This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2840993, IEEE Transactions on Knowledge and Data Engineering

Correlated Matrix Factorization for Recommendation with Implicit Feedback

Yuan He, Cheng Wang, *Senior Member, IEEE*, Changjun Jiang

Abstract—As a typical latent factor model, Matrix Factorization (MF) has demonstrated its great effectiveness in recommender systems. Users and items are represented in a shared low-dimensional space so that the user preference can be modeled by linearly combining the item factor vector V using the user-specific coefficients U. From a generative model perspective, U and V are drawn from two *independent* Gaussian distributions, which is not so faithful to the reality. Items are produced to maximally meet users' requirements, which makes U and V strongly correlated. Meanwhile, the linear combination between U and V forces a bijection (one-to-one mapping), which thereby neglects the mutual correlation between the latent factors. In this paper, we address the upper drawbacks, and propose a new model, named Correlated Matrix Factorization (CMF). Technically, we apply Canonical Correlation Analysis (CCA) to map U and V into a new semantic space. Besides achieving the optimal fitting on the rating matrix, one component in each vector (U or V) is also tightly correlated with every single component in the other. We derive efficient inference and learning algorithms based on variational EM methods. The effectiveness of our proposed model is comprehensively verified on four public datasets. Experimental results show that our approach achieves competitive performance on both prediction accuracy and efficiency compared with the current state of the art.

Index Terms—Probabilistic Graphical Model, Recommender systems, Matrix Factorization, Canonical Correlation Analysis

1 Introduction

The prevalence of e-commerce has strongly propelled the popularity of recommender systems. Practice has proven that robust and accurate recommendations would increase both satisfaction for users and revenue for item providers. Previous work has focused on two different kinds of inputs for recommender systems. The most convenient is the high quality *explicit feedback*, where users' ratings directly reflect their preferences on items. In most cases, negative and positive attitudes distribute uniformly in the whole dataset, which provides comprehensive profiles for the items. For example, users in Netflix give explicit star ratings to movies to indicate their personal preferences. However, explicit ratings are always difficult to obtain or even not available in many applications. More often, users interact with items through *implicit feedback*, which contains more diverse types, such as the purchase history, browsing history or even mouse movements. In other words, implicit data is a natural byproduct of users' behavior, which makes it more abundant and also enables new innovations in recommendation. But different from explicit feedback, users avoid to interact with items they do not like [1], which leads to the natural scarcity of negative data in implicit feedback (also known as the one-class problem [2]). Only modeling the observable positive data would result in biased representations of users' preferences. Broadly speaking, implicit feedback provides better expressiveness than explicit feedback, but it's also more challenging to be well utilized.

Among the various methods, matrix factorization is the most popular and effective algorithm, in which a user rating is modeled by linearly combining the item factor vector V using user-specific coefficients U [3]. The idea behind such models is that the preference of a user is determined by a relatively small number of unobservable factors. Early work on MF-based algorithms formulates recommendation as a rating prediction problem. Since most real-world datasets are extremely sparse, directly solving the matrix factorization is computationally intractable. To speed up the calculation, researchers apply the low-rank approximations based on minimizing the sum-square distance using Singular Value Decomposition (SVD). In other words, such algorithms dismiss the large volume of missing data, and learn the model parameters only based on the non-zero values with the stochastic gradient descent method. On account of its great effectiveness and efficiency, tremendous amount of work has been devised, such as NSVD [4] and SVD++ [5]. To overcome the one-class problem, researchers [6], [7], [8], [9] propose to treat the missing data as negative feedback according to the items' population. The basic hypothesis of such work is that popular items have higher probability to be exposed to users, which makes the missing of ratings more probable to come from deliberate choices. Instead of learning the optimal fitting on the training data, Tikhonov regularization [10] (or called ridge regression) is usually added to the optimization task to avoid overfitting. Inspired by this idea, many studies [11], [12], [13] try to incorporate external knowledge, such as social ties, geographical information and user trust into MF by adding different regularizations.

Due to the remarkable improvement brought by the regularization terms, researchers want to know whether there is any theoretical interpretation for them. Salakhutdinov et al.

- *Y. He, C. Wang and C. Jiang are with the Department of Computer Science and Engineering, Tongji University, the Key Laboratory of Embedded System and Service Computing, Ministry of Education, No. 4800, Caoan Road, Shanghai 201804, China, and the Shanghai Electronic Transactions and Information Service Collaborative Innovation Center.*
E-mail: yaronhe@outlook.com, {chengwang, cjjiang}@tongji.edu.cn.

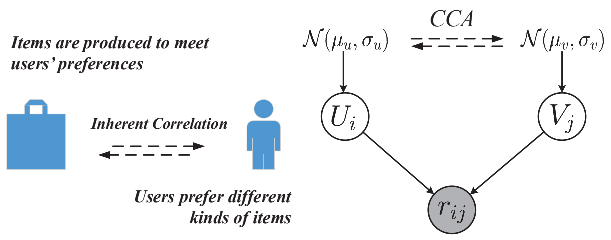

Items are produced to meet users' preferences
Inherent Correlation
Users prefer different kinds of items
$\mathcal{N}(\mu_u, \sigma_u)$
CCA
$\mathcal{N}(\mu_v, \sigma_v)$
U_i
V_j
r_{ij}

Fig. 1. The prototype of our proposed model

[3] propose that the objective function for MF is equivalent to maximizing the log-posterior of a probabilistic generative model, named probabilistic matrix factorization (PMF). The regularization terms actually come from the two Gaussian priors for U and V, which implicitly indicates that the priors are important factors for accurate recommendation. Wallach et al. [14] analyze another very famous graphical model for recommendation, named Latent Dirichlet Allocation (LDA) [15]. Different from PMF, LDA introduces two Dirichlet priors in the generative process. They have demonstrated that the prior structure substantially increases the robustness of the model. Moreover, asymmetric Dirichlet priors would result in significantly better performance than symmetric ones. Blei et al. [16] further propose the Correlated Topic model, which simply replaces the Dirichlet distribution of LDA with Gaussian prior. CTM aims to capture the mutual correlation between the latent factors, and it reports obvious improvement over LDA. In a nutshell, appropriate and informative priors in probabilistic graphical models can significantly benefit the final recommendation performance.

However, in the generative process of PMF, we find that U and V are drawn from two independent Gaussian distributions with zero means and constant variances, which means that the priors encode very little information. Meanwhile, U and V are assumed to be conditionally independent, which is not so faithful to the reality. The vectors U and V implicitly indicate the users' preferences and the item features. Items are produced to maximally meet users' requirements, thus there exists strong correlation between them. Besides the optimal fitting on the rating matrix, we should also incorporate such valuable prior knowledge. In other words, our model aims to replace the simple $L2$-regularization with a better one so that the correlation information can be properly incorporated. However, *correlation* only is a statistical concept, thus it is impractical to directly add it to the objective function. In this paper, we propose to encode such information into the priors of U and V, which is equivalent to modifying the regularization terms. Furthermore, MF-based models assume that the ratings are drawn from a Gaussian distribution with its mean parametrized by the dot product between U and V. This forces a bijection (one-to-one mapping) between the user and the item factors, and neglects the mutual correlation between latent factors.

In this paper, we address the aforementioned drawbacks of traditional matrix factorization, and propose a pure generative model, named Correlated Matrix Factorization (CMF). We introduce Canonical Correlation Analysis (CCA) [17] to capture the prior semantic association between the user and the item factors. CCA is a well-known machine learning algorithm, which introduces a new latent factor to maximize the correlation between two random sets. In the probabilistic interpretation of CCA, variables in the two random sets are drawn from two different *normal* distributions (Section 3.2) with their means decided by the shared correlation factor. Coincidentally, U and V are also assumed to be drawn from two *normal* distributions. Thus we can naturally combine CCA and MF by regarding U and V as the two shared Gaussian distributions. In other words, U and V are drawn from two correlated Gaussian distributions, and the model infers all the parameters from data. With Canonical Correlated Analysis (CCA), the correlation between U and V is maximized along with optimization process. CCA also relaxes the constrain that the dimensions of U and V have to be the same. In reality, we can always describe the preferences of users with countable features, but assign relatively more attributes for items. Setting different dimensions for U and V would be more reasonable. With CCA, the ratings are measured as the maximized semantic correlation between U and V rather than the simple inner product, which makes the model more expressive. With these improvements, the recommendation for unseen items would be more accurate. The prototype of our model is illustrated in Figure 1.

The inefficiency is always the main reason that limits the practice of PGMs. After investigating some existing work, we find that the inefficiency mainly comes from two aspects: one is the update of parameters without analytical solutions, the other is the traversal over the whole rating matrix. The gradient descent or Newton-Raphson methods are always applied to calculate parameters without analytical solutions, but they contains inner loops, which dramatically increases the time complexity. In this paper, we apply some math skills, and derive analytical updating formulas for all parameters to approximate the true posterior distribution. Since the missing values are treated as negative data, a traversal over all data can not be avoided. But inspired by the techniques applied in weighted matrix factorization [7], we memoize some independent terms in the update equations so that only the non-zero entries need to be visited. Experiments show that our model converges very fast, and each step has acceptable computational burden.

The experimental evaluation is comprehensively conducted on four different public datasets. The main contributions of this paper are as follows:

- It proposes a novel model, named Correlated Matrix Factorization (CMF). CMF achieves outstanding recommendation performance and competitive efficiency compared to the state-of-the-art algorithms with implicit feedback.
- Canonical Correlation Analysis (CCA) is introduced to elegantly model the prior correlation between the user and the item factors (U and V). It also enables us to measure the semantic association between U and V rather than the simple dot product in traditional MF.
- We derive efficient mean-field variational EM algorithm for approximate posterior inference. Some elaborate tricks are applied to accelerate the learning phase.
- Comprehensive evaluations on four different datasets are conducted to compare the proposed model with state-of-the-art baselines.

The rest of this paper is organized as follows: Section 2 introduces the related work. Section 3 summarizes the traditional methods applied in recommender systems. Section 4 describes the details of our proposed CMF model, and gives several methods to accelerate the parameter learning. Section 5 shows the experimental results compared with other baselines. Conclusions are given in Section 6.

