Privileged Matrix Factorization for Collaborative Filtering

Yali Du[†], **Chang Xu**[‡], **Dacheng Tao**[‡]
[†]Center for Artificial Intelligence, FEIT, University of Technology Sydney
[‡]UBTech Sydney AI Institute, The School of IT, FEIT, The University of Sydney
yali.du@student.uts.edu.au, {c.xu, dacheng.tao}@sydney.edu.au

Abstract

Collaborative filtering plays a crucial role in reducing excessive information in online consuming by suggesting products to customers that fulfil their potential interests. Observing that users' reviews on their purchases are often in companion with ratings, recent works exploit the review texts in modelling user or item factors and have achieved prominent performance. Although effectiveness of reviews has been verified, one major defect of existing works is that reviews are used in justifying the learning of either user or item factors without noticing that each review associates a pair of user and item concurrently. To better explore the value of review comments, this paper presents the privileged matrix factorization method that utilize reviews in the learning of both user and item factors. By mapping review texts into the privileged feature space, a learned privileged function compensates the discrepancies between predicted ratings and groundtruth values rating-wisely. Thus by minimizing discrepancies and prediction errors, our method harnesses the information present in the review comments for the learning of both user and item factors. Experiments on five real datasets testify the effectiveness of the proposed method.

1 Introduction

Recommendation systems are valuable for both providers and customers in many fields such as e-commerce, social media and entertainment. For customers, an efficient recommender system helps to narrow down the set of choices, discover new things and explore various options. For providers, a trustable recommendation to customers can increase sales or click through rates, offer personalized service and create opportunities for promotion.

The huge demand for online shopping has encouraged plentiful works to exploit users' ratings on products and hence make predictions of their future intent. Given a rating matrix composed by users' preferences on various items, recommender systems aim to predict a user's preferences on the items that are not consumed. Content-based filtering and collaborative filtering methods are two popular methods for recommender systems and have achieved prominent performance in the past. Content-based methods [Pazzani and Billsus, 2007] make a recommendation according to a user's past interests by matching up this user's perferences with the item features. On the other hand, the rooted observation for collaborative filering is that similar users have similar interests on the same item, and colaborative filtering methods [Lee *et al.*, 2012] recommend items to users that are liked by their "similar" users based on the preference connections, which can be discovered by the collaborative filtering model. Low-rank matrix factorization is one of the most popular collaborative filtering algorithms. The sparse rating matrix is factorized into the product of two low-rank matrices, user matrix and item matrix respectively, and the missing ratings are estimated by a dot product of a user vector and an item vector.

One of the main challenges in current recommendation tasks is the data sparsity, for instance, in the Amazon review dataset [Mcauley, 2013], up to 99.9% of ratings are missing. In order to enhance performance, many recommender systems start to pay attention to auxiliary information accompanying the ratings. Thesedays, a lot of online review systems contain users' reviews of an item along with its ratings, for example in Amazon review system, users are encouraged to write down their comments about the product they have purchased from this website and give an overall rating at the same time. A user might give a product lower rating because of its high price, and explains why his overall ratings are different from a spendthrift user who focuses very little on the price of products. The users' opinion might be expressed in their reviews such as "so high price!".

The powerfulness of text-based side information has been exploited by several state-of-the-art methods. Hidden factor as topics (HFT) [Mcauley, 2013] modeled multinomial topic distributions of review comments by Latent Dirichlet Allocation and casts rating prediction as a matrix factorization problem while topic distribution variables are linked to user vector or item vector. BoWLF [Almahairi *et al.*, 2015] used a bag of words to represent review comments and maximized joint probabilities of these words along with minimizing prediction error of matrix factorization model for collaborative prediction. While HFT used a topic distribution to describe all reviews from an item or user, BoWLF explore representative word distributions for reviews from different items. Overall, both of them use item vectors to represent topic distributions

or word distributions of reviews from an item.

Existing matrix factorization based methods commendably utilize the review information for collaborative filtering, however, an obvious deficiency is that they explore review comments to describe either users or items, but ignore the phenomenon that review comments are simultaneously associated with both users and items. In practice, different users may have distinct comments on same items, meanwhile, different items can receive disparate comments from the same user. Thus it is necessary to maximize the values of review comments in collaborative filtering from both user and item aspects.

In this paper, we propose privileged matrix factorization method (PriMF for short) that utilizes extra information in the learning of both user and item vectors for collaborative filtering. We take the review comments as privileged information which are corresponding to rating values. After representing each review as a feature vector in privileged feature space, we learn a privileged function on these feature vectors that describe the discrepancies between predictions and groundtruth ratings. By minimizing the prediction error and the discrepancies depicted by the privileged function, the proposed PriMF benefits the learning of latent factors for both users and items. Experiments on five real datasets show the effectiveness of proposed privileged matrix factorization method.

In Section 2, we discuss related works. We introduce the formalization of our method in Section 3, and give the optimization procedure in Section 4. Empirical verifications are presented in Section 5, followed by conclusions in Section 6.

