# Semi-Nonnegative Matrix Factorization with Global Statistical Consistency for Collaborative Filtering

Hao Ma, Haixuan Yang†, Irwin King, Michael R. Lyu
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
{hma, king, lyu}@cse.cuhk.edu.hk

†Department of Computer Science
Royal Holloway University of London
United Kingdom
haixuan@cs.rhul.ac.uk

## ABSTRACT

*Collaborative Filtering*, considered by many researchers as the most important technique for information filtering, has been extensively studied by both academic and industrial communities. One of the most popular approaches to collaborative filtering recommendation algorithms is based on low-dimensional factor models. The assumption behind such models is that a user's preferences can be modeled by linearly combining item factor vectors using user-specific coefficients. In this paper, aiming at several aspects ignored by previous work, we propose a semi-nonnegative matrix factorization method with global statistical consistency. The major contribution of our work is twofold: (1) We endow a new understanding on the generation or latent compositions of the user-item rating matrix. Under the new interpretation, our work can be formulated as the semi-nonnegative matrix factorization problem. (2) Moreover, we propose a novel method of imposing the consistency between the statistics given by the predicted values and the statistics given by the data. We further develop an optimization algorithm to determine the model complexity automatically. The complexity of our method is linear with the number of the observed ratings, hence it is scalable to very large datasets. Finally, comparing with other state-of-the-art methods, the experimental analysis on the EachMovie dataset illustrates the effectiveness of our approach.

## Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information filtering; G.1.6 [**Numerical Analysis**]: Optimization

## General Terms

Algorithms, Experimentation

## Keywords

Recommender Systems, Collaborative Filtering, Matrix Factorization, Optimization

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
*CIKM'09,* November 2–6, 2009, Hong Kong, China.
Copyright 2009 ACM 978-1-60558-512-3/09/11 ...$10.00.

## 1. INTRODUCTION

*Recommender Systems* attempt to suggest items (movies, books, music, news, Web pages, images, etc.) that are likely to interest the users. Typically, recommender systems are based on *Collaborative Filtering*, which refers to the technique for the task of predicting preferences of users by collecting taste information from many other users. The underlying assumption of collaborative filtering is that the active user will prefer those items which the similar users prefer [16]. Due to the potential commercial values and the great research challenges, *Collaborative Filtering* techniques have drawn much attention in both information retrieval [16, 25, 35, 36] and machine learning [19, 21, 23, 24, 37] communities. Collaborative filtering algorithms suggesting personalized recommendations greatly increase the likelihoods of customers making the purchases online. Hence, the developed recommendation applications have been widely deployed in several large and famous commercial Web sites, such as Amazon[1], Ebay[2], Netflix[3], Apple[4], etc.

A number of algorithms have been proposed to improve both the recommendation quality and the scalability problems. These collaborative filtering algorithms can be divided into two main categories: neighborhood-based and model-based approaches [2, 25]. Different methods are based on different assumptions. The neighborhood-based recommendation algorithms are based on the assumption that those who agreed in the past tend to agree again in the future. They usually fall into two classes: user-based approaches [2, 6] and item-based approaches [5, 25]. To predict a rating for an item from a user, user-based methods find other similar users and leverage their ratings to the item for prediction, while item-based methods use the ratings to other similar items from the user instead [4]. Despite their success in the industry, neighborhood-based methods suffer from both the data sparsity and the scalability problems. In addition to the neighborhood-based approach, the model-based approaches use the observed user-item ratings to train a predefined model. Algorithms in this category include Bayesian model [35], aspect model [9], etc.

Recently, due to its efficiency in handling very large datasets, low-dimensional factor models have become one of the most popular approaches in the model-based collaborative filtering algorithms. The premise behind a low-dimensional fac-

[1]http://www.amazon.com
[2]http://www.half.ebay.com
[3]http://www.netflix.com
[4]http://www.apple.com

**Table 1:** User-Item Matrix

| | $i_1$ | $i_2$ | $i_3$ | $i_4$ | $i_5$ | $i_6$ | $i_7$ | $i_8$ |
|---|---|---|---|---|---|---|---|---|
| $u_1$ | 5 | 2 | | 3 | | 4 | | |
| $u_2$ | 4 | 3 | | | 5 | | | |
| $u_3$ | 4 | | 2 | | | | 2 | 4 |
| $u_4$ | | | | | | | | |
| $u_5$ | 5 | 1 | 2 | | 4 | 3 | | |
| $u_6$ | 4 | 3 | | 2 | 4 | | 3 | 5 |

**Table 2:** Predicted User-Item Matrix

| | $i_1$ | $i_2$ | $i_3$ | $i_4$ | $i_5$ | $i_6$ | $i_7$ | $i_8$ |
|---|---|---|---|---|---|---|---|---|
| $u_1$ | 5 | 2 | **2.5** | 3 | **4.8** | 4 | **2.2** | **4.8** |
| $u_2$ | 4 | 3 | **2.4** | **2.9** | 5 | **4.1** | **2.6** | **4.7** |
| $u_3$ | 4 | **1.7** | 2 | **3.2** | **3.9** | **3.0** | 2 | 4 |
| $u_4$ | **4.8** | **2.1** | **2.7** | **2.6** | **4.7** | **3.8** | **2.4** | **4.9** |
| $u_5$ | 5 | 1 | 2 | **3.4** | 4 | 3 | **1.5** | **4.6** |
| $u_6$ | 4 | 3 | **2.9** | 2 | 4 | **3.4** | 3 | 5 |

tor model is that there is only a small number of factors influencing the preferences, and that a user's preference vector is determined by how each factor applies to that user [21]. Most recently, some assumptions are developed to enhance the factor models. For examples, in [31], a matrix factorization method is proposed to constrain the norms of $U$ and $V$ instead of their dimensionality; a probabilistic linear model with Gaussian observation noise is proposed in [24]; and Gaussian-Wishart priors are placed on the user and item hyperparameters in [23]. These models achieve promising prediction results.