2 Related Work

Matrix factorization has become very popular in recommender systems on account of its outstanding effectiveness and efficiency. For both explicit and implicit feedback, MF-based models have been widely applied. However, due to the convenience of acquisition and challenge in modeling, more and more studies have put their emphasis on implicit data. Different from explicit feedback which contains comprehensive opinions of users, implicit feedback is inherently lack of negative opinions. Therefore, how to better handle missing data is an obligatory task confronted by most previous work. Two different strategies have been proposed, which are **sample based learning** [2], [8], [18] and **whole-data based learning** [7], [19], [20]. The first strategy randomly samples negative instances from the missing data, while the second one treats all missing values as negative instances. Both strategies have their pros and cons: sample-based methods are more efficient, but have risk in losing valuable information; whole-based methods retain all data, but may overwhelm valid observations. Hu et al. [7] apply a uniform weight to all missing entries in the user-item matrix. Though achieving an obvious improvement, it is not so faithful to the latent semantics of data. Differently, Rendle et al. [18] subsample the missing items at a lower rate in order to reduce their influence on the estimation. To better introduce negative feedback, He et al. [8] propose a new popularity-aware weighting strategy which assigns the missing values with different confidence according to the popularity. The basic idea of this method is that popular items have higher probability to be exposed to users, thus the non-selection is more probable to indicate dislike rather than unknown. Other than effectiveness, efficiency is another concern of many previous studies [8], [21], [22], [23]. Pilaszy et al. [21] propose a fast approximation of the Alternating Least Squares (ALS) technique [24] which is an instantiation of Coordinate Descent (CD) method. Rendle et al. [22] improve it to element-wise Alternating Least Squares (eALS) which is K times faster than ALS, where K is the number of latent factors. Though very useful in practical applications, such researches contribute very little to the models' prediction accuracy. Since negativeness is meaninglessness in reality, Lee et al. [25] propose the Non-negative Matrix Factorization (NMF) which is demonstrated to better capture the parts-based representations of data [26]. Lin et al. [27] propose the Projected Gradient Methods for NMF which presents better convergence properties than the traditional multiplicative update approach applied in [26]. To further improve the performance of MF-based models, Rendle et al. [28] introduce kernel functions to replace the dot product between factor vectors. Zhang et al. [29] similarly introduce kernels into NMF, and propose the kernel NMF (KNMF) model. A kernel function enables us to efficiently compute the correlation of data in a higher-dimensional space, and makes it possible to model the nonlinear interactions between latent factors. Besides the kernel functions, Canonical Correlation Analysis (CCA) [30] is another kind of algorithm which can effectively model the semantical correlation between two sets of random variables. Miao et al. [31] apply CCA to rank target documents according to the strength of their semantic associations with the source document. It aims to explore the corresponding relation between document pairs, such as questions and answers, disease symptoms and diagnoses. Dhillon et al. [32] leverage CCA to compute the correlation between the past and future views of the data on a large unlabeled corpus to find the common latent structure. Li et al. [33] apply CCA to capture the interdependency between two unrelated embedding vectors so that they can be integrated as a consensus one. All these studies show that CCA is powerful in capturing the semantical correlation between two vectors of variables. Ding et al. [34] propose the Non-negative matrix tri-factorization (NMTF) model which adds two orthogonality constraint in NMF and factorizes the rating matrix into three latent factors. They have demonstrated that their model is equivalent to conduct simultaneous K-means clustering of the rows and columns. Wang et al. [35] propose a similar tri-factorization (MTF) method, and they incorporate two linear transformation matrices into the matrix co-factorization framework so that the the matrix factorization of user ratings is regularized by that of social network. However, Rendle et al. [18] propose that though most methods are designed for the item prediction task of personalized ranking, none of them is directly optimized for ranking. Thus they [6], [18] present a generic optimization criterion BPR-OPT derived from the maximum posterior estimation to approximate the optimal personalized ranking. Collaborative Less-is-More Filtering (CLiMF) [36] is another ranking-based model which optimizes a smoothed version of the Reciprocal Rank [37] via a lower bound. Different from BPR, CLiMF puts more emphasis on the relevant items in the top positions of a recommendation list. Such work leverages some *push* techniques such as p-norm push [38] to get accurate rankings at the top of the list. Christakopoulou et al. [39] propose an improved ranking-based model. In their work, they introduce a family of collaborative ranking algorithms to improve accuracy at the top of the ranked list for each user while learning the ranking functions collaboratively. Volkovs et al. [40] propose to utilize the advantages of neighbor approaches, and transform the observed rating matrix into a score matrix by applying neighborhood similarity rescaling. They show that factorizing the score matrix produces more accurate user and item representations than analyzing the original rating matrix. Though very different in methods, all the aforementioned algorithms are based on the conventional matrix factorization framework.

From the Bayesian perspective, there exists another train of thought which absorbs matrix factorization as parts of their probabilistic models. The fundamental of such studies is the probabilistic interpretation of MF (also called PMF) [3]. In PMF, both the user and the item feature vectors are drawn from Gaussian priors. The inner product between them is treated as the expectation of the observable ratings, which are also drawn from a Gaussian. The brevity of PMF makes it easy to incorporate external knowledge into recommendation, such as social relationship [41], associated meta-information [42], [43] and geographical records [44], [45]. Auxiliary external knowledge [42] also enables such models to better deal with the cold-start problem. Fresh users with no ratings are recommended with taste-similar items according to their profiles built from the meta information such as the comments. Besides matrix factorization, the probabilistic factor-based methods also resort to the topic modeling techniques, such as LDA [15], PLSA [46], [47] and HDP [48]. However, such models only incorporate the interactions in the rating matrix, and are not well designed for the missing values. Meanwhile, the intractable model inference always leads to unmanageable inefficiency. In this paper, we propose a pure generative model which absorbs the advantages of MF. Experiments show that it achieves outstanding effectiveness and efficiency compared with the current state of the art.

3 Background

In this section, we introduce the conventional matrix factorization solution to recommendation, and analyze its limitations from the probabilistic point of view. We offer some insights in Canonical Correlation Analysis, and describe our strategy in combining it with matrix factorization.

3.1 Matrix Factorization

Let $\boldsymbol{R} \in \mathbb{R}^{M \times N}$ denote the user-item interaction matrix, where M and N are the numbers of users and items, respectively. We use $i \in \{1, ..., M\}$ as the index for users, and $j \in \{1, ..., N\}$ as the index for items. The rating of the user i on the item i is r_{ij}. Matrix factorization maps both users and items into a latent feature space of dimension K, and represents the user i with a latent factor vector $U_i \in \mathbb{R}^K$ and the item j with a latent feature vector $V_j \in \mathbb{R}^K$. We conduct the prediction of whether the user i will like the item j with the inner product between their latent representations

$$\hat{r}_{ij} = U_i^T V_j$$

In general, the best approximation of $\hat{\boldsymbol{R}}$ to $\boldsymbol{R}$ with respect to least-square is achieved by the singular value decomposition (SVD). The most common approach applied here is to minimize the regularized square error loss

$$\arg\min_{U,V} \sum_{i,j} (r_{ij} - U_i^T V_j)^2 + \lambda_u ||U_i||^2 + \lambda_v |V_j|||^2$$

with the gradient descent method, where λ_u and λ_v are regularization parameters. From the Bayesian perspective, the matrix factorization can be interpreted with an equivalent probabilistic model [3]. In probabilistic matrix factorization (PMF), we have the following generative process

1. For each user $i \in \{1, ..., M\}$
   - draw the user's latent vector $U_i \sim \mathcal{N}(0, \sigma_u^2 \boldsymbol{I}_K)$
2. For each item $j \in \{1, ..., N\}$
   - draw the item's latent vector $V_j \sim \mathcal{N}(0, \sigma_v^2 \boldsymbol{I}_K)$
3. For each entry (i, j) in $\boldsymbol{R}$
   - draw the rating $r_{ij} \sim \mathcal{N}(U_i^T V_j, \sigma_r)$

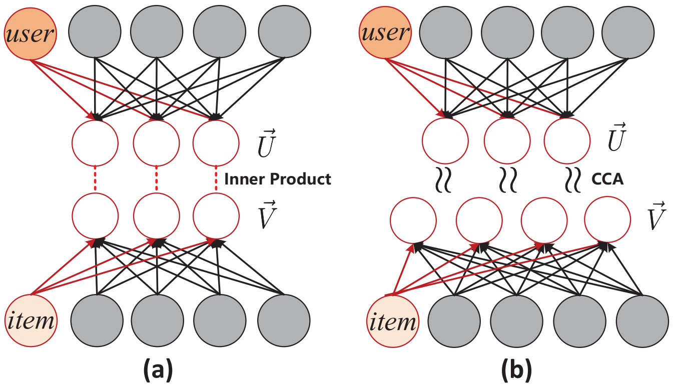


Fig. 2. Illustrations of (a) Matrix Factorization (b) Correlated Matrix Factorization

From the probabilistic interpretation, we can obtain a deeper insight into the matrix factorization model. As depicted in Figure 2(a), both users and items are mapped to the latent space with the same dimension. Meanwhile, there exists a bijection (one-to-one mapping) between the latent factors, of which the inner product is regarded as the expectation of the ratings. The red lines in Figure 2(a) identify the inherent interaction between a specific user and an item. Note that the user latent factors and the item latent features are distinguishing. The reason is that the latent user factors are drawn from a Gaussian distribution which is parameterized by σ_u which only affects the latent factor of users and has no relation with the items'. The same goes for σ_v. Thus we can decompose the probability $p(r|u, v)$ as

$$p(r|u, v) = p(U|\sigma_u)p(V|\sigma_v)p(r|U, V).$$

Since $\boldsymbol{I}_K$ is a K-dimensional identity matrix with ones on the main diagnose, the correlation between latent factors is totally neglected. Meanwhile, the inner product between U and V only allows the-same-factor interactions. From the Bayesian point of view, topic model such as LDA [15] is another train of thought for recommendation. Different from PMF, topic models introduce the latent *topics* to capture the low-dimensional features. For implicit feedback, only the non-interactions in $\boldsymbol{R}$ are regarded as valid observations, and topic models aim to maximize the likelihood that the users consume the items. For any specific interaction between u and v, the probability can be denoted as follows:

$$p(v|u) = \sum_z p(v|z)p(z|u).$$

However, the large portion of non-interactions can hardly be incorporated by topic models, otherwise the parameter estimation phase would be extremely slow. This is also an important reason why topic models do not perform as well as MF-based models in recommendation. Figure 2(b) illustrates the prototype of our model, which is an improvement over traditional matrix factorization. Instead of calculating the inner product between the user and the item latent factors, we apply CCA to capture the semantic correlation between them. CCA also allows U and V to have different dimensions. The techniques applied in MF-based models can be easily applied in our model to accelerate the learning phase so that the missing values can be efficiently taken into consideration as negative evidences.

