Incremental Learning for Dynamic Collaborative Filtering

Shuli Han[1], Yujiu Yang[2] and Wenhuang Liu[3]
Graduate School at Shenzhen, Tsinghua Universtity, Shenzhen, P.R.China
Email: [1]hansl08@mails.tsinghua.edu.cn
[2,3]{yang.yujiu,liuwh}@sz.tsinghua.edu.cn

Abstract—**Collaborative Filtering (CF) is one of the widely used methods for recommendation problem. The key idea is to predict further the interests of a user (ratings) based on the available rating information from many users. Recently, matrix factorization (MF) based approaches, one branch of collaborative filtering, have proven successful for the rating prediction issues. However, most of the state-of-the-art MF models share the same drawback that the established models are static. They are only capable of handling CF systems with static settings, but never practical for a real-world system, which involves dynamic scenarios like new user signing in, new item being added and new rating being given now and then. For conventional MF models, they have to conduct repetitive learning every time dynamic scenario occurs. It is computational expensive and hard to meet the real-time demand. Therefore, an incremental learning framework based on Weighted NMF is proposed. To reduce the computational cost, it utilizes partially the optimization information from the original system, and stores some corresponding information for the subsequent incremental model. Our empirical studies show that the IWNMF scheme for different dynamic scenarios greatly lower the computational cost without degrading the prediction accuracy.**

Index Terms—**Dynamic Collaborative Filtering, Weighted Nonnegative Matrix Factorization, Incremental Learning**

I. Introduction

The introducing of recommendation technologies is changing the way people use Internet. The recommend progress can dominate the user's browse order and lead users discover what they like. It helps users obtain preferred information from huge collections of information resources. Recommendation Systems can now be found in many Internet application domains and has brought huge commercial interest. Examples include: 38% of Google News' more click-through are due to recommendation, 2/3 of Netflix's rented movies are from recommendation, and 35% of Amazon's sales are from recommendation, and so on [1]. Among all, the most typical and influential example in both academic and industrial communities should be the Netflix Prize competition [2], which greatly promotes the development of recommendation technology.

What recommendation system does is to learn preferences of users based on history purchase record or rating data for some items, then to accurately predict which other items users might prefer, and finally provide users with a list of recommended items. A very common task for recommendation is to predict ratings (e.g. one star through five stars) over a set of items. It is an implicit act of recommendation. Lots of work has been done to find more appropriate algorithms for this task [3] [4].

Matrix Factorization models have become popular recently and have provided good performance in recommendation system. The main idea behind such models is that the taste of a user could be represented by a number of latent factors and by applying factorization, the original rating matrix is mapped into a latent space in which objects could have a compact representation. Several matrix factorization models have been proposed for collaborative filtering [4] [5] [6] [7] [8]. They either incorporate a positive regularization item or a nonnegative constraint to enhance the linear factor model. However, all these models share the same drawback that the established models are static. They are designed for the case that users, items and ratings are fixed, therefore can't respond adaptively while the system setting varies. Every time new users, items or ratings enter the system, they have to conduct repetitive learning. For an online system involves hundreds of thousands of users and items, it is always time-consuming to train a prediction model. Relearn the matrix factorization model based on the whole new data is unsuitable for real-time collaborative filtering application. A dynamic collaborative filtering technique is crucial.

In this paper, we consider an incremental learning framework for dynamic collaborative filtering, which is based on one of the weighted-based model, Weighted Nonnegative Matrix Factorization(WNMF) [7]. The basic idea is to keep part of the parameters from the original model available, and update other parameters from the new information. We refer to our model as Incremental Weighted Nonnegative Matrix Factorization(IWNMF).

The proposed model is inspired by some incremental techniques applied in face recognition and dynamic background modeling [9] [10] [11]. All these schemes try to avoid relearning the whole model parameters and deduce computational cost by introducing incremental learning. Differing from the aforementioned models, we exploit how to use the incremental techniques directly for

doi:10.4304/jsw.6.6.969-976

a dynamic collaborative system. Meanwhile, some mathematical tricks are provided to further reduce the calculated amount.