2 Related Work

Recommender systems can be generally classified into two types: content-based filtering [Pazzani and Billsus, 2007] and collaborative filtering (CF). Content-based filtering algorithms recommend items to users according to their previous interests. Specifically, a rating is estimated by matching up item features according to users' preferences. Collaborative filtering alligns with the assumption that users who share similar interests on an item in the past are more likely to hold similar opnions on other items compared with a randomly chosen user.

While collaborative filtering has become one of the most used recommendation approaches, it has two important subclasses: memory-based and model-based approaches [Lee *et al.*, 2012]. Memory-based methods [Sarwar *et al.*, 2001] make use of the rating matrix and make recommendations by the relationship between the queried user and item and the known ratings. Model-based filering algorithms fit a parametric model to the sparse rating matrix and then issue recommendations using the fitted model. For example, Bayesian networks [Su and Khoshgoftaar, 2006; Miyahara and Pazzani, 2002] train a Bayesian classifier based on the given rating matrix, and clustering-based methods [Wang *et al.*, 2015] split the set of users or items on the given rating matrix thus taking them as basis for future predictions. Recently, low-rank matrix factorization methods are successfully applied into recommender systems [Rennie and Srebro, 2005; Salakhutdinov and Mnih, 2011]. Under the low-rank assumption, a rating matrix can be expressed as a product of two low-rank matrices [Liu and Tao, 2016]. By fitting the two low-rank matrices to the given ratings, prediction of a user's interests on a new item can be made by multiplying corresponding user vector and item vector. Other recommerder approaches leverage rating data and side information to enhance performance [Shi *et al.*, 2014] .

Inductive matrix completion (IMC) is a typical example of using side infromation for collaborative filtering. It was initially proposed and analyzed by [Jain and Dhillon, 2013]. With both user information and item properties at hand such as MovieLens, the main idea is to fit the rating matrix by corresponding user's feature vector and item's feature vector along with an underlying unknown matix. IMC has been applied to multi-label, multi-class learning and semi-supervised clustering problems. By taking a multi-label or multi-class learning as a collaborative filtering problem in which the known labels compose a label matrix, the label predictions are now equivalent to predicting the missing entries in the label matrix based on the given label information and sample features[Xu *et al.*, 2016]. Semi-supervised clustering can be taken as a collaborative filtering as well. Given a set of objects to be clustered and some known clusters of these objects, any two objects' relationship (in or not in the same cluster) can be encoded in a pairwise similarity matrix. Taking object featuer vectors as side information, the clustering task thus is accomplished by a IMC model [Yi *et al.*, 2013].

The side information applied in recommender systems varies. Item features such as "title" and "genre" as side information are employed in [Tso and Schmidt-Thieme, 2006]; and user features such as "gender", "age" and "occupation" are used in [Kim *et al.*, 2016; Agarwal and Chen, 2009; Park *et al.*, 2013] on MovieLens datasets. The movie plots as item features are applied in [Ning and Karypis, 2012] which can be fetched from IMDB. Other researchers explore customized side information, such as "date" and users' previous rating history [Porteous *et al.*, 2010] in Netflix prize dataset, and "clicks", "views", number of ads [Menon *et al.*, 2011] in Yahoo! traffic stream data. Another interesting side information is the user's tags on items which implies connections between users and items [Saha *et al.*, 2015; Fernández-Tobías and Cantador, 2014; Bao *et al.*, 2012]. In social networks, neighbors of a user are usually provided. Given a trust network connecting each user and its trustees, some user-based collaborrative filtering methods generate predictions by aggregating the ratings of a user's trustees [Massa and Avesani, 2007].

Despite the various types of side information discussed above, thesedays, more and more attention is paid to review texts that are often accompanied with the ratings. Observing the rich information in the review text, there are severeal efforts that try to improve rating prediction by incorporating these information into latent factor methods. [Mcauley, 2013] modeled the rating prediction problem as matrix factorization. To link the latent dimension of user vectors (or item vectors) to the number of hidden topics, the topic distribution variable one user's reviews is represented by the user vector through softmax normalization. [Ling *et al.*, 2014] proposes

a combined approach of content-based and collaborative filtering which harness information from ratings and reviews together. [Almahairi *et al.*, 2015] presents a distributed Bag-of-Words method which infers probability distribution of BoW by affine transformation on item representation with softmax normalization.

3 Privileged Matrix Completion

In this section, we give the formalization of our method and discuss its intuitions in detail.

3.1 Problem Statement

Considering a database composed of (user, item, rating) tuples from n users' preferences on m items, this database can be compactly denoted as a preference matrix R, which is usually sparse. The elements in the matrix R represent corresponding user's liking of the item. In real applications, the values are usually integer ratings settled in $[1, L]$. A recommender system manages to predict a user's preference of an unconsumed item which corresponds an unseen entry in the rating matrix R.