Although these methods can effectively predict missing values, several disadvantages are unveiled, which will potentially decrease the prediction accuracy. First, in low-rank factor-based approaches, both item factor vectors and user-specific coefficients are understood as latent factors which have no physical meanings, and hence uninterpretable. Moreover, the lack of interpretability will result in the improper modeling of the latent factors. For example, these latent factors in [23, 24] are set to be in the Euclidean space, while they are nonnegative in [34]. Second, due to the sparsity of the user-item rating matrix (the density of available ratings in commercial recommender systems is often less than 1% [25]), many matrix factorization methods fail to provide accurate recommendations. In the sparse user-item rating matrix, the ratings for training the user features are rare, hence the learned user features and the coefficients cannot accurately reflect the taste of users, which will result in the bad prediction accuracy.

In this paper, aiming at providing solutions for the issues analyzed above, we propose a Semi-Nonnegative Matrix Factorization with Global Statistical Consistency (SNGSC) approach for collaborative filtering. First, we endow a new understanding on the latent compositions of the ratings, which is based on the following assumptions: (1) there are totally a number of $d$ types of items; (2) on each type of items, every user has a confidence value indicating the taste of this user on the type; (3) each item also has a quality value on each type. Based on these assumptions, we formulate the collaborative filtering algorithm as the Semi-Nonnegative Matrix Factorization problem, and propose an optimization formulation with sensitive analysis. Second, based on the observation that the statistics of the predicted ratings are not consistent with the statistics of the training data, we propose to impose the consistency between them. This consideration generates very good performance when the dataset is spare. Furthermore, we develop an algorithm to determine the model complexity automatically. The complexity of our method is linear with the number of the observed ratings, which can be applied to very large datasets. Finally, comparing with other state-of-the-art methods, the experimental analysis on the EachMovie dataset shows the effectiveness of our approach.

The rest of this paper is organized as follows. We interpret the physical meaning to latent factors in Section 2.2, conduct the sensitivity analysis in Section 2.3, and formulate the optimization problem in Section 2.4. In Section 2.5, we propose a method that determines the dimensionality automatically. In Section 3, we present an approach of imposing the consistency between the statistics given by predicted values and the statistics given by the observed data. The experimental results on EachMovie dataset are shown in Section 4. The related work is introduced in Section 5. Finally, we draw the conclusions in Section 6.

# 2. FRAMEWORK

## 2.1 Problem Definition

Without loss of generality, in this paper, we use the movie recommender systems as the example. In a collaborative prediction movie recommendation system, the inputs to the system are user ratings on the movies the users have already seen. Prediction of user preferences on the movies they have not yet seen are then based on patterns in the partially observed rating matrix $X \in R_+^{n \times m}$, where $n$ is the number of users, and $m$ is the number of movies. The value $X_{ij}$ indicates the score of item $j$ rated by user $i$. This approach contrasts with feature-based approach where predictions are made based on features of the movies (e.g. genre, year, actors, external reviews) and the users (e.g. age, gender, explicitly specified preferences, social trust networks [17, 18]). Users "collaborate" by sharing their ratings instead of relying on external information [21].

Table 1 and Table 2 are the toy examples on the problem we study. As illustrated in Table 1, each user (from $u_1$ to $u_6$) rated some items (from $i_1$ to $i_8$) on a 5-point integer scale to express the extent of favor of each item. The problem we study in this paper is how to predict the missing values of the user-item matrix effectively and efficiently. Usually, as introduced in Section 1, the density of available ratings in commercial recommender systems ($X$) is often less than 1% [25].

## 2.2 How is user-item matrix $X$ generated?

The $n \times m$ matrix $X$ contains the ratings of users on items. $X$ is generated by the users who rate the movies according to their overall feeling about the movies that they have seen. By anatomizing their overall feeling, we give a detailed analysis on the rating process as follows.

Each user has a different taste on different type of genre, actors, or something else. But with the only given rating matrix, the information for genre or actors is unknown, so we assume there are $d$ different unknown types of objects, which are named as latent types. We further assume that user $i$ has confidence $U_{ik}$ ($U_{ik} \in R_+$) on $k$-th type, and $U_{ik}$ is also the taste of user $i$ in ranking objects of type $k$; on the other hand, on $k$-th type, each item $j$ has a "true" quality value $V_{jk}$ ($V_{jk} \in R$). So to user $i$, item $j$ should be rated by user $i$ as $U_{ik} * V_{jk}$. As a result, on $k$-th type, if both the quality of object $j$ and the taste of user $i$ are high, then user $i$ will rate object $j$ with a high score.

These $d$ latent types may have cross-effects on each other. For example, War type movies may also belong to classic Hollywood sub-category. Considering the cross-effects, we assume a symmetric non-negative matrix $\Sigma_{d\times d}$, in which $\Sigma_{kl} = \Sigma_{lk}$ denotes the cross-effect between type $k$ and $l$, and $\Sigma_{kk} = \lambda_k$. Ideally, we hope that the $d$ latent types are independent, and their significance can be ordered, i.e., nonnegative significance values $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$ can be assigned to the $d$ latent types.

Consequently, on type $k$, user $i$ rates item $j$ with a score

$$\sum_{l=1}^{d} U_{ik} * V_{jl} * \Sigma_{kl},$$

where the quality $V_{jl}$ of item $j$ on type $l$ is transferred to quality $V_{jl} * \Sigma_{kl}$ by $\Sigma_{kl}$. Note that, if $\Sigma_{d\times d}$ is diagonal, then it becomes $\sum_{l=1}^{d} \lambda_k * U_{ik} * V_{jk}$. Accumulating all the different unknown types, we obtain that

$$\sum_{k=1}^{d}\sum_{l=1}^{d} U_{ik} * V_{jl} * \Sigma_{kl} = (U\Sigma V^T)_{ij},$$

where $U_k$ is the vector consisting of $U_{ik}$, $V_k$ is the vector consisting of $V_{jk}$, and $U = (U_1, U_2, \ldots, U_d)$ and $V = (V_1, V_2, \ldots, V_d)$. We consider factorizations of the form $X \approx U\Sigma V^T$, where $U \in R_+^{n\times d}$, $\Sigma \in R_+^{d\times d}$, and $V \in R^{m\times d}$.