3.2 Canonical Correlation Analysis

Canonical Correlation Analysis (CCA) is a famous machine learning algorithm, and our previous work [49] has demonstrated its great effectiveness in discovering the semantic correlation between two kinds of heterogeneous sources. However, to make the paper easy to understand, we first briefly introduce the probabilistic interpretation of CCA. Given two random sets of variables x_1 and x_2, CCA is concerned with finding a pair of linear transformations so that one component within each set of transformed variables is correlated with a single component in the other set. Meanwhile, the correlation between x_1 and x_2 is maximized in the transformed space. Bach et al. [17] provide a probabilistic interpretation of CCA, which enables the use of local CCA models as components of a larger probabilistic model.

Suppose $x_1 \in \mathbb{R}^{m_1}$ and $x_2 \in \mathbb{R}^{m_2}$, both of which depend on the latent correlation factor $y \in \mathbb{R}^L$. The generative process of CCA can be described as follows:

$$\begin{aligned}
&y \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_L), \;\; min\{m_1, m_2\} \geqslant L \geqslant 1 \\
&x_1|y \sim \mathcal{N}(T_1 y + \mu_1, \Psi_1), \;\; T_1 \in \mathbb{R}^{m_1 \times L}, \Psi_1 \succeq 0 \\
&x_2|y \sim \mathcal{N}(T_2 y + \mu_2, \Psi_2), \;\; T_2 \in \mathbb{R}^{m_2 \times L}, \Psi_2 \succeq 0
\end{aligned}$$

A very important feature that CCA possesses is that it can capture the correlation between two sets in different dimensional spaces. In our model, this feature enables us to represent users and items with different numbers of factors in exploring the rating matrix. Meanwhile, the semantical association between U and V is maximized in the latent correlation space. In other words, besides looking for the optimal fit, CCA encourages the positive correlation to be more positive, and the negative correlation to be more negative from a statistical point of view. This implicitly enhances the modeling of both positive and negative feedback in the raring matrix. Positive correlation leads to interactions, while negative correlation encourages non-interactions. Both are very important patterns for accurate recommendation. By incorporating the correlation between U and V, CCA implicitly changes the regularization terms to improve the expressive power and the generalization ability of matrix factorization.

4 Modeling Implicit Feedback

In this section, we give the details of our proposed Correlated Matrix Factorization (CMF), which is an instantiation of Probabilistic Graphical Model (PGM). The model description that follows assumes the reader is familiar with Bayesian network and statistical inference, which have already been widely used in topic modeling [15], [16] and many other machine learning fields.

4.1 Correlated Matrix Factorization

One advantage of latent factor models is that they reduce the dimension of data, and aggregate the large number of observable variables to relatively small number of underlying concepts. In other words, it recognizes the patterns underlying the data, and applies them for further prediction. Traditional matrix factorization represents users and items in a shared latent low-dimensional space with two latent vectors, U and V, drawn from two independent Gaussians. In this paper, we place U and V into two different latent spaces with different dimensions. Applying CCA as components of our model, we introduce a new latent correlation factor y to priorly couple U and V, meanwhile their correlation is maximized. In fact, y lies in a new space which captures the semantic association between U and V. r_{ij} is measured in the new semantic space, and it is denoted as the distance between the transformed U and V. With y playing as the intermediary, users and items more tightly interact with each other. To better incorporate the missing values as negative evidences, we introduce a weight variable c_{ij}, which is similar to the work of Hu et al. [7]. Specifically, c_{ij} indicates different confidence levels in observing an interaction r_{ij} between the user i and the item j, and an observable value always owns higher weight than the missing ones. The potential cause is that not taking any positive action on an item can stem from many other reasons beyond not liking it, such as being unaware of the existence of the item.

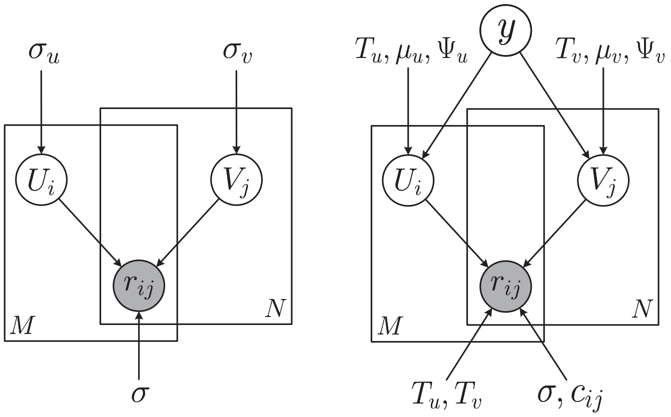


Fig. 3. The graphical model of Probabilistic Matrix Factorization (Left) and Correlated Matrix Factorization (Right)

The directed graphical model of CMF is depicted in Figure 3. Following the notations defined in Section 3, we add some new symbols in our model. Let K be the dimension of the user factor U, and T be the dimension of the item factor V. L is the dimension of the latent correlation factor y in CCA. The generative process of CMF is as follows:

1. Draw the L-dimensional Gaussian correlation factor: $y \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_L)$
2. For each user $i \in \{1, ..., M\}$
   - draw the user's latent vector: $U_i \sim \mathcal{N}(T_u y + \mu_u, \Psi_u); \quad T_u \in \mathbb{R}^{K \times L}, \Psi_u \succeq 0$
3. For each item $j \in \{1, ..., N\}$
   - draw the item's latent vector: $V_j \sim \mathcal{N}(T_v y + \mu_v, \Psi_v); \quad T_v \in \mathbb{R}^{T \times L}, \Psi_v \succeq 0$
4. For each entry (i, j) in $\boldsymbol{R}$
   - draw the rating $r_{ij} \sim \mathcal{N}^{c_{ij}}(U_i^T T_u T_v^T V_j, \sigma^2)$

$**$ The weight variable is defined as $c_{ij} = 1 + \alpha r_{ij}$ where α is a constant.

We make the following notes for the Correlated Matrix Factorization:

(1) The generative steps (step 1, 2 and 3) form the main body of CCA. The maximum likelihood estimation lead to the maximum of correlation between U and V. Meanwhile, the generative steps (step 2, 3 and 4) constitute the skeleton of matrix factorization. Thus U and V are also endowed with the patterns underlying the rating matrix. With U and V playing as the shared parts, we elegantly combine MF and CCA to a unified model.

(2) The model parameters in CMF comprise the parameter set $\Theta = \{T_u, T_v, \mu_u, \mu_v, \Psi_u, \Psi_v, U, V\}$. In our model, the parameters T_u and T_v are matrices with dimensions $K \times L$ and $T \times L$ respectively. They linearly transform U and V from their own spaces to the space of y (step 2 and 3). The random variable y forms the variable set $\Phi = \{y\}$. The observable variable is the whole rating matrix $\boldsymbol{R}$. c_{ij} can be interpreted as a weight parameter to indicate different confidence levels of observing r_{ij}. In other words, the non-interactions (i.e., $r_{ij} = 0$) are associated with lower confidence than the observable interactions. In our experiments, $\alpha_{ij} = 30$ can always generate good results.

The following equation gives the probability that $\boldsymbol{R}$ arises from the CMF model given the model parameters Θ

$$p(\boldsymbol{R}|\Theta) = \int_y p(y) \prod_{i=1}^{M} p(U_i|y) \prod_{j=1}^{N} p(V_j|y) \prod_{i,j} p(r_{ij}|U, V) dy.$$

Now our task turns to find the optimal model parameters Θ which can maximize the posterior probability given the observed ratings:

$$\arg\max_{\Theta} \log p(\boldsymbol{R}|\Theta)$$

However, it is computationally intractable since the random variable y is continuous, and tightly coupled with both U and V. Traditional EM algorithm will not work since we cannot calculate the expected value of the log likelihood function with respect to y in the E-step. In this paper, we resort to the variational EM algorithm [50] which has been widely applied by Blei et al. [15]. To achieve better efficiency, we give analytical solutions for all parameters, and apply some *memoizing* methods to accelerate the learning phase.

4.2 Variational Inference and Parameter Estimation

Though the unified CMF model is elegant in modeling implicit feedback, posterior inference is the key challenge to use it. In this paper, we make use of variational EM methods to efficiently obtain an approximation of the posterior distribution. The derivation is complicated, but it considerably reduces the computational burden when executing.

4.2.1 Variational Inference, E-step

In the E-step, we update the posterior distribution over the unobservable variable set $\Phi = \{y\}$. According to the mean-field variational method, each latent variable is assigned with a simple distribution with free parameters so that the approximation is close in the Kullback-Leibler divergence to the true posterior. In this paper, we introduce a factorized distribution $q(\Phi)$ in which the latent variables are independent of each other. Since we only have one latent variable y, we have $q(\Phi)$:

$$y \sim \mathcal{N}(\bar{y}, \Sigma)$$

where $\bar{y} \in \mathbb{R}^L$ and $\Sigma \in \mathbb{R}^{L\times L}$. Now we have a new set of variational parameters $\delta = \{\bar{y}, \Sigma\}$. Following Wainwright et al. [50], we bound the log likelihood using Jensen's inequality. That is

$$\log p(\boldsymbol{R}|\Theta) \geqslant \mathbb{E}_q(\log p(\boldsymbol{R}, \Phi|\Theta, \delta)) + H(q)$$

where $p(\boldsymbol{R}, \Phi|\Theta, \delta)$ is the log likelihood function for the complete data which contains both latent variables and observable ratings. $H(q)$ is the entropy of the variational distribution $q(\Phi)$. The lower bound of the log likelihood can be expanded as follows:

$$\begin{aligned}\log p(\boldsymbol{R}|\Theta) \geqslant \sum_{i=1}^{M} \mathbb{E}_q(\log p(U_i|y)) + \sum_{j=1}^{N} \mathbb{E}_q(\log p(V_j|y)) \\ + \sum_{i=1}^{M}\sum_{j=1}^{N} \mathbb{E}_q(\log p(r_{ij}|U_i, V_j)) + \mathbb{E}_q(\log p(y)) + H(q).\end{aligned} \tag{1}$$

We can further expand the upper equation with respect to the model parameters Θ and the variational parameters δ. Each term on the right-hand side can be expanded as follows:

$$\mathbb{E}_q(\log p(y)) = -\frac{L}{2}\log(2\pi) - \frac{1}{2}tr(\Sigma) - \frac{1}{2}\bar{y}^{\intercal}\bar{y}$$

where $tr(\Sigma)$ returns the trace of the input matrix Σ.