3.2 Privileged Matrix Factorization

Given the rating matrix R, the common assumption is that R is low-rank and a matrix factorization method fits R with a low-rank matrix $X = UV^\top$. To predict ordinal values of R with real-valued X, a set of thresholds $\{\theta_1, ..., \theta_{L-1}\}$ are needed. In the hard-margin settings, X would be required as:

$$\theta_{R_{ij-1}} + 1 \leq X_{ij} \leq \theta_{R_{ij}} - 1, \tag{1}$$

where θ_0 and θ_L are $-\infty$ and $+\infty$ respectively. The hard-margin setting is often known to be powerless in non-separable case. Therefore to tame this problem, slack variables ξ_{ij} are introduced, and Eq. (1) is relaxed to

$$\theta_{R_{ij-1}} + 1 - \xi_{ij} \leq X_{ij} \leq \theta_{R_{ij}} - 1 + \xi_{ij},\ \xi_{ij} > 0. \tag{2}$$

To avoid the predictions that cross rating-boundaries, not only two immediate constraints $X_{ij} \leq \theta_{R_{ij}} - 1$ and $X_{ij} \geq \theta_{R_{ij-1}} + 1$ are penalized, but also $X_{ij} \leq \theta_l - 1, \forall l > R_{ij}$ and $X_{ij} \geq \theta_l + 1, \forall l < R_{ij}$. By introducing indicating variable T:

$$T_{ij}^l = \begin{cases} +1 & \text{for } l \geq R_{ij}, \\ -1 & \text{for } l < R_{ij}, \end{cases}$$

Eq. (2) is reformulated to the following compact form:

$$T_{ij}^l \cdot (\theta_l - X_{ij}) \geq 1 - \xi_{ij};\ \forall (i,j) \in \Omega, 0 \leq l \leq L. \tag{3}$$

We denote $\Phi = \{U, V\}$ as the variable set and $\mathcal{R}(\Phi)$ as the regularization term on variables in Φ to control the model complexity. Under the above constraints in Eq. (3), the optimization problem is formulated as:

$$\begin{aligned} \min_{\Phi,\theta} \quad & \mathcal{R}(\Phi) + C \sum_{(i,j)\in\Omega} \xi_{ij} \\ \text{s.t.} \quad & T_{ij}^l \cdot (\theta_l - X_{ij}) \geq 1 - \xi_{ij}. \end{aligned} \tag{4}$$

Observing that a rating indicating the user's preference is often accompanyed with a review text, it is believed that a piece of comment text contains rich information about the user's opinions on an item. In real world, it is obvious that people need less training data when they learn a new task than that needed by training a machine on the same task. The rooted reason behind this phenomenon is that we are taught by teachers usually. A teacher has the experience on this task and has her own trained system in her mind. Although she cannot transfer her decision mechanism directly into your brain, she can teach you by generating privileged information that reflects her belief in the development of this mechanism. For example, when it comes to predict a user's preference over two movie genres like "comedy" and "thriller", the training data are two movies with genre "comedy|drama" and "drama|thriller" and both movies are rated as 5. In this case, matrix factorization model might be confused at the ratings. But from the comments that "A great comedy for family time" and "I don't like the kind of plots, but Thomas's acting is exciting", it is more confident to conclude that this consumer prefers "comedy" more than "thriller". This is exactly the role that we think the review comments can play in the training process. It reflects a user's thoughts in the decision procedure, and thus explain discrepancies between predictions and ground truth ratings.

Review information is from a different resource compared to rating data and lies in another feature space. The feature vector which is denoted as Z_{ij} corresponds to rating R_{ij}. Now we consider a rating prediction problem based on the quaternions (user, item, rating, review). Considering review texts as privileged information of corresponding ratings, the privileged function is defined as $\phi(\boldsymbol{Z}_{ij}, w, b) = w\boldsymbol{Z}_{ij} + b, w \in \mathbb{R}^d$. To ease the presentation, we append the bias term b to the end of the weight vector w and still denote it as w, $w \leftarrow [w, b]$. Similarly, we append 1 to the end of every feature vector as $Z_{ij} \leftarrow [Z_{ij}, 1]$. The slack variables in Problem (4) can be reformulated as $\xi_{ij} = \phi(\boldsymbol{Z}_{ij}, w), \forall (X_{ij}, \boldsymbol{Z}_{ij})$. Since we can learn different privileged functions for different items, we use W to denote coefficient variables in ϕ.

Adding W to the variable set $\Phi = \{U, V, W\}$, $\mathcal{R}(\Phi)$ takes the form as:

$$\mathcal{R}(\Phi) = \frac{1}{2}(\|U\|_F^2 + \|V\|_F^2 + \gamma\|W\|_F^2). \tag{5}$$

Our objective function thus can be written as:

$$\begin{aligned} \min_{\Phi,\theta} \quad & \mathcal{R}(\Phi) + \alpha \sum_{(i,j)\in\Omega} [\phi(\boldsymbol{Z}_{ij}, W)]_+ \\ \text{s.t.} \quad & T_{ij}^l \cdot (\theta_{jl} - X_{ij}) \geq 1 - [\phi(\boldsymbol{Z}_{ij}, W)]_+, \end{aligned} \tag{6}$$

where $[y]_+ = \max\{0, y\}$.