**Remark**. According to the physical meaning of $U$ and $V$, $U$ is nonnegative while $V$ should be unrestricted. For example, a movie may be very bad so that everyone dislikes it, and hence the quality of this movie can be scored as $-1$. The confidence is the ability of a user to rate a movie, and so should not be negative. To explain it further, if the confidence of a user is also set as $-1$, then the product of -1 and -1 will be 1, which means that a user with low confidence rates a bad movie with a high score, which is not true in reality. On the contrary, the setting $U_i \in R_+^n$ avoids such unreasonable cases, leading to the advantage of the interpretability of $U$.

## 2.3 Sensitivity Analysis

We find $U$, $\Sigma$, and $V$ so that $P = U\Sigma V^T$ approximates $X$ well. But it is not preferable that small changes (due to computing errors or error propagated from observation errors in $X$) in these three matrices result in a big change in their product. Since the derivatives with respect to the variables $U$, $\Sigma$, and $V$ mean the change rate, we examine the square sum of the corresponding derivatives. Let the notation $||\cdot||_F$ denote the Frobenius norm.

By $\frac{\partial(BA)_{ij}}{\partial B_{mn}} = \delta_{im}(A)_{nj}$, we have

$$\begin{aligned}\sum_{ijmn}\left(\frac{\partial(U\Sigma V^T)_{ij}}{\partial U_{mk}}\right)^2 &= \sum_{ijmk}\left(\delta_{im}(\Sigma V^T)_{kj}\right)^2 \\ &= \sum_{ijk}\left((\Sigma V^T)_{kj}\right)^2 \\ &= n\sum_{jk}\left((\Sigma V^T)_{kj}\right)^2 \\ &= n||\Sigma V^T||_F^2. \end{aligned} \quad (1)$$

Similarly we have

$$\sum_{ijmn}\left(\frac{\partial(U\Sigma V^T)_{ij}}{\partial V_{mk}}\right)^2 = m||U\Sigma||_F^2, \quad (2)$$

$$\sum_{ijmn}\left(\frac{\partial(U\Sigma V^T)_{ij}}{\partial \Sigma_{mk}}\right)^2 = d||U||_F^2||V||_F^2. \quad (3)$$

## 2.4 Optimization Problem

Considering both the approximation $X \approx U\Sigma V^T$ and the sensitivity analysis, a factorization problem can be cast as an optimization problem.

$$\begin{aligned}\min_{U,\Sigma,V} & \sum_{(i,j)\in OI}(X_{ij} - (U\Sigma V^T)_{ij})^2 \\ + & \quad \lambda\left(n||\Sigma V^T||_F^2 + m||U\Sigma||_F^2 + d||U||_F^2||V||_F^2\right), \\ s.t. & \quad U \geq 0, \\ & \quad \Sigma \geq 0. \end{aligned} \quad (4)$$

where $\lambda$ is a hyperparameter that controls the balance between the approximation and the sensitivity, and $OI$ denote the set of observed index pairs.

## 2.5 Problem Simplification and Solution

Let $U_i$'s and $V_i$'s be the columns of $U$ and $V$ respectively. Without loss of generality, we set $||U_k||_F = 1, ||V_k||_F = 1$ for $1 \leq k \leq d$. As a result, $||U||_F^2 = d, ||V||_F^2 = d$. For the purpose of simplifying the solution, we further assume that $\Sigma_{d\times d}$ is diagonal, i.e., $\Sigma_{d\times d} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_d)$. Consequently,

$$||\Sigma V^T||_F^2 = \sum_{k=1}^{d}\lambda_k^2,$$

and

$$||U\Sigma||_F^2 = \sum_{k=1}^{d}\lambda_k^2.$$

In order to simplify the notation, we denote $U\Sigma$ as $U$, then $\Sigma$ disappears, and the conditions $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$ can be changed to $||U_1||_F \geq ||U_2||_F \geq \ldots \geq ||U_d||_F$. Based on the above simplification, Eq. (4) can be reformulated as follows.

Given an $n \times m$ nonnegative matrix $X$, solve

$$\begin{aligned} \min_{U_k, V_k} \quad & \sum_{(i,j)\in OI} \left( X_{ij} - \sum_{k=1}^{d} (U_k V_k^T)_{ij} \right)^2 \\ + \quad & \lambda n \sum_{k=1}^{d} ||U_k||_F^2 + \lambda m \sum_{k=1}^{d} ||V_k||_F^2 + \lambda d^3, \\ s.t. \quad & U_k \geq 0, \\ & ||U_k||_F \geq ||U_{k+1}||_F, \\ & ||V_k||_F = 1. \end{aligned} \tag{5}$$

In order to obtain the most informative latent features and find the dimension $d$, we fit the incomplete matrix $X$ step by step in such a way that when $U_k$ and $V_k$ are learned, $U_j$ $(j \leq k-1)$ and $V_j$ $(j \leq k-1)$ are fixed, and we only learn $U_k$ and $V_k$ based on the residual $R$. $R$ is defined as

$$R = X - \sum_{j=1}^{k-1} U_j V_j^T$$

on $OI$, and $R = 0$ on others for convenience. The process continues until there is no useful information retained in $R$. When the process stops, the dimension can be determined. So we only focus on the following problem:

$$\begin{aligned} \min_{U_k, V_k} \quad & \sum_{(i,j)\in OI} \left( R_{ij} - (U_k V_k^T)_{ij} \right)^2 \\ + \quad & \lambda n ||U_k||_F^2 + \lambda m ||V_k||_F^2, \\ s.t. \quad & U_k \geq 0, \\ & ||U_{k-1}||_F \geq ||U_k||_F, \\ & ||V_k||_F = 1. \end{aligned} \tag{6}$$