$$\begin{aligned}\mathbb{E}_q(\log p(U_i|y)) = -\frac{K}{2}\log(2\pi) - \frac{1}{2}\log(|\Psi_u|) \\ - \frac{1}{2}(T_u y + \mu_u - U_i)^T \Psi_u^{-1}(T_u y + \mu_u - U_i) \\ - \frac{1}{2}tr(T_u \Sigma T_u^T \Psi_u^{-1})\end{aligned}$$

where $|\Psi_u|$ denotes the determinant of matrix Ψ_u. This term is expanded by utilizing the property of matrix normal distribution $X \sim \mathcal{MN}(M, U, V)$ of which $\mathbb{E}(X^{\intercal}BX) = Vtr(UB^{\intercal}) + M^{\intercal}BM$. Similarly, we can easily expand the log likelihood for V_j with respect to the variational parameters as

$$\begin{aligned}\mathbb{E}_q(\log p(V_j|y)) = -\frac{T}{2}\log(2\pi) - \frac{1}{2}\log(|\Psi_v|) \\ - \frac{1}{2}(T_v y + \mu_v - V_j)^T \Psi_v^{-1}(T_v y + \mu_v - V_j) \\ - \frac{1}{2}tr(T_v \Sigma T_v^T \Psi_v^{-1}).\end{aligned}$$

The rating r_{ij} is drawn from a univariate normal distribution, and the variance is a global scalar. Here we have

$$\begin{aligned}\mathbb{E}_q(\log p(r_{ij}|U_i, V_j)) = -\frac{1}{2}\log(2\pi) - \frac{1}{2}\log(\sigma^2) \\ - \frac{1}{2\sigma^2}(r_{ij} - c_{ij}U_i^T T_u T_v^T V_j)^2\end{aligned}$$

The entropy of the variational distribution $q(\Phi)$ is

$$H(q) = \frac{1}{2}\log(|\Sigma|)$$

We substitute the expansions of all terms into the log likelihood function, and maximize the lower bound by taking the partial derivatives with respect to the variational parameters δ and setting them zero. The update equations for the variational parameters $\delta{=}\{\bar{y}, \Sigma\}$ are listed as follows:

$$\Sigma = (MT_u^T \Psi_u^{-1} T_u + NT_v^T \Psi_v^{-1} T_v + \boldsymbol{I}^L)^{-1} \tag{2}$$

$$\bar{y} = \Sigma' \Big(\sum_{i=0}^{M} T_u^T \Psi_u^{-1} (U_i - \mu_u) + \sum_{j=0}^{N} T_v^T \Psi_v^{-1} (V_i - \mu_v) \Big) \tag{3}$$

where Σ' denotes the updated value of Σ, or we can directly substitute the update equation of Σ into that of $\bar{y}$. We can easily find that both Σ and $\bar{y}$ can be efficiently updated with an L-dimensional matrix inversion $\mathcal{O}(L^3)$ and two independent traversals over all users and items $\mathcal{O}(M + N)$. Since L is always very small, the calculation is very fast.

4.2.2 Parameter Estimation, M-step

In this step, we fit the model by finding maximum likelihood estimates for each of the *model parameters* Θ based on the updated *variational parameters* δ. Specifically, we calculate the partial derivatives with respect to each model parameter and set them zero. The update equations are listed as follows:

$$\begin{cases} \mu_u = \dfrac{1}{M} \displaystyle\sum_{i=1}^{M} (U_i - T_u \bar{y}) \\ \Psi_u = T_u \Sigma T_u^T + \dfrac{1}{M} \displaystyle\sum_{i=1}^{M} (T_u \bar{y} + \mu_u - U_i)(T_u \bar{y} + \mu_u - U_i)^T \end{cases} \tag{4}$$

$$\begin{cases} \mu_v = \dfrac{1}{N} \displaystyle\sum_{j=1}^{N} (V_j - T_v \bar{y}) \\ \Psi_v = T_v \Sigma T_v^T + \dfrac{1}{N} \displaystyle\sum_{j=1}^{N} \Big(T_v \bar{y} + \mu_v - V_j)(T_v \bar{y} + \mu_v - V_j\Big)^T \end{cases} \tag{5}$$

The update equations for $\{\mu_u, \Psi_u\}$ and $\{\mu_v, \Psi_v\}$ have very similar form, and we only need to traverse over the users and items respectively. Since both K and T are relatively small, the matrix operations are very fast. For the user i, the update equation for U_i is

$$\begin{aligned} U_i =& (\Psi_u^{-1} + \sum_{j=1}^{N} \frac{c_{ij}}{\sigma^2} T_u T_v^T V_j V_j^T T_v T_u^T)^{-1} \Big[\Psi_u^{-1} (\mu_u + T_u \bar{y}) \\ &+ \sum_{j=1}^{N} \frac{c_{ij} r_{ij}}{\sigma^2} T_u T_v^T V_j \Big]. \end{aligned} \tag{6}$$

Clearly, the computational bottleneck lies in the summation over all data portion in the second term, which requires a traversal over the whole rating matrix. To accelerate the calculation, we substitute the expression of c_{ij}, and rewrite the second term by separating the observed data part.

$$\begin{aligned} \sum_{j=1}^{N} \frac{c_{ij}}{\sigma^2} T_u T_v^T V_j V_j^T T_v T_u^T &= \frac{1}{\sigma^2} T_u T_v^T \Big[\sum_{j=1}^{N} c_{ij} V_j V_j^T \Big] T_v T_u^T \\ &= \frac{1}{\sigma^2} T_u T_v^T \Big[\sum_{j=1}^{N} V_j V_j^T + \alpha \sum_{j=1}^{N} r_{ij} V_j V_j^T \Big] T_v T_u^T \end{aligned}$$

By this reformulation, the major computation (the $\sum_{j=1}^{N} V_j V_j^T$ term that iterates over all items) is independent of user i. A naive implementation that repeatedly computes it is unnecessary when updating the latent factor for different users. By memoizing it, we can achieve a significant speed-up. Furthermore, the terms, $T_u T_v^T V_j$ and $V_j V_j^T$, can also be pre-calculated and cached so that the iteration over the observed data only needs a simple summation.

Similarly, we can derive the update rules for V_j:

$$\begin{aligned} V_j =& (\Psi_v^{-1} + \sum_{i=1}^{M} \frac{c_{ij}}{\sigma^2} T_v T_u^T U_i U_i^T T_u T_v^T)^{-1} \Big[\Psi_v^{-1} (\mu_v + T_v \bar{y}) \\ &+ \sum_{i=1}^{M} \frac{c_{ij} r_{ij}}{\sigma^2} T_v T_u^T U_i \Big] \end{aligned} \tag{7}$$

The second term can also be reformulated as follows, and similar memoizing methods can be applied.

$$\begin{aligned} \sum_{i=1}^{M} \frac{c_{ij}}{\sigma^2} T_v T_u^T U_i U_i^T T_u T_v^T &= \frac{1}{\sigma^2} T_v T_u^T \Big[\sum_{i=1}^{M} c_{ij} U_i U_i^T \Big] T_u T_v^T \\ &= \frac{1}{\sigma^2} T_v T_u^T \Big[\sum_{i=1}^{M} U_i U_i^T + \alpha \sum_{i=1}^{M} r_{ij} U_i U_i^T \Big] T_u T_v^T \end{aligned}$$

By taking the first derivative with respect to T_u, we have

$$\begin{aligned} \triangledown T_u =& \sum_{i=1}^{M} \Big(\Psi_u^{-1} (U_i - \mu_u) \bar{y}^T - \Psi_u^{-1} \boldsymbol{T_u} (\Sigma + \bar{y}\bar{y}^T) \Big) \\ &+ \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{c_{ij}}{\sigma^2} \Big(r_{ij} U_i V_j^T T_v - U_i U_i^T \boldsymbol{T_u} T_v^T V_j V_j^T T_v \Big) \end{aligned}$$

If we set this equation zero, we have

$$\begin{aligned} & M \Psi_u^{-1} \boldsymbol{T_u} (\Sigma + \bar{y}\bar{y}^T) + \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{c_{ij}}{\sigma^2} U_i U_i^T \boldsymbol{T_u} T_v^T V_j V_j^T T_v \\ &= \sum_{i=1}^{M} \Big[\Psi_u^{-1} (U_i - \mu_u) \bar{y}^T + \sum_{j=1}^{N} \frac{c_{ij} r_{ij}}{\sigma^2} U_i V_j^T T_v \Big] \end{aligned}$$

Since T_u is wrapped by two different square matrices, finding the analytical solution becomes difficult. However, the upper equation can be generalized as a linear function to solve X:

$$\sum_{i=1}^{p} A_i X B_i = E$$

where $A_i \in \mathbb{R}^{m \times m}$ and $B_i \in \mathbb{R}^{n \times n}$. Now $X \in \mathbb{R}^{m \times n}$ is the parameter that we want to calculate. Previous work [51], [52] has devoted a lot in efficiently solving such problem. The most general method for solving this equation is based on the Kronecker product. In this approach, the equation can be rewritten as

$$\Big\{ \sum_{i=1}^{p} \big(A_i \otimes B_i^T \big) \Big\} v(X) = v(E)$$

where $v(X) = (x_1^T, x_2^T, ..., x_m^T)^T$, with x_i^T the i-th row of X. The symbol $\otimes$ denotes the Kronecker product. Now the unique solution can be easily obtained by means of the inversion of an $mn \times mn$ matrix.

In our model, we use T^u_{AB} to denote the summation $\sum_{i=1}^p A_i \otimes B_i^T$, and apply T^u_E to denote the matrix E. The details of T^u_{AB} and T^u_E can be dented as follows:

$$T^u_{AB} = M\Psi_u^{-1} \otimes (\Sigma + \bar{y}\bar{y}^T) + \sum_{i=1}^M \sum_{j=1}^N \frac{c_{ij}}{\sigma^2}(U_i U_i^T) \otimes (T_v^T V_j V_j^T T_v)$$

$$T^u_E = \sum_{i=1}^M \Big[\Psi_u^{-1}(U_i - \mu_u)\bar{y}^T + \sum_{j=1}^N \frac{c_{ij} r_{ij}}{\sigma^2} U_i V_j^T T_v\Big]$$

where $T^u_{AB} \in \mathbb{R}^{KL \times KL}$ and $T^u_E \in \mathbb{R}^{KL}$. Thus $\upsilon(T_u)$ can be analytically computed as

$$\upsilon(T_u) = (T^u_{AB})^{-1} \upsilon(T^u_E) \tag{8}$$

where T_u can be easily obtained by reshaping $\upsilon(T_u)$. The matrix inversion has the time complexity $\mathcal{O}((KL)^3)$. Since both K and L are relatively small, the calculation is efficient. However, when we compute T^u_{AB}, the computational bottleneck still exists. In the second term of the T^u_{AB} update equation, we have to iterate over all the data to obtain the summation. An inner Kronecker product makes the computation extremely time consuming. To speed up the calculation, we substitute the expression of c_{ij}, and rewrite the second term by separating the observed data part.

$$\sum_{i=1}^M \sum_{j=1}^N \frac{c_{ij}}{\sigma^2}(U_i U_i^T) \otimes (T_v^T V_j V_j^T T_v) =$$
$$\frac{1}{\sigma^2}\sum_{i=1}^M (U_i U_i^T) \otimes \Big[\sum_{j=1}^N T_v^T V_j V_j^T T_v + \alpha \sum_{j=1}^N r_{ij}(T_v^T V_j V_j^T T_v)\Big].$$

Now the major computation (the $\sum_{j=1}^N T_v^T V_j V_j^T T_v$ term that iterates over all items) is independent of user i, thus it can be pre-calculated and cached. The Kronecker product is moved from the inner loop to the outer one, and only needs to be executed M times.