The variable ξ_{ij} in Problem (4) is introduced to represent slacks triggered by rating data that are not linearly separable. With privileged information in hand, the slacks in non-separable situation can be learned by a correcting function that is dependent on the additional information and has low VC dimension. This learning paradigm is firstly applied to support vector machine in classification problem [Vapnik and Vashist, 2009; Vapnik and Izmailov, 2015] and then extended to various applications [Yang *et al.*, 2016]. In collaborative filtering, this slack margin can be triggered by various possibilities, such as users' special appetite on an item that is not consistent with item features, or users' sudden change in one

criterion for rating. And it is even possible that rating data is corrupted when it is collected. Therefore, one linear correcting function $\phi(Z_{ij}, W)$ is not representative enough for correcting the non-linearly separable ratings.

To reinforce the model's tolerance to outliers or noises, we represent ξ_{ij} in Problem (4) by a linear privileged function and a tolerance term ζ_{ij}. The final model takes the form:

$$\begin{aligned} \min_{\Phi,\theta} \quad & \mathcal{R}(\Phi) + \alpha \sum_{(i,j)\in\Omega} [\phi(\boldsymbol{Z}_{ij}, W)]_+ + \beta \sum_{(i,j)\in\Omega} \zeta_{ij} \\ \text{s.t.} \quad & T^l_{ij} \cdot (\theta_{jl} - X_{ij}) + [\phi(\boldsymbol{Z}_{ij}, W)]_+ \geq 1 - \zeta_{ij}, \\ & \zeta_{ij} \geq 0. \end{aligned} \tag{7}$$

Thus the offsets for hard-margins in Problem (4) are splited into a discrepancy term represented by privileged function and a tolerance term that allow the existence of outliers. The above problem is equavelent to the following problem:

$$\begin{aligned} \min_{\Phi,\theta} \quad & f(\Phi,\theta) = \mathcal{R}(\Phi) + \alpha \sum_{(i,j)\in\Omega} [\phi(\boldsymbol{Z}_{ij}, W)]_+ \\ & + \beta \sum_{(i,j)\in\Omega} h(T^l_{ij} \cdot (\theta_{jl} - X_{ij}) + [\phi(\boldsymbol{Z}_{ij}, W)]_+). \end{aligned} \tag{8}$$

where $h(u)$ is the hinge loss and it can be, for example, $h(u) = \max\{0, 1-u\}$.

After Problem (8) is addressed, we get the optimal solution of Problem (8) that $\Phi^* = \{U^*, V^*, W^*, \theta^*\}$. Denote $X^* = U^* V^{*T}$, we infer the ratings of each item as:

$$R^*_{*j} = 1 + \max\{l | X^*_{*j} \geq \theta^*_{jl}, l = 0, 1, ..., L-1\}, \tag{9}$$

where R^* is the optimal output of our algorithm.

4 Optimization Methods

In the following, we present our optimization method. Let U_{i*} and V_{j*} denote the i-th and j-th row of U and V respectively. For simplicity, the independent variable of function $h(u)$ is defined as $u^l_{ij} = T^l_{ij}(\theta_{jl} - X_{ij}) + [\phi(\boldsymbol{Z}_{ij}, W)]_+$. Taking derivatives w.r.t. U in Problem (8), we obtain:

$$\frac{\partial f}{\partial U_{i*}} = U_{i*} - \beta \sum_{jl} T^l_{ij} V_{j*} \cdot \frac{\partial h}{u}|_{u=u^l_{ij}}, \tag{10}$$

and the partial derivative w.r.t. V is analogous. Define the indicator function $\mathbf{1}(x)$ as 1 if $x > 0$ and 0 otherwise. The partial derivative with regard to w_j is:

$$\begin{aligned} \frac{\partial f}{\partial w_j} = w_j + \alpha \sum_i Z_{ij} \cdot \mathbf{1}(w_j^T Z_{ij}) \\ + \beta \sum_{il} Z_{ij} \cdot \mathbf{1}(w_j Z_{ij}) \frac{\partial h}{u}|_{u=u^l_{ij}}. \end{aligned} \tag{11}$$

And the partial derivative w.r.t. θ_{jl} element-wisely is:

$$\frac{\partial f}{\partial \theta_{jl}} = \beta \sum_i T^l_{ij} \cdot \frac{\partial h}{u}|_{u=u^l_{ij}}. \tag{12}$$

With these gradients in hand, we can turn to gradident descent methods for solving Problem (8).