Note that the elements in $R$ may be negative. If we ignore the variant $\lambda_k$, the Lagrangian of the above problem is

$$\begin{aligned} J \quad = \quad & \sum_{(i,j)\in OI} \left( R_{ij} - (U_k V_k^T)_{ij} \right)^2 \\ + \quad & \lambda(m+n) ||U_k||_F^2 \\ + \quad & \mu_k (U_k^T U_k - U_{k-1}^T U_{k-1}) \\ + \quad & \nu_k (V_k^T V_k - 1) - Y^T U_K, \end{aligned} \tag{7}$$

where $Y \in R_+^n$, and $\mu_k \in R_+$. Let the $i$-th element of $U_k$, the $j$-th element of $V_k$, and the $i$-th element of $Y$ be $U_{ki}$, $V_{kj}$ and $Y_i$ respectively. In order to solve this problem, take derivative on $J$ with respect to $U_{ki}$ and $V_j$. We have

$$\begin{aligned} \frac{\partial J}{\partial U_{ki}} \quad = \quad & \sum_{j:(i,j)\in OI} 2(R_{ij} - U_{ki} V_{kj})(-V_{kj}) \\ + \quad & 2\mu_k U_{ki} - Y_i = 0, \end{aligned} \tag{8}$$

$$\begin{aligned} \frac{\partial J}{\partial V_{kj}} \quad = \quad & \sum_{i:(i,j)\in OI} 2(R_{ij} - U_{ki} V_{kj})(-U_{ki}) \\ + \quad & 2\nu_k V_{kj} = 0. \end{aligned} \tag{9}$$

If $U_k$ is given, then minimizing the quadratic function in Eq. (7), we obtain that

$$V_{kj} = \frac{\sum_{i:(i,j)\in OI} R_{ij} U_{ki}}{\sum_{i:(i,j)\in OI} U_{ki}^2 + \nu_k}, \tag{10}$$

where $\nu_k$ is a parameter such that $||V_k||_F = 1$.

If $V_k$ is given, considering the constraints that $U_k \geq 0$ and $||U_{k-1}||_F \geq ||U_k||_F$, we obtain

$$\begin{aligned} U_{ki} \quad &= \quad \frac{\sum_{j:(i,j)\in OI} R_{ij} V_{kj} + Y_i/2}{\sum_{j:(i,j)\in OI} V_{kj}^2 + \mu_k} \\ &= \quad \frac{(\sum_{j:(i,j)\in OI} R_{ij} V_{kj})_+}{\sum_{j:(i,j)\in OI} V_{kj}^2 + \mu_k}, \end{aligned} \tag{11}$$

where $Y_i$ is the minimum positive number such that

$$\sum_{j:(i,j)\in OI} R_{ij} V_{kj} + Y_i/2 \geq 0,$$

i.e.,

$$Y_i = 0 \text{ if } \sum_{j:(i,j)\in OI} R_{ij} V_{kj} \geq 0,$$

and

$$Y_i = -\sum_{j:(i,j)\in OI} R_{ij} V_{kj} \text{ if } \sum_{j:(i,j)\in OI} R_{ij} V_{kj} < 0,$$

and $\mu_k$ is the minimum positive number such that

$$||U_k||_F \leq ||U_{k-1}||_F .$$

We name our algorithm as Semi-Nonnegative Matrix Factorization with Global Statistical Consistency (SNGSC). In Algorithm 1, we summarize a learning algorithm by employing Eq. (10) and Eq. (11). The criterion that no useful information can be mined in $R$ is specified in our experiments as: the difference between the mean residual $\frac{1}{|OI|}\sum_{(i,j)\in OI} |R_{ij}|$ in the current dimension $d$ and that in the previous dimension is smaller than 0.0005.

From the algorithm, we can see the time complexity of SNGSC is linear on the number of ratings, i.e., O($|OI|$), because we only need to calculate the multiplications when the ratings values are not missing. Moreover, with the proper physical meaning in $U$ and $V$, our algorithm is expected to achieve more accurate results.

**Algorithm 1**: SNGSC Learning Algorithm

**Input**: Incomplete matrix $X \geq 0$
**Output**: $d$, $\{U_k\}_{k=1}^d$, and $\{V_k\}_{k=1}^d$

```
Initialize d = 0, k = 1.
repeat
  if k == 1 then
    R = X
  else
    R = R − U_{k−1} V_{k−1}^T
  end if
  repeat
    for j = 1 TO m do
      V_kj = (Σ_{i:(i,j)∈OI} R_ij U_ki) / (Σ_{i:(i,j)∈OI} U_ki^2 + ν_k)
    end for
    for i = 1 TO n do
      U_ki = ((Σ_{j:(i,j)∈OI} R_ij V_kj)_+) / (Σ_{j:(i,j)∈OI} V_kj^2 + μ_k)
    end for
  until Converge
  k = k + 1
until No useful information can be mined in R
d = k − 1
```

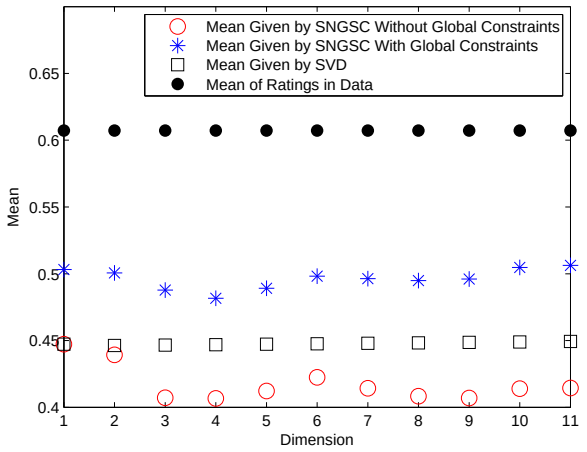


**Figure 1:** An illustration showing the problem of SNGSC and SVD without controlling the global statistics. The means predicted by models are far away from the true means.