Similarly, we can derive the update rules for T_v:

$$\triangledown T_v = \sum_{j=1}^N \Big(\Psi_v^{-1}(V_j - \mu_v)\bar{y}^T - \Psi_v^{-1} T_v(\Sigma + \bar{y}\bar{y}^T)\Big)$$
$$+ \sum_{i=1}^M \sum_{j=1}^N \frac{c_{ij}}{\sigma^2}\Big(r_{ij} V_j U_i^T T_u - V_i V_i^T T_v T_u^T U_i U_i^T T_u\Big)$$

The matrices T^v_{AB} and T^v_E have the same definition of T^u_{AB} and T^u_E. We only change the superscript to match it with T_v. Now we have the equations

$$T^v_{AB} = N\Psi_v^{-1} \otimes (\Sigma + \bar{y}\bar{y}^T) + \sum_{i=1}^M \sum_{j=1}^N \frac{c_{ij}}{\sigma^2}(V_j V_j^T) \otimes (T_u^T U_i U_i^T T_u)$$

$$T^v_E = \sum_{j=1}^N \Big[\Psi_v^{-1}(V_i - \mu_v)\bar{y}^T + \sum_{i=1}^M \frac{c_{ij} r_{ij}}{\sigma^2} V_j U_i^T T_u\Big]$$

where $T^v_{AB} \in \mathbb{R}^{TL \times TL}$ and $T^v_E \in \mathbb{R}^{TL}$. Thus $\upsilon(T_v)$ can be analytically computed as

$$\upsilon(T_v) = (T^v_{AB})^{-1} \upsilon(T^v_E) \tag{9}$$

Algorithm 1: Learning algorithm for CMF model

```
Input:
Rating Matrix $\boldsymbol{R}$, Model Parameters $\Theta$, Variational
Parameters $\delta$, the maximal number of iterations $n$,
and the convergence threshold $\epsilon$
Output:
The learned model parameters $\Theta$ and the learned
variational parameters $\delta$

Initialize $\Theta$ and $\delta$ with random values
for $i = 1 \rightarrow n$ do
    Update variational parameters $\delta$ with Eq. (2)-(3)
    Update model parameters $\Theta$ with Eq. (4)-(9)
    Calculate lower bound of log-likelihood with Eq. (1)
    if (Increase of the log-likelihood) $< \epsilon$ then
        break
return $\Theta$ and $v$ for further evaluations
```

The second term of T^v_{AB} can also be reformulated as follows, and similar memoizing methods can be applied.

$$\sum_{i=1}^M \sum_{j=1}^N \frac{c_{ij}}{\sigma^2}(V_j V_j^T) \otimes (T_u^T U_i U_i^T T_u) =$$
$$\frac{1}{\sigma^2}\sum_{j=1}^N (V_j V_j^T) \otimes \Big[\sum_{i=1}^M T_u^T U_i U_i^T T_u + \alpha \sum_{i=1}^M r_{ij}(T_u^T U_i U_i^T T_u)\Big]$$

As depicted in Algorithm 1, we iteratively execute the E-step and M-step until all the parameters converge.

5 Experiment

In this section, we present both quantitative and qualitative evaluations for our proposed model with some state-of-the-art baselines. We adopt the *leave-one-out* [18], [8], [53] protocol to evaluate the prediction accuracies. This means that the latest interaction of each user is held out for prediction, and the models are trained on the remaining data. As we want to solve an implicit feedback task, we remove the rating scores from the datasets, and each entry is marked as 0/1 indicating whether the user has consumed the item. For implicit feedback, predicting which item a user may consume is always more important than directly giving the rating scores. For example, we would not like to know how many times a user has viewed a item, but want to know whether he will buy it. Therefore, we apply two ranking-based metrics *Hit Ratio* (HR) and *Normalized Discounted Cumulative Gain* (NDCG). The ground-truth item is defined as the leave-out one. Since there is only one test item for each user, we truncate the ranked list at 100 to ensure a relatively large HR and NDCG. HR measures whether the ground truth item is present in the ranked list, and NDCG accounts for the position of hit. We report the scores averaged by all test interactions. Specifically, we denote *rank*(i, j) as the rank of item j in the user i's predicted list and y^{test} as the set of all items in the held-out test set.

- **HR** is computed as follows:

$$\text{HR} = \sum_{j \in y^{test}} \frac{\mathbb{1}(rank(i,j) \leqslant 100)}{|y^{test}|}$$

where $\mathbb{1}(\cdot)$ is the indicator function which returns one only when the input is true.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2840993, IEEE Transactions on Knowledge and Data Engineering

TABLE 1
Statistics of the evaluation datasets

Dataset	Review#	Item#	User#	Sparsity
Yelp	731671	25677	25915	99.89%
Flixster	1838118	11730	31606	99.50%
MovieLens	1000209	6040	3706	95.53%
Ciao	40189	1141	13226	99.73%

- **NDCG** emphasizes the importance of the top ranks by logarithmically discounting ranks, which is computed as follows:

$$\mathrm{NDCG} = \frac{1}{|y^{test}|} \sum_{j \in y^{test}} \frac{\log(\hat{r}+1)}{\log(rank(i,j)+1)}$$

  where $\hat{r}$ denotes the perfect ranking of the test item which is always set 1 in the experiments.

Previous work has investigated many different trains of thought in offering recommendation when dealing with implicit feedback. Therefore, we elaborately choose four typical state-of-the-art baselines for comparison:

- **WMF** [7]: Weighted Matrix Factorization is the conventional baseline for collaborative filtering with implicit data. It assigns different weights to the observable and missing data to indicate different confidence levels in observing them. It reports great improvements over many traditional methods such as PMF [3].
- **FastALS** [8]: FastALS is an efficient MF-based model which adopts the element-wise Alternating Least Squares (eALS) technique. The main contribution of this work is that it assigns variably weighted values to the large portion of missing data according to the items' popularity. It also demonstrates that its popularity-aware weighting strategy has the same intuition with negative sampling [6].
- **BPR** [18]: Different from MF-based models, Bayesian Personalized Ranking (BPR) directly optimizes the pairwise ranking between positive and negative observations by maximizing the posterior estimator derived from the Bayesian analysis of the recommendation problem. To achieve better efficiency, it subsamples positive-negative pairs from the rating matrix, and applies stochastic gradient descent (SGD) to learn the model parameters.
- **LDA** [15]: Latent Dirichlet Allocation (LDA) is a conventional unsupervised topic model. The latent factor (known as topic) reveals the semantic relationships between the document and the word, which can be analogically replaced by the user and the item in recommendation problems. Previous work [54] has also demonstrated its great efficiency in offering personalized recommendations.
- **SRFRM** [35]: Social Regulatory Factor Regression Model (SRFRM) incorporates two linear transformation matrices into the matrix co-factorization framework so that the matrix factorization of user ratings is regularized by that of social network. If we neglect the social information, SRFRM boils down to a Matrix tri-factorization (MTF) model which factorizes the rating matrix into three factors U, W and V. W is a linear transformation matrix which allows U and V to have difference dimensions.

5.1 Datasets and Settings

Datasets: We apply our model on four publicly available datasets: **Yelp**[1], **Flixster**[2], **MovieLens**[3] and **Ciao**[4]. These four datasets have been widely used in the evaluations of previous work [8]. The items in Yelp mainly consist of crowd-sourced reviews about local businesses. Movies and TV shows form the main components of items in the other three datasets. To achieve practical and meaningful results, we follow the common practice [8], [18], and remove users with less than 10 interactions. The statistics of the four processed datasets are shown in Table 1.

Parameter Settings: A mixture between EM and a Monte Carlo sampler is utilized to effectively learn the parameters of the LDA model. Thus we do not need to alter them. WMF, FastALS and BPR are implemented according to the details given in the original papers, and the parameters are set with the suggested strategies given by the original papers. In most cases, such settings achieve the best performance. In our model, only the three dimensional parameters K, T and L need to be predefined before the learning phase, and we always set them the same value for fair comparison with the baselines. To further investigate the influence of the latent correlation factor y, we also assign L with different values when fixing K and T. The variational EM algorithm relaxes us from tunning both the model and the variational parameters. By fitting the data, they can be learned to achieve the best performance. All models are implemented with the same programming language, and executed on the same machine for a better comparison on efficiency.