The rest of this paper is organized as follows. In Section 2, briefly review of the conventional WNMF model is presented. The IWNMF framework and theoretical analysis for different dynamic scenarios are given in Section 3. In Section 4, the results of the experiments are demonstrated. In Section 5, an overview of state-of-the-art methods for collaborative filtering is provided. Finally, a conclusion is drawn in Section 6.

II. Weighted NMF Overview

A. Recommender System

Suppose that there are m users and n items in the recommendation system. There exists between every user-item pair a preference measurement which could be quantified as a specific value, referred to as rating. Ratings are a set of discrete numbers coming from $\{1, 2, \cdots, R\}$. All these $m \times n$ ratings form a complete matrix $\bar{A}$, each element of which shows how much a user likes or dislikes an item. However, in reality application $\bar{A}$ is partially observed, i.e. we can only obtain a matrix A with a small part of its elements known. The task of recommendation then becomes learning a prediction models based on the observed elements, so that the fill in matrix could approximate the complete matrix $\bar{A}$ as close as possible [12]. Fig.1 example is the toy example on the problem we discuss.

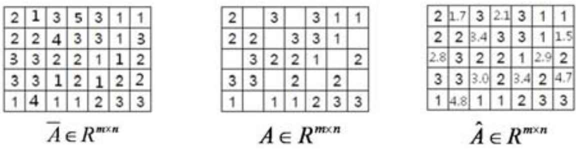
This user preference prediction task can be formulated as a matrix factorization problem. Namely, for a given user-item rating matrix A, we need to find two matrixes $U \in R^{m\times k}$ and $V \in R^{k\times n}$, so that $A \sim UV$, where parameter k represents the number of latent features reflect preference, each row of U shows how a user appreciate each factor, and each row of V indicates that to what extent an item possess each feature. Then rating on item j by user i can be formulated as:

$$\hat{r} = \sum_{t=1}^{k} U_{it} V_{tj} \ . \tag{1}$$

B. WNMF Framework

For the rating matrix is incomplete and sparse in the application of recommender system, a zero/one weighting factor is introduced to NMF [7]. Weighted nonnegative matrix factorization (WNMF) temps to find a linear matrix that maximizes the log-likelihood of the observed data. The optimization model could be described as follows:

$$\begin{aligned} \min \ & \sum_{ij} W_{ij} (A_{ij} - (UV)_{ij})^2 \\ s.t. \ & U, V \geq 0 \qquad , \\ & U \in R^{m\times k}, V \in R^{k\times n} \end{aligned} \tag{2}$$



2	1	3	5	3	1	1
2	2	4	3	3	1	3
3	3	2	2	1	1	2
3	3	1	2	1	2	2
1	4	1	1	2	3	3

$\bar{A} \in R^{m\times n}$

2		3		3	1	1
2	2		3	3	1	
	3	2	2	1		2
3	3		2		2	
1		1	1	2	3	3

$A \in R^{m\times n}$

2	1.7	3	2.1	3	1	1
2	2	3.4	3	3	1	1.5
2.8	3	2	2	1	2.9	2
3	3	3.0	2	3.4	2	4.7
1	4.8	1	1	2	3	3

$\hat{A} \in R^{m\times n}$

Figure 1. Toy example: $\bar{A}$ denotes the complete matrix; A represents the partially observed matrix, missing elements of which are set to 0; $\hat{A}$ is the prediction matrix.

where

$$W_{ij} = \begin{cases} 1, \ if \ A_{ij} \ is \ observed \\ 0, \ if \ A_{ij} \ is \ unknown \end{cases} .$$

The following updating rules can be derived by applying Lagrange Multiplier and Karush-Kuhn-Tucker complementary condition:

$$u_{ij}^{(t+1)} = u_{ij}^{(t)} \frac{((A \odot W)V^T)_{ij}}{(((UV) \odot W)V^T)_{ij}} \ , \tag{3}$$

$$v_{ij}^{(t+1)} = v_{ij}^{(t)} \frac{(U^T (A \odot W))_{ij}}{(U^T ((UV) \odot W))_{ij}} \ . \tag{4}$$

It is proven by [13] that the WNMF algorithm is guaranteed to converge. According to the updating rules (3) and (4), the computational complexity for WNMF is $O(mn + mnk)$, i.e. $O(mnk)$.