# 3. CONSISTENCY WITH GLOBAL INFORMATION

Until now, we only constrain the expression $\sum_{k=1}^{d}(U_k V_k^T)$ in Eq. (5) by fitting its values on the user-item pairs with the training data. However, we observe that this partial constraint cannot make the values $\sum_{k=1}^{d}(U_k V_k^T)$ follow the global statistics such as the first moment and the second moment. The previous low-dimensional factor models share this problem because no action is taken on controlling the global statistics. For example, the mean of ratings in EachMovie Data is 0.607357 (after scaling to the interval [0,1]), but the mean given by SVD and SNGSC is far away from the true mean. In Figure 1, we demonstrate this problem.

Based on the above observation, we propose to impose the consistency on SNGSC between the predicted statistics and those given in the data samples. Ideally we should consider moments of all orders and the data priors, but considering the computation cost and the model complexity, we only include the first moment $\bar{X}$–the mean of ratings in this paper. The predicted values are given by $\sum_{k=1}^{d}(U_k V_k^T)$, and hence the predicted mean by the model is

$$\frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{d}(U_{ki}V_{kj}) = \sum_{k=1}^{d}(\bar{U}_k \bar{V}_k^T),$$

where $\bar{U}_k$ and $\bar{V}_k$ are the vector means of $U_k$ and $V_k$ respectively. Let $\eta$ be the parameter balancing the tradeoff of fitting the data and fitting the mean of ratings. Then we should optimize

$$\begin{aligned}
\min_{U_k,V_k} & \sum_{(i,j)\in OI}\left(R_{ij} - (U_k V_k^T)_{ij}\right)^2 \\
+ & \quad \lambda n||U_k||_F^2 + \lambda m||U_k||_F^2 \\
+ & \quad \eta\frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}(\sum_{l=1}^{k}U_{li}V_{lj} - \bar{X})^2, \\
s.t. & \quad U_k \geq 0, \\
& \quad ||U_{k-1}||_F \geq ||U_k||_F, \\
& \quad ||V_k||_F = 1. \qquad (12)
\end{aligned}$$

When $\eta = 0$, no global information is included; when $\eta = +\infty$, all the predicted values $\sum_{l=1}^{k}U_{li}V_{lj}$ will be equal to $\bar{X}$ such that the first moment is perfectly fitted. The best $\eta$ should be in the middle of these two extreme cases. In our experiments, we set $\eta = \sqrt{nm}/|OI|$ based on experiences. An ordinary calculus can result in similar equations as Eq. (10) and Eq. (11).

# 4. EXPERIMENTS

In this section, we conduct several experiments to compare the recommendation quality of our approach with other state-of-the-art collaborative filtering methods. Our experiments are intended to address the following questions:

1. How does our approach compare with the published state-of-the-art collaborative filtering algorithms?
2. How does the model parameter $\eta$ (the global consistency parameter) affect the accuracy of the prediction?
3. How do the non-negative constraints affect the accuracy of the recommendation quality?
4. What is the performance comparison on users with different observed ratings?

## 4.1 Description of Dataset

We evaluate our algorithms on the EachMovie dataset[5], which is commonly used in previous work [19, 21, 37]. The EachMovie dataset contains 74,424 users, 1,648 movies, and 2,811,718 ratings in the scale of zero to five. We map the ratings 0,1,2,3,4 and 5 to the interval $[0, 1]$ using the linear function $t(x) = x/5$.

As to the training data, we employ three settings: 80%, 50% and 20% for training, where 80% means we randomly select 80% ratings as training data to predict the remaining 20% ratings. Selecting 80% as training data is the standard evaluation setting which is widely employed in the previous work. However, in this paper, we are also interested in the settings to include 50% and 20% as training data, since these two settings can be used to examine how well the algorithms are under the sparse data settings. The reported results in all of the experiments in this paper are the average of ten runs of the algorithms on the ten random partitions of the dataset.

## 4.2 Metrics

We use the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) metrics to measure the prediction quality of our proposed approach in comparison with other collaborative filtering methods. MAE is defined as:

$$MAE = \frac{\sum_{i,j}|r_{i,j} - \widehat{r}_{i,j}|}{N}, \qquad (13)$$

where $r_{i,j}$ denotes the rating user $i$ gave to item $j$, $\widehat{r}_{i,j}$ denotes the rating user $i$ gave to item $j$ as predicted by our approach, and $N$ denotes the number of tested ratings. RMSE is defined as:

$$RMSE = \sqrt{\frac{\sum_{i,j}(r_{i,j} - \widehat{r}_{i,j})^2}{N}}. \qquad (14)$$

[5]http://www.research.digital.com/SRC/EachMovie/. It is retired by Hewlett-Packard (HP).

**Table 3:** Comparison with other popular algorithms. The reported values are the mean RMSE and MAE on the EachMovie Dataset achieved by ten runs from dividing the data into 80%, 50%, and 20% for training data, respectively.

| Dataset | Traning Data | Metrics | User Mean | Item Mean | MMMF | PMF | SNGSC |
|---|---|---|---|---|---|---|---|
| EachMovie | 80% | **RMSE** | 1.426 | 1.386 | 1.173 | 1.151 | **1.122** |
| | | Variance | $\leq 10^{-4}$ | $\leq 10^{-4}$ | $\leq 0.001$ | $\leq 0.001$ | $\leq 10^{-5}$ |
| | | **MAE** | 1.141 | 1.102 | 0.928 | 0.901 | **0.860** |
| | | Variance | $\leq 10^{-4}$ | $\leq 10^{-4}$ | $\leq$0.001 | $\leq$0.001 | $\leq 10^{-5}$ |
| | 50% | **RMSE** | 1.438 | 1.387 | 1.342 | 1.335 | **1.176** |
| | | Variance | $\leq 10^{-4}$ | $\leq 10^{-4}$ | $\leq 0.001$ | $\leq 0.001$ | $\leq 10^{-5}$ |
| | | **MAE** | 1.149 | 1.103 | 0.978 | 0.963 | **0.891** |
| | | Variance | $\leq 10^{-4}$ | $\leq 10^{-4}$ | $\leq$0.001 | $\leq$0.001 | $\leq 10^{-5}$ |
| | 20% | **RMSE** | 1.484 | 1.388 | 1.466 | 1.451 | **1.266** |
| | | Variance | $\leq 0.001$ | $\leq 0.001$ | $\leq 0.01$ | $\leq 0.01$ | $\leq 10^{-4}$ |
| | | **MAE** | 1.180 | 1.103 | 1.143 | 1.085 | **0.973** |
| | | Variance | $\leq 0.001$ | $\leq 0.001$ | $\leq$0.01 | $\leq$0.01 | $\leq 10^{-4}$ |

## 4.3 Performance Comparisons

We compare our SNGSC approach with other four approaches.