For optimization of U, V, W, and θ we choose conjugate gradient descent method [Hager and Zhang, 2006]. For simplicity of presentation, we denote y as concatenated vector of variables U, V, W and θ, and denote $g(y) = f(\Phi, \theta)$. We have the general form of conjugate gradient as

$$y_{t+1} = y_t + \eta_t \cdot p_t, \tag{13}$$

where η_t is a sequence of step sizes generated by a line search algorithm that can guarantee the decrease in objective function. p_t is the descent direction genearted by the following rule:

$$p_t = -\nabla g_t + \tau_{t-1} p_{t-1}, \tag{14}$$

and ∇g_t is the abbreviation of $\nabla g(y_t)$. τ_t is the conjugate gradient parameter and it is chosen to ensure the conjugacy between conjugate gradient directions.

Several options of τ_k can be used in conjugate gradient algorithm. Here we choose the Polak-Ribiere (PR+) method for calculating the CG update parameter:

$$\tau_t = \max\{\frac{\langle \nabla g_t, \nabla g_t - \nabla g_{t-1} \rangle}{\langle \nabla g_{t-1}, \nabla g_{t-1} \rangle}, 0\}. \tag{15}$$

Note that τ_t is chosen to ensure that p_t is orthogonal to all previous search directions by a symmetric positive definite matrix. When two consecutive gradients are far away from orthogonal or conjugate directions are almost orthogonal to gradients, we restart the searching process by reset the exploration direction to the steepest descent direction. The restart condition takes the form: $\langle \nabla g_t, \nabla g_{t-1} \rangle / \langle \nabla g_t, \nabla g_t \rangle \geq \nu$ or $\langle p_{t+1}, \nabla g_{t+1} \rangle \geq 0$. Here ν is set to 0.1 as recommended by [Nocedal and Wright, 2006].

After selecting the search direction, strong Wolfe conditions [Nocedal and Wright, 2006] are applied to choose step size η_k that can guarantee sufficient decrease on objective function. The conditions take the form as:

$$g(y_t + \eta_t p_t) \leq g(y_t) + c_1 \eta_t \nabla g_t^T p_t, \tag{16a}$$

$$|\nabla g(y_t + \eta_t p_t)| \leq c_2 |\nabla g_t^T p_t|. \tag{16b}$$

A secant method is used to find the root of directional derivatives. The overall algorithm is summarized in Algorithm 1.

Due to the undifferentiability of standard hinge loss at zero point, the smooth hinge loss is adopted in this paper [Rennie and Srebro, 2005]:

$$h_u = \begin{cases} 0 & \text{for } u \geq 1, \\ \frac{(1-u)^2}{2} & \text{for } 0 \leq u < 1, \\ \frac{1}{2} - u & u < 0. \end{cases}$$

The smooth hinge loss does not continously reward the correct predictions, meanwhile, it is differentiable at zero point.

Strong Wolfe conditions guarantee the proposed optimization to converge. However, due to the non-convexity of the objective function, the solution may fall into the local minima. In experiments we will repeat the training process several times to avoid spurious local minima.

5 Experiments

We verify the proposed method on five datasets from Amazon reviews [Mcauley, 2013]. These data were collected from Amazon website with the years spanning from 1998 to 2013. The five datasets refer to five category of products, watches,

Algorithm 1 Conjugate Gradient Method for Problem (8)

Input: R: Incomplete matrix, ϵ: stopping criteria, k: latent dimension, T: max iteration
Output: R^*: the optimal approximation of R

Init: $y_0 = [U_0(:); V_0(:); \theta_0(:); W_0]$;
eval: $g_0 = g(y_0),\ \nabla g_0 = \nabla g(y_0)$;
set: $p_0 = -\nabla g_0, t \leftarrow 0$;
while $t < T$ **and** $\|\nabla g_t\| > \epsilon$ **do:**
choose step size: η_t satisfies Cond. (16);
update $y_{t+1} = y_t + \eta_t p_t$;
eval: ∇g_{t+1};
get CG direction:
$\tau_{t+1}^{PR} \leftarrow \frac{\langle \nabla g_{t+1}, \nabla g_{t+1} - \nabla g_t \rangle}{\langle \nabla g_{t+1}, \nabla g_{t+1} \rangle}$;
$\tau_{t+1}^{PR+} \leftarrow \max\{\tau_{t+1}^{PR}, 0\}$;
$p_{t+1} \leftarrow -\nabla g_{t+1} + \tau_{t+1}^{PR+} p_t$;
$t \leftarrow t + 1$;
check restart condition:
if $\frac{\langle \nabla g_k, \nabla g_{k-1} \rangle}{\langle \nabla g_k, \nabla g_k \rangle} \geq \nu$ **or** $\langle p_{t+1}, \nabla g_{t+1} \rangle \geq 0$:
reset: $p_{t+1} = \nabla g_{t+1}$;
end if
end while
Return: R^* predicted by Eq. 9.

musical instruments, industrial and scientific, gourmet foods and books. Every rating in the datasets is associated with a piece of review comment. The detailed information for these datasets are in Table 1.

5.1 Experiments Settings

We follow the settings used in [Mcauley, 2013; Almahairi *et al.*, 2015]. For each dataset, 80% ratings are randomly selected for training and the remaining 20% are evenly splited into validation and test. The review information is only available in the trainning phase.