III. Incremental Weighted Nonnegative Matrix Factorization

In this section, we will present how online updates could be applied to deal with the dynamic changes of the system without having to relearn the whole model. The solution of new user, new item and new rating problem will be discussed in III-B.1, III-B.2 and III-B.3 respectively.

A. Problem Description & Notations

From the above discussion, we can tell that the aim of collaborative filtering system is to learn the optimized features for both users and items, and then make the recommendation. To be concise and to the point, the recommender progress for both static and dynamic collaborative filtering are illustrates in Fig.2, where <User, Item, rating> stands for the original triplets, and Δ <User, Item, rating> stands for the increments. Let's discuss the difference between WNMF and IWNMF framework under a dynamic case. Figure in the middle shows that the conventional WNMF framework simply integrates the increments into the original <User, Item, rating> triplets, and then rerun the iterative updates, which utilizes none of the features information from the original factorization. As to IWNMF framework for new entries (See the figure at the bottom), we can see the increments act on the result of the original model directly. It makes use of the old data, therefore could lower the computational cost.

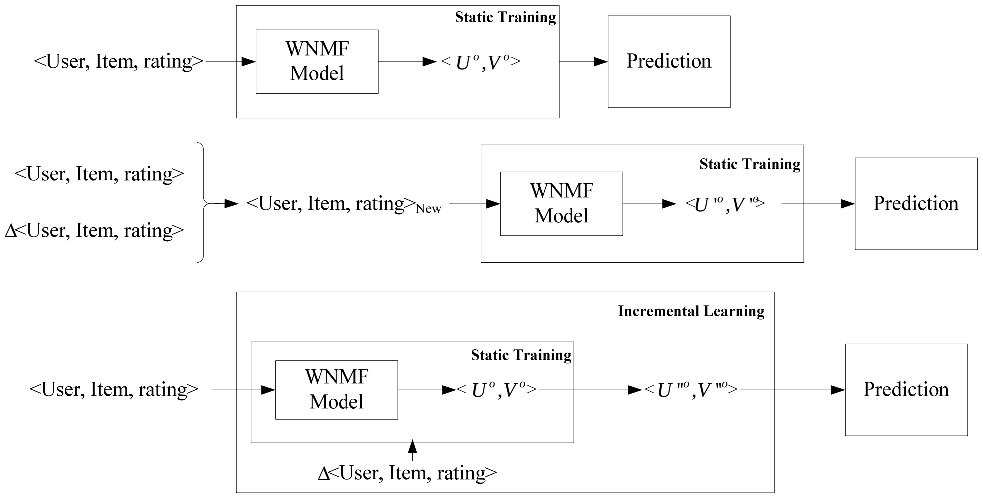


Figure 2. Diagrams for both Static and Dynamic Collaborative Filtering: the top diagram illustrates the WNMF framework for static environment; the middle diagram shows the WNMF for dynamic scenario; the bottom demonstrates the IWNMF for dynamic scenario.

For convenience, we unify the notations that used in the following sections in Table I.

B. IWNMF Models

Before we start our discussion, we need to declare a prerequisite, which is: For the original static system, we already get $A \sim U^o V^o$, where $A \in R^{m\times k}$, $U^o \in R^{m\times k}$ and $V^o \in R^{k\times n}$. That is to say,

$$\begin{aligned} J(U^o, V^o) &= \arg \min_{(U,V)} J(U,V) \\ &= \arg \min_{(U,V)} \sum_{i=1}^{m}\sum_{j=1}^{n}(W_{ij}(A_{ij}-(UV)_{ij}))^2 \end{aligned} .$$

1) New User enters

When a new user u_{m+1} with a few ratings signs in, the new rating matrix becomes:

$$A^U = \begin{bmatrix} A \\ x_{m+1}^T \end{bmatrix}.$$

where x_{m+1}^T is a row vector that records the rating information of u_{m+1} , and the corresponding weighting vector is w_{m+1}^T .