1. **User Mean**: This is a baseline method which predicts a user's missing rating on an item by the sample mean of this user's ratings.
2. **Item Mean**: This is a baseline method which predicts a user's missing rating on an item by the sample mean of this item's ratings.
3. **MMMF** [21, 31]: This method constrains the norms of $U$ and $V$ instead of their dimensionality. This corresponds to constraining the overall "strength" of the factors, rather than their number.
4. **PMF** [24]: This method proposes a probabilistic framework to employ $U_i^T V_j$ with Gaussian noise fitting each rating observation.

The prediction accuracies evaluated by Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) are shown in Table 3. In SNGSC, the parameter $\lambda$ is set to be 0.000004, and the parameter $\eta$ is set to be $\sqrt{nm}/|OI|$, where $|OI|$ is the number of observed ratings. The dimensions for SNGSC are automatically determined at each of the ten runs, and they are between 25 and 30. In order to compare other algorithms fairly, we set the dimensions of MMMF and PMF to 30.

From Table 3, we can observe that our algorithm consistently performs better than the other methods in all the settings. When we use a sparse dataset (20% as training data), we find that our method generates much better performance than MMMF and PMF. However, MMMF and PMF do not address the problem of sparsity, hence they even perform worse than the Item Mean method when using 20% as training data. This demonstrates the advantage of our algorithm in handling the sparsity problem.

In Figure 2 and Figure 3, we also plot the percentages of performance increase of our algorithm against other four methods in terms of RMSE and MAE on the EachMovie dataset, respectively. From these figures, we observe an interesting phenomenon: as the sparsity of the data increases, the percentages of performance increase against MMMF and PMF keep increasing. This observation again proves the advantage of our algorithm. On the other hand, we can also notice that as the sparsity increases, although our method still can generates much better recommendation qualities than User Mean and Item Mean methods, the percentages of performance increase against these two methods keep dropping. This observation is reasonable because our random testing data generation method does not change the distribution of the ratings. Hence, the User Mean and Item Mean algorithms should be relatively stable against the sparsity problem.

In order to show the usefulness of each key part of SNGSC, we also evaluate our algorithm on its various degraded cases as follows:

1. SNGSC-1: It is the SNGSC algorithm without the global consistency ($\eta = 0$);
2. SNGSC-2: It is the SNGSC algorithm without the nonnegative constraint (a modified version of SVD with global consistency);
3. SNGSC-3: It is the SNGSC algorithm with nonnegative constraints on both $U$ and $V$ (a modified version of NMF with global consistency).

The results on the EachMovie dataset are reported in Table 4. From the results, we observe that our Semi-Nonnegative setting is the best among all these variants, which empirically demonstrates the need of introducing SNMF.

However, the global consistency achieves only a little accuracy improvement in this experimental setting (See SNGSC-1 and SNGSC). This phenomenon may be caused by the setting that majority (80%) of data is chosen as training data. In the extreme case that the rating data is very sparse and each user only rates one movie, then the latent features $U$ and $V$ do not have much meanings, but we can at least predict all the missing ratings as the mean of training data. We believe that the sparser the training data, the better the global consistency approach. To demonstrate the effectiveness of the global consistency approach, we run both SNGSC-1 and SNGSC in a different setting: 20% of the data are chosen for training and 80% for testing. The results are shown in Table 5. From the results, we can see SNGSC

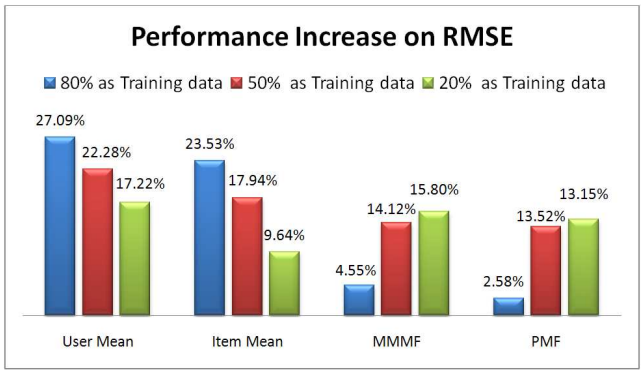


**Figure 2:** Performance Increase on RMSE (EachMovie)

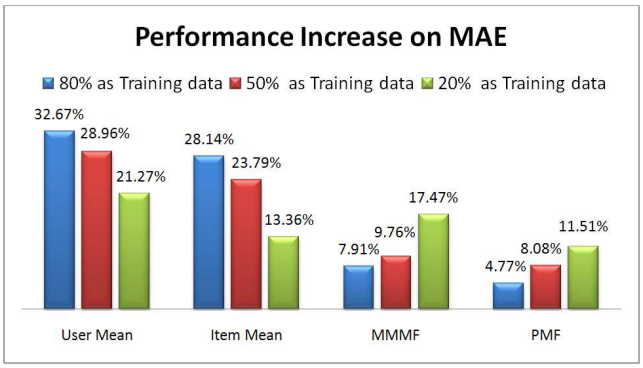


**Figure 3:** Performance Increase on MAE (EachMovie)

with the global consistency significantly outperforms the one without the global consistency (SNGSC-1). In such a setting, it is not surprising to see that the difference between SNGSC and SNGSC-2 is small, because the latent feature is not very meaningful and hence the sign setting is not so important; therefor, the global consistency dominates the results.