5.2 Comparison with Other Models

In this part, we investigate both the convergence speed and prediction accuracy of different models on the four datasets. The number of latent factors is fixed to 20 in consideration of both effectiveness and efficiency. Figure 4 depicts the Hit Ratio and NDCG with respect to different numbers of iterations. Each metric is averaged across all the users. We can see that the HR and NDCG have very similar variation patterns, and a higher hit ratio always indicates a larger NDCG score. Therefore, we do not distinguish the two metrics in the following analysis. We notice that CMF consistently achieves the best performance on all the datasets after convergence (the standard errors are on the order of 10^{-4}). We further conduct the one-sample paired t-test to verify that the improvement is statistically significant (p-value $<$ 0.01) for both metrics. This result indicates the improvement of our model is trustable. Meanwhile, CMF converges very fast just after a few iterations, which is much faster than LDA, BPR and FastALS. WMF obtains very competitive results, and it is just a little worse than CMF on Yelp and MovieLens. Since we set K, T, and L the same value, we believe the advantages of CMF over WMF mainly come from the correlation that we introduce to priorly couple the user and the item factors. On all the datasets, SRFRM achieves much worse results than other methods. Though SRFRM introduces a transition matrix and allows more degrees of

[1] http://www.yelp.com/dataset_challenge
[2] http://www.cs.ubc.ca/~jamalim/datasets/
[3] http://grouplens.org/datasets/movielens/
[4] http://www.librec.net/datasets.html

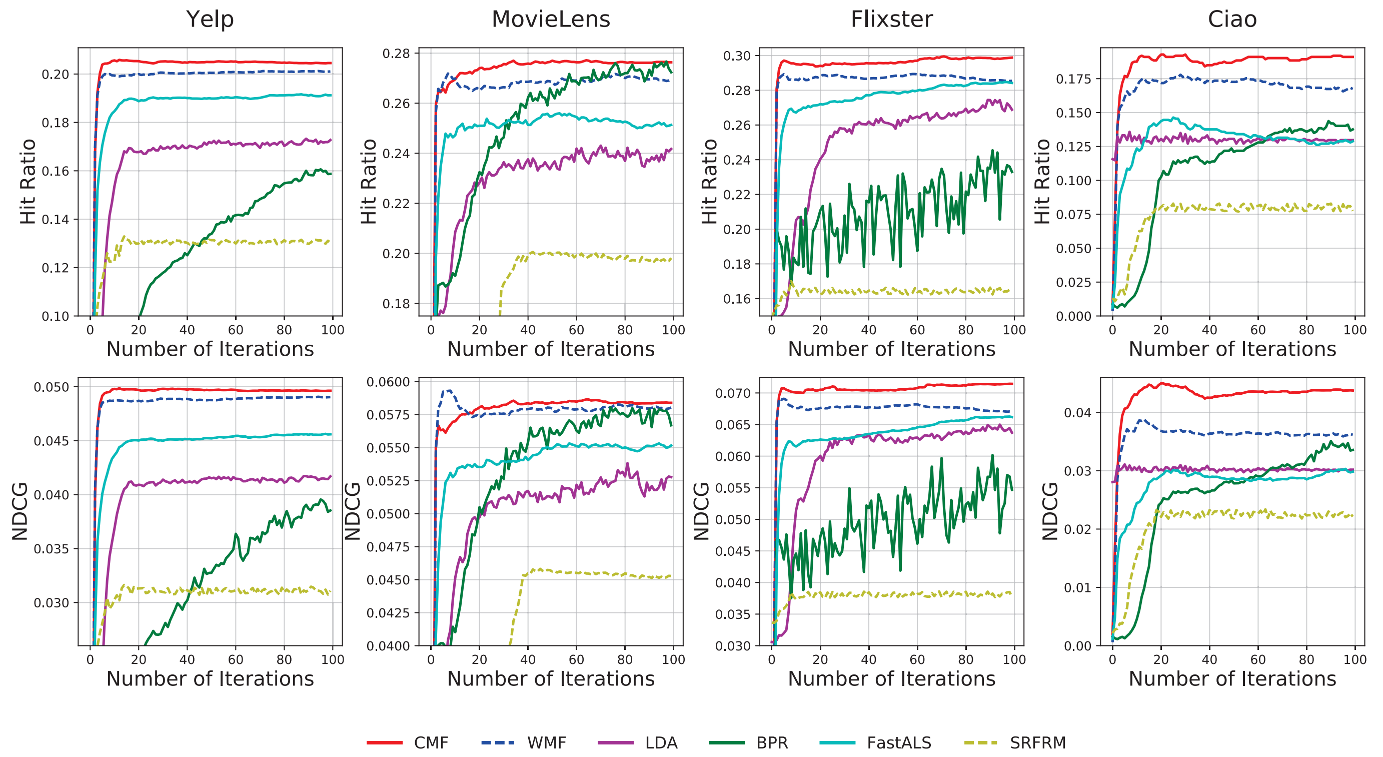


Fig. 4. Prediction accuracy of different models in each iteration (K=20)

freedom, it neglects the large portion of non-interactions. For implicit feedback, only considering the interactions would lose much valuable information underlying the data. We also find that the performance of SRFRM decreases extremely fast with less latent factors. The reason may be that SRFRM applies a logistic function to bound the value of UV to the range $(0, 1)$, which significantly decays the fitting ability of MF. Since the results of SRFRM are not in an order of magnitude with other baselines in the following experiments, we do not plot SRFRM for clarity. On Ciao dataset, the advantage of CMF in prediction becomes obvious. This may come from the small size of Ciao dataset, which offers limited information for other models to mine the underlying patterns. However, CMF measures the distance between the user and the item factors at semantic level, which is more expressive and can better fit the data. FastALS performs just a little worse than WMF on Yelp and Flixster. Technically, the only difference between FastALS and WMF is that FastALS introduces a popularity-aware weighting strategy to generate non-uniform weighting values for the missing data. This strategy is based on an intuitive assumption that the unselected popular items have higher probability to come from the deliberate choices of users, so they are more probable to be negative evidences. However, many objective factors such as the price can affect the choices of users. The strategy may be effective only on some specific datasets, but applying it on others may deteriorate the model's performance. Thus on MovieLens and Ciao, we observe a large margin between WMF and FastALS. BPR achieves very different performance on the four datasets. In most cases, it converges much slower than the other models. On MovieLens, it performs much better than FastALS and LDA, and it is also competitively compared with WMF and CMF. But on Yelp and Ciao, BPR obtains much worse results than WMF and CMF, and its performance is similar to that of LDA. On Flixster, BPR gets bad prediction accuracy which is also turbulent with the iterations. The main reason may be that BPR is a sample-based method which aims to optimize the pair-wise ranking between the positive and negative samples. One important factor that affects the performance of BPR is the proportion of valuable samples which are randomly generated. On small datasets, BPR may work

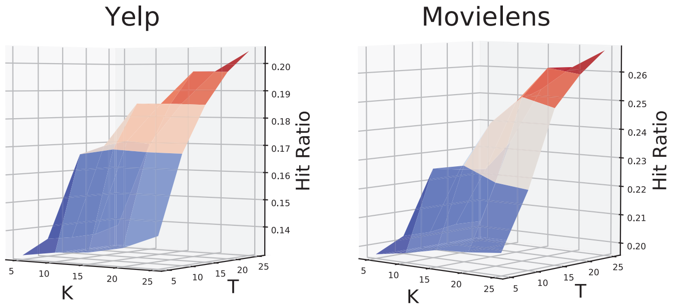


Fig. 5. Prediction accuracy with different numbers of K and T

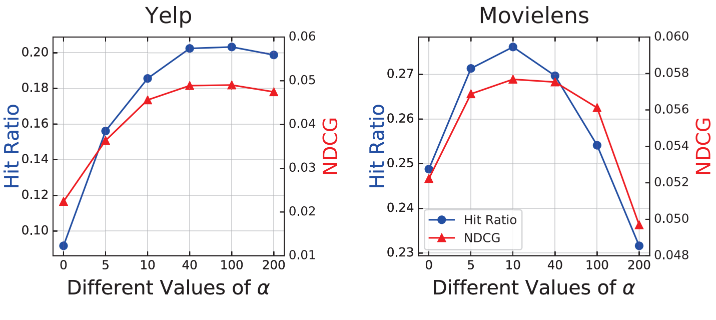


Fig. 6. Prediction accuracy with different values of α

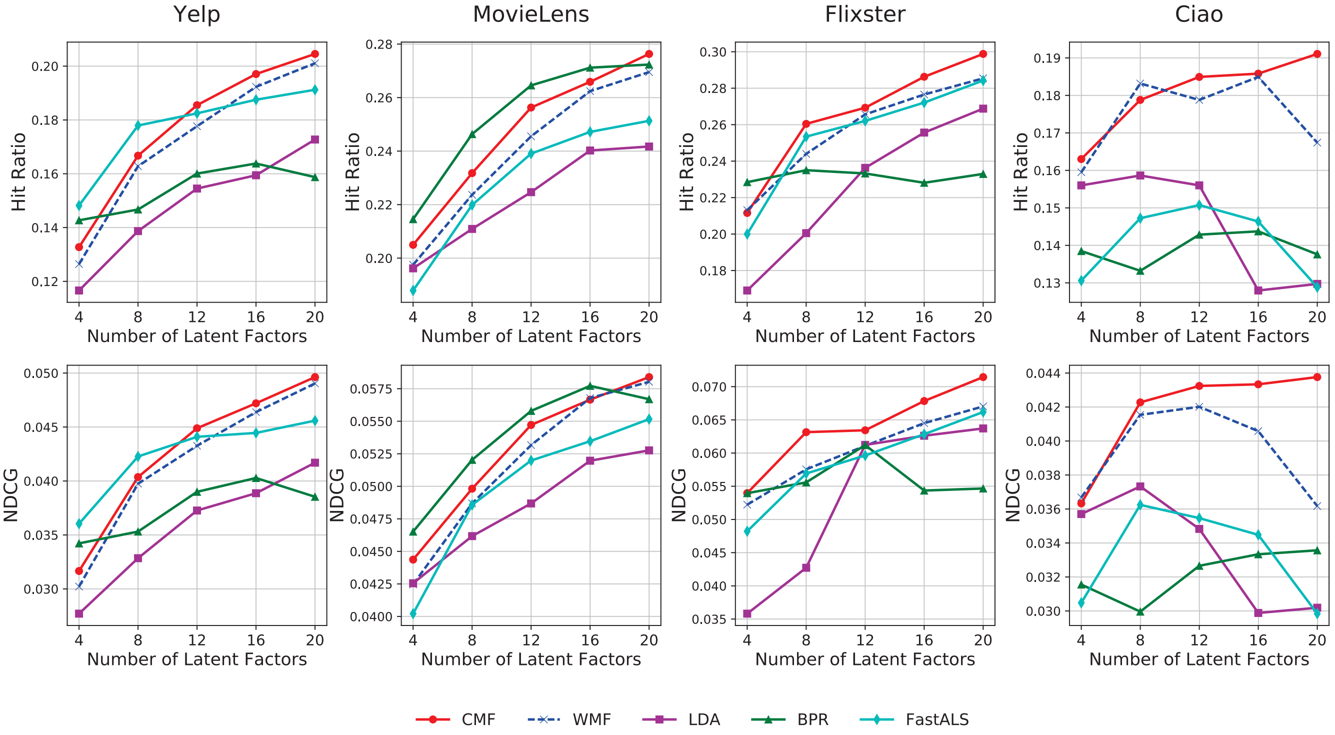


Fig. 7. Performance evaluation of each model with different numbers of latent factors

well. However, on big datasets, the pair-wise sample space becomes incredibly large, and the number of valid samples may vary in different iterations, which makes BPR less reliable than other models. LDA, as a typical probabilistic graphical model (PGM), shows satisfying results on all the datasets. However, it is still less effective than state-of-the-art MF-based models. The main reason may be that we can hardly efficiently incorporate the missing data as negative evidences into PGMs. A small modification of the model structure would lead to significant changes in the learning algorithms of LDA, and a large training set would also make the learning phase extremely slow. The cache-based techniques applied in MF-based models can not be used by PGMs to accelerate the parameter learning either.