To extract feature vectors from reviews of different lengths, we adopt a neural network language model, Paragraph Vector [Le and Mikolov, 2014] for help. As an extension of Word Vector [Mikolov *et al.*, 2013b; 2013a], it can preserve semantics of words as well.

The size of the privileged feature vector is fixed to $k = 10$ which is consistent with the setting in HFT. The rank r of X is determined by cross validation. In the experiments, the optimum r on different datasets is around $r = 50$. Baseline methods have the similar parameter settings. The measurement used in the paper is mean square error (MSE):

$$\text{MSE} = \frac{\|P_\Omega(R^* - R)\|_F^2}{|\Omega|}. \tag{17}$$

Where R^* is the optimal estimation of the ground truth rating matrix R and P_Ω is the sampling operator.

5.2 Baseline Methods

We compare the proposed PriMF with four baseline methods.

- **FM3F:** Fast Maximum Margin Matrix Factorization [Rennie and Srebro, 2005] is a basic collaborative filtering model that minimizes the discrepancies between erroneous predictions and ground truth ratings.
- **HFT:** Hidden Factors as Topics [Mcauley, 2013] is a combined approach of matrix factorization and LDA. It models multinomial topic distributions by using either user factors or item factors which means that it takes all reviews by a particular item as a document or that by a particular user as a document and maximizes the joint probabilities of the text corpus.
- **BoWLF:** Bag-of-Words latent factor model [Almahairi *et al.*, 2015] takes all reviews by an item as a document. For different document, the word distributions in the vocabulary are modeled by a weight matrix-vector product with the item correspondingly. By representing each review as a bag of words, BoWLF maximize the joint probabilities of document corpus by multiplying over all words in each document.
- **LMLF:** This is a recurrent neural network language model [Almahairi *et al.*, 2015] that takes a sequence of words as input and preserve their order. It models different items to have their respective word distributions. The probability of current word conditioned on previous words is modeled by an affine transformation of a recurrent function and LSTM is adopted as its recurrent function module.

5.3 Evaluation Results

For both the comparative methods and PriMF, we randomly initialize the training for five times and select parameters that have lowest error on validation set. We select α, β and γ for PriMF from $\{10^{-5}, 10^{-4}, ..., 10^4\}$. For baseline methods, parameters are validated in the recommended range in original papers. The averaged MSE and standard deviation on test sets are reported in Table 2.

PriMF achieves a remarkable improvement over FM3F after introducing privileged information and gains consistent improvement on other three methods. In general PriMF and comparative methods can all be classified into matrix factorization category and all of them make predictions by multiplying user vector and item vector respectively. The difference exists in the way of using reviews in the training phase. While HFT, BoWLF and LMLF model topic or word distributions of review contents by item vectors without considering user vectors, their performance sacrifices.

To test the performance of these approaches under different splits of data, we repeat random splits of data for five times and keep the percentage used in previous experiments. In every split, the collaborative filtering are trained on a different set of ratings and review comments. The parameters are chosen on the validation set and the performance of different approaches on test set are reported in Fig. 1. From Fig. 1, we find that the performance of each method under different splits varies dramatically, but the relative performance across five methods remains consistent. This observation indicates that the results tested under different splits of data are not straightforwardly comparable, but each single random split can be used to evaluate and select models as long as the split is kept the same for all evaluated methods.

Table 1: Dataset Information

Dataset	users	items	review	words	words/review	reviews/item
Watches	62,041	10,318	68,356	5,436,671	79.53	6.62
Musical Instruments	67,007	14,182	85,405	7,442,294	87.14	6.02
Industrial Scientific	29,590	22,622	137,042	6,920,151	50.50	6.06
Gourmet Foods	112,544	23,476	154,635	10,542,984	68.18	6.59
Books	2,588,991	929,264	12,886,488	1,613,603,531	125.22	13.87

Table 2: Mean square error comparisons with baseline methods on different datasets. Values in brackets indicate standard deviation error.

	(a) FM3F	(b) HFT	(c) BoWLF	(d) LMLF	(e) PriMF	PriMF improvement over (a)	(b)	(c)	(d)
Watches	1.571(0.02)	1.468 (0.03)	1.466 (0.03)	1.473 (0.03)	1.451(0.02)	7.62%	1.13%	1.00%	1.47%
Musical Instrument	1.448(0.03)	1.382 (0.02)	1.375 (0.02)	1.388 (0.02)	1.355(0.01)	6.42%	1.95%	1.45%	2.37%
Industrial	0.354(0.01)	0.354 (0.01)	0.352 (0.01)	0.356 (0.01)	0.324(0.01)	8.47%	8.47%	7.95%	8.99%
Gourmet Foods	1.732(0.01)	1.486 (0.02)	1.464 (0.02)	1.478 (0.02)	1.454(0.02)	16.07%	2.18%	0.71%	1.65%
Books	1.381(0.01)	1.141 (0.00)	1.10 (0.02)	1.110 (0.01)	1.09(0.01)	20.80%	4.15%	0.57%	1.47%
Overall	1.297	1.166	1.151	1.161	1.136	12.46%	2.63%	1.38%	2.19%