## 5. RELATED WORK

Recommender systems have been developed to automate the recommendation process [10]. Examples of research prototypes of recommender systems are PHOAKS [32], Syskills and Webert [20], Fab [1] and GroupLens [13, 26]. These systems recommend various types of Web resources, online news, movies, among others, to potentially interested parties [10]. They are becoming part of the standard e-business technology that can enhance e-commerce sales by converting browsers to buyers, increasing cross-selling, and building customer loyalty [27].

As stated in [10], one of the most commonly-used and successful recommendation approaches is the collaborative filtering approach [7, 22, 28]. In the field of collaborative filtering, two types of methods are widely studied: neighborhood-based approaches and model-based approaches.

The neighborhood-based approaches are well studied and successfully applied to lots of commercial recommender systems [14, 22]. The most analyzed examples of neighborhood-based collaborative filtering include user-based approaches [2, 6, 11] and item-based approaches [5, 14, 25]. User-based approaches predict the ratings of active users based on the ratings of similar users found, and item-based approaches predict the ratings of active users based on the computed information of items similar to those chosen by the active user. User-based and item-based approaches often use the PCC algorithm [22] and the VSS algorithm [2] as the similarity computation methods. PCC-based collaborative filtering can generally achieve higher performance than the other popular algorithm VSS, since it considers the differences of user rating style. Another set of related work considers how to employ the user-based and item-based approaches together [16]. Ma et al. [16] presented a method to employ both user information and item information to firstly fill some missing values before making predictions for active users. As mentioned in Section 1, despite the success in the industry, almost most of the neighborhood-based methods suffer from the data sparsity and scalability problems.

In contrast to the neighborhood-based approaches, the model-based approaches to collaborative filtering use the observed user-item ratings to train a compact model that explains the given data, so that ratings could be predicted via the model instead of directly manipulating the original rating database as the neighborhood-based approaches do [15]. Algorithms in this category include the aspect models [8, 9, 29] and the latent factor model [3]. [12] presented an algorithm for collaborative filtering based on hierarchical clustering, which tried to balance both robustness and accuracy of predictions, especially when few data were available. [8] proposed an algorithm based on a generalization of probabilistic latent semantic analysis to continuous-valued response variables.

Recently, due to the efficiency in dealing with large datasets, several low-dimensional matrix approximation methods [21, 23, 24, 30] have been proposed for collaborative filtering. These methods focus on fitting a factor model to the data, and use it in order to make further predictions.

Low-rank matrix approximations based on minimizing the sum-squared errors can be easily solved using Singular Value Decomposition (SVD), and a simple and efficient Expectation Maximization (EM) algorithm for solving weighted low-rank approximation is proposed in [30]. In [31], Srebro et al. proposed a matrix factorization method to constrain the norms of $U$ and $V$ instead of their dimensionality. Salakhutdinov et al. presented a probabilistic linear model with Gaussian observation noise in [24]. In [23], the Gaussian-Wishart priors are placed on the user and item hyperparameters. Although low-dimensional methods are proved to be very effective and efficient, these methods still suffer several disadvantages that are unveiled. In the SVD method, as well as other well-known methods such as the weighted low-rank approximation method [30], Probabilistic Principal Component Analysis (PPCA) [33], Probabilistic Matrix Factorization (PMF) [24] and Constrained Probabilistic Matrix Factorization [24], the latent features are uninterpretable, and there is no range constraint bound on the latent features vectors. The lack of interpretability results in the improper modeling of the latent factors, hence downgrades the recommendation accuracy. In [34], a non-negative constraint is imposed on both user-specific features $U$ and item-specific features $V$ (Nonnegative Matrix Factorization), but this work is also unable to interpret the physical meanings of the latent factors. Furthermore, the low-rank approximation methods also suffer from the data sparsity problem. Hence, in this paper, we propose a novel

**Table 4:** Comparison with variants of SNGSC in a setting with 80% for training and 20% for testing on the EachMovie dataset. (1) SNGSC-1: SNGSC without the global consistency ($\eta = 0$); (2) SNGSC-2: SNGSC without the nonnegative constraint (a modified version of SVD with global consistency); and (3) SNGSC-3: SNGSC with nonnegative constraints on both $U$ and $V$ (a modified version of NMF with global consistency).

| Algorithms | SNGSC-1 | SNGSC-2 | SNGSC-3 | SNGSC |
|---|---|---|---|---|
| **RMSE** | 1.151 | 1.212 | 1.258 | 1.122 |
| Variance | $\leq 10^{-5}$ | $\leq$0.001 | $\leq$0.001 | $\leq 10^{-5}$ |
| **MAE** | 0.883 | 0.932 | 0.971 | 0.860 |
| Variance | $\leq 10^{-5}$ | $\leq$0.001 | $\leq$0.001 | $\leq 10^{-5}$ |

**Table 5:** Comparison with variants of SNGSC in a 20% for training 80% for testing setting on the EachMovie dataset.

| Algorithms | SNGSC-1 | SNGSC-2 | SNGSC-3 | SNGSC |
|---|---|---|---|---|
| **RMSE** | 1.423 | 1.356 | 1.365 | 1.266 |
| Variance | $\leq 10^{-4}$ | $\leq$0.01 | $\leq$0.01 | $\leq 10^{-4}$ |
| **MAE** | 1.095 | 1.048 | 1.060 | 0.973 |
| Variance | $\leq 10^{-4}$ | $\leq$0.01 | $\leq$0.01 | $\leq 10^{-4}$ |

matrix factorization method to solve the analyzed problems and remedy the aforementioned deficiencies.

## 6. CONCLUSIONS AND FUTURE WORK

We demonstrate a Semi-Nonnegative Matrix Factorization method with Global Statistical Consistency for collaborative filtering, in which the user-specific latent feature $U_{ik}$ includes the meaning of the confidence of user $i$ on the $k$-th latent type of the item, and the item-specific latent feature $V_{jk}$ includes the meaning of the quality of the item $j$ on the $k$-th latent type of the item. This work has showed that the latent features with physical meanings can achieve not only the model interpretability but also the prediction accuracy. Moreover, we propose a novel method that imposes the consistency between the statistics of training data and the statistics of the predicted ratings. The experimental analysis shows that our method outperforms other state-of-the-art algorithms.