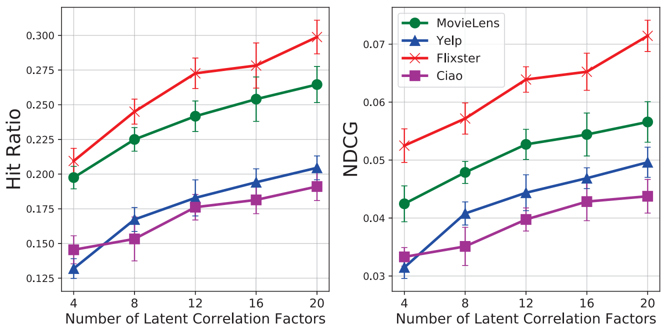


Fig. 8. Prediction accuracy of CMF with different numbers of latent correlation factors ($K = 20$ and $T = 20$)

5.3 Different Numbers of Latent Factors

In this part, we investigate the influence of different numbers of latent factors on CMF and the other four baselines. In Figure 5, we provide a 3d plot to illustrate the performance of our model with different numbers of K and T, and we set $L = min(K, T)$. We can easily find that solely increasing K or T indeed improve the recommendation performance, but the benefit is less remarkable than increasing L. Therefore, we observe obvious slope in Figure 5, and the values across the diagnose increase the fastest. Though setting U and V different dimensions does not significantly affects the prediction accuracy, it still offers the flexibility to allow more degrees of freedom. In Figure 7, the dimensional parameters K, T and L in CMF are set the same value for fair comparison with other models. Figure 7 shows the prediction accuracy with varying numbers of factors. One of the most important and interesting observations is the performance of CMF on Ciao dataset. We can find that other than CMF, all other models achieve a peak performance when the number of latent factors K is set around 8. The reason is very simple. Since the size of Ciao is relatively small, a large K might have the risk of overfitting, which would deteriorate the baselines' performance in predicting unseen data. However, CMF measures the distance of the latent factors in a semantic space, which endows it with a better expressive power. Therefore, its performance consistently increases, and presents a very different tendency with WMF. We can also see that CMF consistently outperforms other baselines on Flixster. On Yelp dataset, FastALS obtains the best performance when K is smaller than 12. However, the performance of WMF and CMF improves very rapidly with the increasing of K. When K is larger than 12, CMF gets the best results. This observation shows that the popularity-aware weighting strategy introduced in FastALS indeed has some positive impact on the model's performance, but the influence is very limited. On Flixster, BPR obtains the worst prediction accuracy with the lowest HR and NDCG when K is larger than 12. This is consistent with the observations

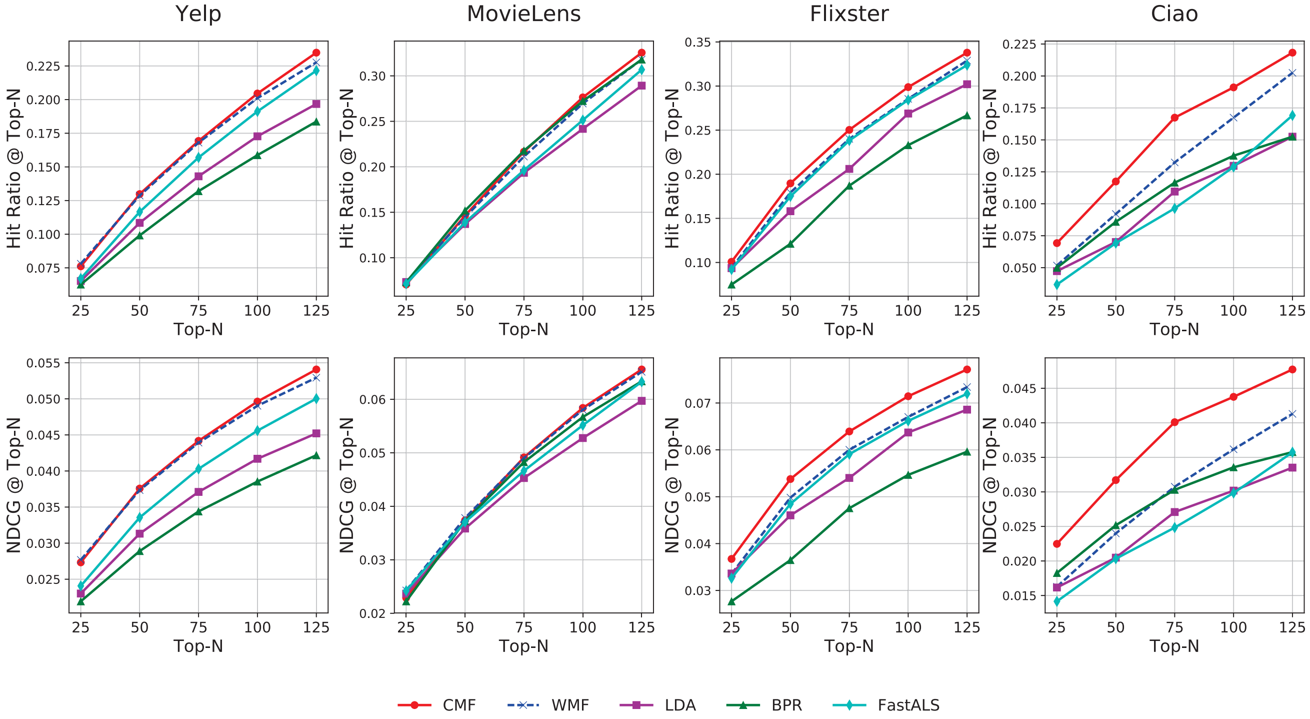


Fig. 9. Prediction accuracy of different models in the top-N ranked list

in the previous section, and the increasing of K seems to have little influence on its performance. The main reason may still come from the sampling strategy applied in BPR, and the large search space makes it hard for BPR to find the best model parameters. On MovieLens, BPR obtains the best performance when K is smaller than 16. However, BPR seems to benefit the least from the increasing of K, and CMF rapidly outperforms BPR when K reaches 20. WMF still shows its great power in recommendation, and it is just a little worse than our model in most cases. On all the four datasets, LDA obtains acceptable and reliable results. When K is small, it is competitive to the MF-based models, and just a little worse than WMF. But when K increases, the margin becomes larger. The reason may be that LDA does not distinguish the latent factors, and different latent factors may represent the same concept. Therefore, the performance of LDA will not prominently benefit from the increasing of K as MF-based models do. In Figure 6, we further investigate the influence of α on the final prediction accuracy. Due to the limitation of space, we only illustrate the results on Yelp and Movielens, which reports totally different patterns. On Yelp dataset, we find that the accuracy reaches the peak values when $\alpha = 40$, and the results keep stable with larger α. On Movielens, we find that both HR and NDCG decrease fast when α becomes larger than 10. However, setting a relatively small positive α would significant benefit the prediction accuracy on both datasets. Since α mainly relies on the dataset, there is no good strategy in choosing the best-performing values. The recommended method is conducting some heuristic approaches such as grid-search.

Figure 8 depicts the prediction accuracy of CMF with different numbers of latent correlation factors y. In this experiment, we fix $K = 20$ and $T = 20$. We can easily find that the final results of CMF increase rapidly with more latent correlation factors on all the four datasets. This observation demonstrates that the introduced latent correlation factor y really matters in modeling the semantic association between U and V. In practical applications, we can assign T a large value, which indicates that the items own a large number of attributes. K is always set a relatively small number, and the size of L must be selected in consideration of both efficiency and effectiveness. In Figure 8, we can find that the increasing speed (i.e., the gradient of the lines) of both HR and NDCG reaches its peak value between 8 and 12, thus setting $L = K/2$ may be a reasonable choice for CMF.

5.4 Top-N and Efficiency Analysis

In the previous experiments, we truncate the ranked list at 100, and apply the top-100 items to calculate the average HR and NDCG scores. In this part, we truncate the ranked list at different sizes N, and investigate the performance of each model with different Ns. Obviously, a larger N would surely lead to a better result. But a more powerful model would make its advantage greater with the increasing of N, which, in other words, indicates a denser hit at different pieces of the top-N ranked list. More specifically, when N increases, the margin between the hit radios (and NDCG) of different models should become larger if a model is inherently more powerful than another. Figure 9 illustrates the performance of different models truncated at different Ns. The same as the observations in previous experiments, CMF achieves the best performance on all the four datasets. Meanwhile, we can also see that the margin between CMF and other baselines becomes larger when N increases. This means that CMF obtains the denser hit at each piece of the ranked list, thus its prediction accuracy increases faster than other baselines. We notice that the increasing speed of LDA declines faster than MF-based models, which means a sparser hit at the tail of the ranked list. We can also find that

TABLE 2
Training time per-iteration of different models with varying latent factors K

			Yelp						**MovieLens**		
K	**4**	**8**	**12**	**16**	**20**	**K**	**4**	**8**	**12**	**16**	**20**
CMF	2.88s	3.81s	8.45s	13.92s	59.99s	**CMF**	**1.57s**	**2.53s**	**2.89s**	**5.09s**	10.29s
WMF	8.71s	15.86s	23.83s	29.85s	34.71s	**WMF**	2.36s	3.45s	5.33s	6.81s	8.59s
FastALS	**1.49s**	**2.87s**	**3.05s**	**3.97s**	**5.36s**	**FastALS**	2.26s	3.26s	4.49s	6.03s	**7.10s**
			Flixster						**Ciao**		
K	**4**	**8**	**12**	**16**	**20**	**K**	**4**	**8**	**12**	**16**	**20**
CMF	**2.84s**	**3.98s**	**6.43s**	15.85s	40.72s	**CMF**	0.33s	0.89s	2.42s	5.65s	12.42s
WMF	6.94s	12.65s	18.01s	24.38s	30.58s	**WMF**	0.37s	0.71s	1.25s	1.60s	2.25s
FastALS	3.90s	7.81s	9.26s	**10.90s**	**13.30s**	**FastALS**	**0.18s**	**0.30s**	**0.44s**	**0.59s**	**0.65s**

CMF almost achieves linear growth with the increasing of N, which means that the density of hits is balanced in the ranked list.