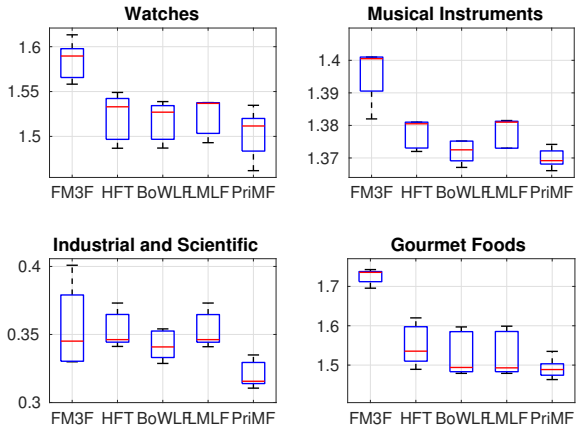

Figure 1: Box and whisker plot of different random split of data. Center line represents median. Box extents show first quarter and third quarter. Whisker extents illustrate maximum and minimum values

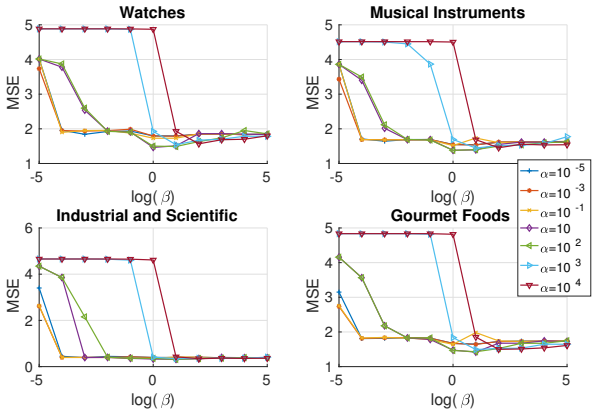

Figure 2: Mean square error of PriMF changing with α and β on different datasets

5.4 Parameter Analysis

In this section, we analyze the influence of two hyperparameters α and β in optimization problem (8). α controls the weight of privileged function that models discrepancies between predictions and groundtruth values, while β constrains prediction loss. We report the performance of PriMF under different choices of α and β in Fig. 2. When α is fixed, we can choose the value of β from a large range that leads to consistent good performance of PriMF, and vice versa. We conclude that our method is stable with respect to different choices of parameters.

6 Conclusions

In this paper, we study a privileged matrix factorization method for collaborative filtering. We utilize review texts that are in companion with rating values to assist the learning of user and item factors. In contrast to existing approaches that use review texts to describe either user or item factors, we notice that a review comment is related to a pair of user and item concurrently, and utilize it in the learning of both user and item factors. We take review texts as privileged information for matrix factorization and express them as feature vector through a Paragraph Vector transformation. The discrepancies between predictions and ground truth ratings are modeled by a privileged function dependent on review vector. We testify our model on several challenging Amazon review datasets and achieve better performance. While the remarkable strength of our method is discussed above, the weakness shared with the other collaborative filtering methods is the implicit assumption on the complete review comments. In practice, not every rating comes with a piece of review comment. In the future, we are considering the scenario in which some reviews are missing which is a common issue as well.

Acknowledgments

This work is supported by Australian Research Council Projects FT-130101457, DP-140102164, LP-150100671.

References

[Agarwal and Chen, 2009] Deepak Agarwal and Bee-Chung Chen. Regression-based latent factor models. In *SIGKDD*, pages 19–28. ACM, 2009.

[Almahairi *et al.*, 2015] Amjad Almahairi, Kyle Kastner, Kyunghyun Cho, and Aaron Courville. Learning Distributed Representations from Reviews for Collaborative Filtering. *RecSys '15*, pages 147–154, 2015.

[Bao *et al.*, 2012] Tengfei Bao, Yong Ge, Enhong Chen, Hui Xiong, and Jilei Tian. Collaborative filtering with user ratings and tags. In *CDDM*, page 1. ACM, 2012.

[Fernández-Tobías and Cantador, 2014] Ignacio Fernández-Tobías and Iván Cantador. Exploiting social tags in matrix factorization models for cross-domain collaborative filtering. In *CBRecSys@ RecSys*, pages 34–41, 2014.

[Hager and Zhang, 2006] William W Hager and Hongchao Zhang. A survey of nonlinear conjugate gradient methods. *Pacific journal of Optimization*, 2(1):35–58, 2006.

[Jain and Dhillon, 2013] Prateek Jain and IS Dhillon. Provable inductive matrix completion. *arXiv preprint*, pages 1–22, 2013.

