehrdad Farajtabar, Hongyuan Zha, and Hamid R Rabiee. Recurrent poisson factorization for temporal recommendation. In *KDD'17*, pages 847–855, 2017.

[Hu *et al.*, 2014] Longke Hu, Aixin Sun, and Yong Liu. Your neighbors affect your ratings: on geographical neighborhood influence to rating prediction. In *SIGIR'14*, pages 345–354, 2014.

[Jaakkola and Jordan, 1997] T Jaakkola and M Jordan. A variational approach to bayesian logistic regression models and their extensions. In *Sixth International Workshop on Artificial Intelligence and Statistics*, 1997.

[Johnson, 2014] Christopher C Johnson. Logistic matrix factorization for implicit feedback data. In *NIPS 2014 Workshop on Distributed Machine Learning and Matrix Computations*, 2014.

[Koren, 2009] Yehuda Koren. Collaborative filtering with temporal dynamics. In *KDD'09*, pages 89–97, 2009.

[Lian *et al.*, 2016] Defu Lian, Zhenyu Zhang, Yong Ge, Fuzheng Zhang, Nicholas Jing Yuan, and Xing Xie. Regularized content-aware tensor factorization meets temporal-aware location recommendation. In *ICDM'16*, pages 1029–1034, 2016.

[Liu *et al.*, 2010] Nathan N Liu, Min Zhao, Evan Xiang, and Qiang Yang. Online evolutionary collaborative filtering. In *RecSyS'10*, pages 95–102, 2010.

[Liu *et al.*, 2014] Yong Liu, Wei Wei, Aixin Sun, and Chunyan Miao. Exploiting geographical neighborhood characteristics for location recommendation. In *CIKM'14*, pages 739–748, 2014.

[Liu *et al.*, 2015] Yong Liu, Peilin Zhao, Aixin Sun, and Chunyan Miao. A boosting algorithm for item recommendation with implicit feedback. In *IJCAI'15*, pages 1792–1798, 2015.

[Liu *et al.*, 2016] Yong Liu, Min Wu, Chunyan Miao, Peilin Zhao, and Xiao-Li Li. Neighborhood regularized logistic matrix factorization for drug-target interaction prediction. *PLoS Computational Biology*, 12(2):e1004760, 2016.

[Liu *et al.*, 2017] Yong Liu, Peilin Zhao, Xin Liu, Min Wu, and Xiao-Li Li. Learning user dependencies for recommendation. In *IJCAI'17*, pages 2379–2385, 2017.

[Liu, 2015] Xin Liu. Modeling users' dynamic preference for personalized recommendation. In *IJCAI'15*, pages 1785–1791, 2015.

[Pan and Chen, 2013] Weike Pan and Li Chen. Gbpr: Group preference based bayesian personalized ranking for one-class collaborative filtering. In *IJCAI'13*, pages 2691–2697, 2013.

[Rendle *et al.*, 2009] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI'09*, pages 452–461, 2009.

[Shi *et al.*, 2014] Yue Shi, Martha Larson, and Alan Hanjalic. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys*, 47(1):3, 2014.

[Wang *et al.*, 2016] Yichen Wang, Nan Du, Rakshit Trivedi, and Le Song. Coevolutionary latent feature processes for continuous-time user-item interactions. In *NIPS'16*, pages 4547–4555, 2016.

[Wang *et al.*, 2017] Yixin Wang, Alp Kucukelbir, and David M Blei. Robust probabilistic modeling with bayesian data reweighting. In *ICML'17*, pages 3646–3655, 2017.

[Xiong *et al.*, 2010] Liang Xiong, Xi Chen, Tzu-Kuo Huang, Jeff Schneider, and Jaime G Carbonell. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *SDM'10*, pages 211–222, 2010.

[Yu *et al.*, 2016] Hsiang-Fu Yu, Nikhil Rao, and Inderjit S Dhillon. Temporal regularized matrix factorization for high-dimensional time series prediction. In *NIPS'16*, pages 847–855, 2016.