Efficiency is another important criteria in developing recommender systems. The same as MF-based models, CMF only needs a traversal over the observed data by resorting to some memoizing strategies. Furthermore, since we have derived analytical solutions for all the parameters, and the update equation of each latent factor (U_i or V_j) is independent of each other, we can easily achieve an efficient and parallelized implementation for CMF. Since there are too many update equations in our model, we do not give the detailed time complexity analysis. Here we compare the efficiency of different models empirically, and show the actual training time per iteration in Table 2. As we can see, CMF achieves competitive efficiency compared with the other MF-based models, WMF and FastALS. When K is smaller than 16, CMF performs the best on MovieLens and Flixster. In most cases, CMF is faster than WMF, and just a little slower than FastALS. However, one drawback of CMF is that when K becomes large, the running time increases dramatically. The reason comes from the Kronecker product which leads to the inversion of an $mn \times mn$ matrix in updating the transforming matrices, T_u and T_v. However, precious work [52], [51] has already derived very efficient algorithms which reduce the time complexity to the inversion of an $m \times m$ matrix. In this paper, we only give the general but inefficient method for derivation clarity. Even so, the running time is acceptable for practical applications.

6 Conclusion

This paper proposes a novel model named Correlated Matrix Factorization (CMF) for personalized recommendation with implicit feedback. CMF elegantly combines MF and CCA into a unified model so that the prior correlation between the user and the item factors is well captured. Meanwhile, the ratings are measured as the semantic association between U and V rather than a simple inner product, which makes CMF more expressive in modeling the underlying semantics of data. Comprehensive evaluations on four different datasets show that CMF is competitive, usually better than existing state-of-the-art baselines. With increasing work focusing on recommender systems, we believe that our proposed model is promising to advance the researches in this field.

References

[1] B. M. Marlin, R. S. Zemel, S. Roweis, and M. Slaney, "Collaborative filtering and the missing at random assumption," in *UAI*, 2007.

[2] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang, "One-class collaborative filtering," in *ICDM*, 2008, pp. 502–511.

[3] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *NIPS*, vol. 20, 2011, pp. 1–8.

[4] A. Paterek, "Improving regularized singular value decomposition for collaborative filtering," in *Proceedings of KDD cup and workshop*, vol. 2007, 2007, pp. 5–8.

[5] Y. Koren and R. Bell, "Advances in collaborative filtering," in *Recommender systems handbook*. Springer, 2011, pp. 145–186.

[6] S. Rendle and C. Freudenthaler, "Improving pairwise learning for item recommendation from implicit feedback," in *WSDM*. ACM, 2014, pp. 273–282.

[7] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *ICDM*. Ieee, 2008, pp. 263–272.

[8] X. He, H. Zhang, M.-Y. Kan, and T.-S. Chua, "Fast matrix factorization for online recommendation with implicit feedback," in *SIGIR*, vol. 16, 2016.

[9] D. Liang, L. Charlin, J. McInerney, and D. M. Blei, "Modeling user exposure in recommendation," in *WWW*. International World Wide Web Conferences Steering Committee, 2016, pp. 951–961.

[10] A. N. Tikhonov and V. Y. Arsenin, "Solutions of ill-posed problems," 1977.

[11] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *WSDM*. ACM, 2011, pp. 287–296.

[12] D. Lian, C. Zhao, X. Xie, G. Sun, E. Chen, and Y. Rui, "Geomf: joint geographical modeling and matrix factorization for point-of-interest recommendation," in *SIGKDD*. ACM, 2014, pp. 831–840.

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

[14] H. M. Wallach, D. M. Mimno, and A. McCallum, "Rethinking lda: Why priors matter," in *Advances in neural information processing systems*, 2009, pp. 1973–1981.

[15] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, 2003.

[16] D. Blei and J. Lafferty, "Correlated topic models," *NIPS*, 2006.

[17] F. R. Bach and M. I. Jordan, "A probabilistic interpretation of canonical correlation analysis," *Technical report, Statistics Dept., UC Berkeley*, 2005.

[18] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *UAI*. AUAI Press, 2009, pp. 452–461.

[19] H. Steck, "Training and testing of recommender systems on data missing not at random," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010, pp. 713–722.

[20] R. Devooght, N. Kourtellis, and A. Mantrach, "Dynamic matrix factorization with priors on unknown values," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 189–198.

[21] I. Pilászy, D. Zibriczky, and D. Tikk, "Fast als-based matrix factorization for explicit and implicit feedback datasets," in *Proceedings of the ACM conference on Recommender systems*, 2010, pp. 71–78.

[22] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme, "Fast context-aware recommendations with factorization machines," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 2011, pp. 635–644.

[23] H. Zhang, F. Shen, W. Liu, X. He, H. Luan, and T.-S. Chua, "Discrete collaborative filtering," in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2016, pp. 325–334.

[24] J. D. Carroll and J.-J. Chang, "Analysis of individual differences in multidimensional scaling via an n-way generalization of eckart-young decomposition," *Psychometrika*, vol. 35, pp. 283–319, 1970.

[25] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in neural information processing systems*, 2001, pp. 556–562.

[26] D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[27] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural computation*, vol. 19, no. 10, pp. 2756–2779, 2007.

[28] S. Rendle and L. Schmidt-Thieme, "Online-updating regularized kernel matrix factorization models for large-scale recommender systems," in *Proceedings of the 2008 ACM conference on Recommender systems*. ACM, 2008, pp. 251–258.

[29] D. Zhang, Z.-H. Zhou, and S. Chen, "Non-negative matrix factorization on kernels," in *Pacific Rim International Conference on Artificial Intelligence*. Springer, 2006, pp. 404–412.

[30] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural computation*, vol. 16, no. 12, pp. 2639–2664, 2004.

[31] G. Miao, Z. Guan, L. E. Moser, X. Yan, S. Tao, N. Anerousis, and J. Sun, "Latent association analysis of document pairs," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 1415–1423.

[32] P. Dhillon, D. P. Foster, and L. H. Ungar, "Multi-view learning of word embeddings via cca," in *Advances in Neural Information Processing Systems*, 2011, pp. 199–207.

[33] J. Li, H. Dani, X. Hu, J. Tang, Y. Chang, and H. Liu, "Attributed network embedding for learning in a dynamic environment," in *Proceedings of the 26th ACM International Conference on Conference on Information and Knowledge Management*. ACM, 2017, pp. 387–396.

[34] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix t-factorizations for clustering," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 126–135.

[35] T. Wang, X. Jin, X. Ding, and X. Ye, "User interests imbalance exploration in social recommendation: A fitness adaptation," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, 2014, pp. 281–290.

[36] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, N. Oliver, and A. Hanjalic, "Climf: learning to maximize reciprocal rank with collaborative less-is-more filtering," in *Proceedings of the sixth ACM conference on Recommender systems*. ACM, 2012, pp. 139–146.

[37] R. Baeza-Yates, B. Ribeiro-Neto *et al.*, *Modern information retrieval*. ACM press New York, 1999, vol. 463.

[38] C. Rudin, "The p-norm push: A simple convex ranking algorithm that concentrates at the top of the list," *Journal of Machine Learning Research*, vol. 10, no. Oct, pp. 2233–2271, 2009.

[39] K. Christakopoulou and A. Banerjee, "Collaborative ranking with a push at the top," in *WWW*, 2015, pp. 205–215.

[40] M. Volkovs and G. W. Yu, "Effective latent models for binary feedback in recommender systems," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2015, pp. 313–322.

[41] Y. Shen and R. Jin, "Learning personal+ social latent factor model for social recommendation," in *SIGKDD*. ACM, 2012, pp. 1303–1311.

[42] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in *SIGKDD*. ACM, 2011, pp. 448–456.

[43] D. Agarwal and B.-C. Chen, "flda: matrix factorization through latent dirichlet allocation," in *Proceedings of the third ACM international conference on Web search and data mining*, 2010, pp. 91–100.

[44] B. Liu, Y. Fu, Z. Yao, and H. Xiong, "Learning geographical preferences for point-of-interest recommendation," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 1043–1051.

[45] Y. He, C. Wang, and C. Jiang, "Multi-perspective hierarchical dirichlet process for geographical topic modeling," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2017.

[46] T. Hofmann, "Probabilistic latent semantic indexing," in *SIGIR*, 1999.

[47] Y. He, C. Wang, and C. Jiang, "Modeling document networks with tree-averaged copula regularization," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 2017, pp. 691–699.

[48] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical dirichlet processes," *Journal of the american statistical association*, 2006.

[49] Y. He, C. Wang, and C. Jiang, "Discovering canonical correlations between topical and topological information in document networks," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 2015, pp. 1281–1290.

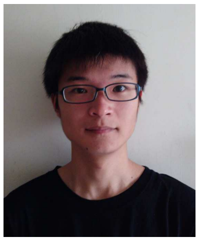
[50] M. Wainwright and M. Jordan, "A variational principle for graphical models," *New Directions in Statistical Signal Processing*, vol. 155, 2005.

[51] V. Hernández and M. Gassó, "Explicit solution of the matrix equation axb- cxd= e," *Linear Algebra and its Applications*, pp. 333–344, 1989.

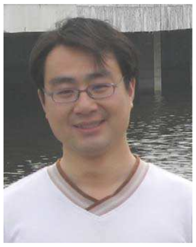
[52] K. wah Eric Chu, "The solution of the matrix equations axbcxd=e and (yadz,ycbz)=(e,f)," *Linear Algebra and its Applications*, pp. 93–105, 1987.

[53] T. Hofmann, "Latent semantic models for collaborative filtering," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 89–115, 2004.

[54] L. Lü, M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang, and T. Zhou, "Recommender systems," *Physics Reports*, vol. 519, no. 1, pp. 1–49, 2012.



Yuan He received the BS degree from the Department of Computer Science and Technology from Tongji University in 2012. He is currently working toward the PhD degree in the Department of Computer Science, Tongji University in Shanghai, China. His research interests include topic modeling, convex optimization, and recommender systems



Cheng Wang received the PhD degree from the Department of Computer Science, Tongji University in 2011. His research interests include cyber security, mobile social networks and machine learning. He is a senior member of the IEEE.

Changjun Jiang received the PhD degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 1995 and conducted postdoctoral research at the Institute of Computing Technology, Chinese Academy of Sciences, in 1997. Currently he is a professor with the Department of Computer Science and Engineering, Tongji University, Shanghai. He is also a council member of China Automation Federation and Artificial Intelligence Federation, the vice director of Professional Committee of Petri Net of China Computer Federation, the vice director of Professional Committee of Management Systems of China Automation Federation, and an Information Area Specialist of Shanghai Municipal Government. His current areas of research are concurrent theory, Petri net, and formal verification of software and intelligent transportation systems.