For the global consistency, we only take the first step, i.e., we only make our models consistent with the first moment currently. By doing so we have already achieved promising results. In order to capitalize on these achievements, further study is needed on the following problems:

1. We would enforce the consistency with the second moment globally in the models without increasing the complexity of our models.

2. There is prior information that all values in the matrix $\sum_{k=1}^{d}(U_k V_k^T)$ should be between zero and one after the mapping. Without taking any action, prediction by $\sum_{k=1}^{d} U_k V_k^T$ will run outside of the range of valid rating values. For this, one choice is to map the values to the interval $[0, 1]$ by some nonlinear functions like logistic function. But in our setting, such a mapping does not match our intuition–the prediction on the user-item pair $(i, j)$ results from a linear combination of the products of $i$'s authority on a latent type and $j$'s quality. For such a consideration, how can we put a constraint that $0 \leq \sum_{k=1}^{d}(U_k V_k^T) \leq 1$ while we can still learn the latent features dimension by dimension.

## 7. REFERENCES


[1] M. Balabanović and Y. Shoham. Fab: content-based, collaborative recommendation. *Commun. ACM*, 40(3):66–72, 1997.

[2] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 43–52, San Francisco, 1998. Morgan Kaufmann.

[3] J. Canny. Collaborative filtering with privacy via factor analysis. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 238–245, 2002.

[4] B. Cao, J.-T. Sun, J. Wu, Q. Yang, and Z. Chen. Learning bidirectional similarity for collaborative filtering. In *ECML PKDD '08: Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases - Part I*, pages 178–194, Berlin, Heidelberg, 2008. Springer-Verlag.

[5] M. Deshpande and G. Karypis. Item-based top-n recommendation. *ACM Transactions on Information Systems*, 22(1):143–177, 2004.

[6] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237, 1999.

[7] W. Hill, L. Stead, M. Rosenstein, and G. Furnas. Recommending and evaluating choices in a virtual community of use. In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 194–201, 1995.

[8] T. Hofmann. Collaborative filtering via gaussian probabilistic latent semantic analysis. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 259–266, 2003.

[9] T. Hofmann. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems*, 22(1):89–115, 2004.

[10] Z. Huang, H. Chen, and D. Zeng. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Trans. Inf. Syst.*, 22(1):116–142, 2004.

[11] R. Jin, J. Y. Chai, and L. Si. An automatic weighting scheme for collaborative filtering. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 337–344, 2004.

[12] A. Kohrs and B. Merialdo. Clustering for collaborative filtering applications. In *Proceedings of CIMCA*, 1999.

[13] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. Grouplens: applying collaborative filtering to usenet news. *Commun. ACM*, 40(3):77–87, 1997.

[14] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, pages 76–80, Jan/Feb 2003.

[15] N. N. Liu and Q. Yang. Eigenrank: a ranking-oriented approach to collaborative filtering. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 83–90, 2008.

[16] H. Ma, I. King, and M. R. Lyu. Effective missing data prediction for collaborative filtering. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 39–46, 2007.

[17] H. Ma, I. King, and M. R. Lyu. Learning to recommend with social trust ensemble. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 203–210, 2009.

[18] H. Ma, H. Yang, M. R. Lyu, and I. King. SoRec: social recommendation using probabilistic matrix factorization. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 931–940, 2008.

[19] B. Marlin. Modeling user rating profiles for collaborative filtering. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[20] M. Pazzani and D. Billsus. Learning and revising user profiles: The identification ofinteresting web sites. *Mach. Learn.*, 27(3):313–331, 1997.

[21] J. D. M. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 713–719, 2005.

[22] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of ACM Conference on Computer Supported Cooperative Work*, 1994.

[23] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *Proceedings of the International Conference on Machine Learning*, volume 25, 2008.

[24] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1257–1264. MIT Press, Cambridge, MA, 2008.

[25] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 285–295, 2001.

[26] B. M. Sarwar, J. A. Konstan, A. Borchers, J. Herlocker, B. Miller, and J. Riedl. Using filtering agents to improve prediction quality in the grouplens research collaborative filtering system. In *CSCW '98: Proceedings of the 1998 ACM conference on Computer supported cooperative work*, pages 345–354, 1998.

[27] J. B. Schafer, J. A. Konstan, and J. Riedl. E-commerce recommendation applications. *Data Min. Knowl. Discov.*, 5(1-2):115–153, 2001.

[28] U. Shardanand and P. Maes. Social information filtering: algorithms for automating "word of mouth". In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 210–217, 1995.

[29] L. Si and R. Jin. Flexible mixture model for collaborative filtering. In *ICML '03: Proceedings of the 20th International Conference on Machine Learning*, 2003.

[30] N. Srebro and T. Jaakkola. Weighted low-rank approximations. In T. Fawcett and N. Mishra, editors, *ICML*, pages 720–727. AAAI Press, 2003.

[31] N. Srebro, J. D. M. Rennie, and T. Jaakkola. Maximum-margin matrix factorization. In *NIPS*, 2004.

[32] L. Terveen, W. Hill, B. Amento, D. McDonald, and J. Creter. Phoaks: a system for sharing recommendations. *Commun. ACM*, 40(3):59–62, 1997.

[33] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[34] S. Zhang, W. Wang, J. Ford, and F. Makedon. Learning from incomplete ratings using non-negative matrix factorization. In J. Ghosh, D. Lambert, D. B. Skillicorn, and J. Srivastava, editors, *SDM*. SIAM, 2006.

[35] Y. Zhang and J. Koren. Efficient bayesian hierarchical user modeling for recommendation system. In *Proc. of SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 47–54, 2007.

[36] D. Zhou, S. Zhu, K. Yu, X. Song, B. L. Tseng, H. Zha, and C. L. Giles. Learning multiple graphs for document recommendations. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 141–150, 2008.

[37] S. Zhu, K. Yu, and Y. Gong. Predictive matrix-variate t models. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1721–1728. MIT Press, Cambridge, MA, 2008.