[Kim *et al.*, 2016] Hyunjik Kim, Xiaoyu Lu, Seth Flaxman, and Yee Whye Teh. Tucker gaussian process for regression and collaborative filtering. *arXiv preprint arXiv:1605.07025*, 2016.

[Le and Mikolov, 2014] Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. In *ICML*, volume 14, pages 1188–1196, 2014.

[Lee *et al.*, 2012] Joonseok Lee, Mingxuan Sun, and Guy Lebanon. A comparative study of collaborative filtering algorithms. *arXiv preprint arXiv:1205.3193*, 2012.

[Ling *et al.*, 2014] Guang Ling, Michael R Lyu, and Irwin King. Ratings Meet Reviews , a Combined Approach to Recommend. *RecSys*, pages 105–112, 2014.

[Liu and Tao, 2016] Tongliang Liu and Dacheng Tao. On the performance of manhattan nonnegative matrix factorization. *IEEE Transactions on Neural Networks and Learning Systems*, 27(9):1851–1863, September 2016.

[Massa and Avesani, 2007] Paolo Massa and Paolo Avesani. Trust-aware recommender systems. In *RecSys*, pages 17–24. ACM, 2007.

[Mcauley, 2013] Julian Mcauley. Hidden Factors and Hidden Topics : Understanding Rating Dimensions with Review Text. In *RecSys*, 2013.

[Menon *et al.*, 2011] Aditya Krishna Menon, Krishna-Prasad Chitrapura, Sachin Garg, Deepak Agarwal, and Nagaraj Kota. Response prediction using collaborative filtering with hierarchies and side-information. In *SIGKDD*, pages 141–149. ACM, 2011.

[Mikolov *et al.*, 2013a] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[Mikolov *et al.*, 2013b] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.

[Miyahara and Pazzani, 2002] Koji Miyahara and Michael J Pazzani. Improvement of collaborative filtering with the simple bayesian classifier. *IPSJ*, 43(11), 2002.

[Ning and Karypis, 2012] Xia Ning and George Karypis. Sparse linear methods with side information for top-n recommendations. In *RecSys*, pages 155–162. ACM, 2012.

[Nocedal and Wright, 2006] Jorge Nocedal and Stephen J Wright. Numerical optimization, second edition. *Numerical optimization*, pages 497–528, 2006.

[Park *et al.*, 2013] Sunho Park, Yong-Deok Kim, and Seungjin Choi. Hierarchical bayesian matrix factorization with side information. In *IJCAI*, 2013.

[Pazzani and Billsus, 2007] Michael J Pazzani and Daniel Billsus. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer, 2007.

[Porteous *et al.*, 2010] Ian Porteous, Arthur U Asuncion, and Max Welling. Bayesian matrix factorization with side information and dirichlet process mixtures. In *AAAI*, 2010.

[Rennie and Srebro, 2005] Jasson DM Rennie and Nathan Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *ICML*, pages 713–719. ACM, 2005.

[Saha *et al.*, 2015] Tanwistha Saha, Huzefa Rangwala, and Carlotta Domeniconi. Predicting preference tags to improve item recommendation. *In submission. SIAM SDM*, 2015.

[Salakhutdinov and Mnih, 2011] Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. Citeseer, 2011.

[Sarwar *et al.*, 2001] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW*, pages 285–295. ACM, 2001.

[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. *CSUR*, 47(1):3, 2014.

[Su and Khoshgoftaar, 2006] Xiaoyuan Su and Taghi M Khoshgoftaar. Collaborative filtering for multi-class data using belief nets algorithms. In *18th ICTAI*, pages 497–504. IEEE, 2006.

[Tso and Schmidt-Thieme, 2006] Karen HL Tso and Lars Schmidt-Thieme. Evaluation of attribute-aware recommender system algorithms on data with varying characteristics. In *PAKDD*, pages 831–840. Springer, 2006.

[Vapnik and Izmailov, 2015] Vladimir Vapnik and Rauf Izmailov. Learning using privileged information: Similarity control and knowledge transfer. *JMLR*, 16:2023–2049, 2015.

[Vapnik and Vashist, 2009] Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural networks*, 22(5):544–557, 2009.

[Wang *et al.*, 2015] Xiangyu Wang, Dayu He, Danyang Chen, and Jinhui Xu. Clustering-based collaborative filtering for link prediction. In *AAAI*, pages 332–338, 2015.

[Xu *et al.*, 2016] Chang Xu, Dacheng Tao, and Chao Xu. Robust extreme multi-label learning. In *KDD*, pages 13–17, 2016.

[Yang *et al.*, 2016] Xun Yang, Meng Wang, Luming Zhang, and Dacheng Tao. Empirical risk minimization for metric learning using privileged information. In *International Joint Conference on Artificial Intelligence*, 2016.

[Yi *et al.*, 2013] Jinfeng Yi, Lijun Zhang, Rong Jin, Qi Qian, and Anil K. Jain. Semi-supervised Clustering by Input Pattern Assisted Pairwise Similarity Matrix Completion. *ICML*, 28:1400–1408, 2013.