The objective function for the dynamic system is:

$$\begin{aligned} J(U,V) &= \sum_{i=1}^{m+1}\sum_{j=1}^{n} W_{ij}^U (A_{ij}^U - (UV)_{ij})^2 \\ &= \sum_{i=1}^{m}\sum_{j=1}^{n} W_{ij}(A_{ij} - (U_m V)_{ij})^2 \\ &+ \sum_{j=1}^{n}(w_{m+1}^T)_j((x_{m+1}^T)_j - (u_{m+1}^T V)_j)^2 \end{aligned} . \quad (5)$$

Considering the case that the number of ratings a new user owes is far less than that of items in the system. Therefore, the entries of the new user don't have the power to significantly affect the distribution of the other users in the feature space. We can assume that the feature representation of the original users in the dynamic system is approximately equal to that in the static original system, so that U_m in (5) can be set to U^o in advance. Then the objective function becomes:

TABLE I.
NOTATIONS

Notation	Description
$J(U,V)$	the objective function of the optimization model for CF
A	the original partially observed rating matrix
A^U	partially observed rating matrix with a new user sigh in
A^I	partially observed rating matrix with a new item added
A^R	partially observed rating matrix with a new rating given
W	weighting matrix correspond to A
W^U	weighting matrix correspond to A^U
W^I	weighting matrix correspond to A^I
W^R	weighting matrix correspond to A^R
U^o	the optimized User feature matrix given A
V^o	the optimized Item feature matrix given A
U_m	feature matrix for the original m users
V_n	feature matrix for the original n items
u_{m+1}^T	feature vector of user $m+1$
v_{n+1}	feature vector of item $n+1$

$$\begin{aligned} J(u_{m+1}, V) &= \sum_{i=1}^{m}\sum_{j=1}^{n} W_{ij}(A_{ij} - (U^o V)_{ij})^2 \\ &+ \sum_{j=1}^{n}(w_{m+1}^T)_j((x_{m+1}^T)_j - (u_{m+1}^T V)_j)^2 \end{aligned} . \quad (6)$$

Perform the Lagrange Multiplier method and using the Karush-Kuhn-Tucker complementary condition, we can get the (7):

$$\frac{\partial J}{\partial u_{m+1}} = -V(x_{m+1} \odot w_{m+1}) + V((V^T u_{m+1}) \odot w_{m+1})$$

$$\frac{\partial J}{\partial V} = -((U^o)^T A + u_{m+1} x_{m+1}^T) + ((U^o)^T ((U^o V) \odot W) \; . \quad (7)$$
$$+ u_{m+1} ((u_{m+1}^T V) \odot w_{m+1}^T)$$

Then the multiplicative updates can be derived as follow:

$$\left(u_{m+1}\right)_a \leftarrow \left(u_{m+1}\right)_a \frac{\left(V\left(x_{m+1} \odot w_{m+1}\right)\right)_a}{\left(V\left(\left(V^T u_{m+1}\right) \odot w_{m+1}\right)\right)_a} , \quad (8)$$

$$v_{ij} \leftarrow v_{ij} \frac{((U^o)^T A + u_{m+1} x_{m+1}^T)_{ij}}{((U^o)^T ((U^o V) \odot W) + u_{m+1} (u_{m+1}^T V) \odot w_{m+1}^T)_{ij}} .(9)$$

Although the IWNMF model under updating rule (8) doesn't need to relearn the features of the original users, the updating rule (9) for calculating V still has a computational complexity of $O(mnk)$. The whole IWNMF model shares the same complexity with WNMF, which is unacceptable. We try to apply some mathematical tricks to handle this issue.

The updating rule for calculating V in WNMF can be rewritten as:

$$v_{ij} \leftarrow v_{ij} \frac{(U^T (A \odot W))_{ij}}{(U^T ((UV) \odot W))_{ij}} , \quad (10)$$

$$v_{ij} \leftarrow v_{ij} \frac{(U^T A)_{ij}}{((U_{(:,i)} \odot W_{(:,j)})^T U) V_{(:,j)}} . \quad (11)$$
$$(W \textit{ is a binary matrix})$$

The updating rule for calculating V in IWNMF can be rewritten as:

$$v_{ij} \leftarrow v_{ij} \frac{((U^o)^T A + u_{m+1} x_{m+1}^T)_{ij}}{((U^o)^T (U^o V \odot W) + u_{m+1} ((u_{m+1}^T V) \odot w_{m+1}^T)_{ij}} ,(12)$$

$$v_{ij} \leftarrow v_{ij} \frac{((U^o)^T A)_{ij} + (u_{m+1} x_{m+1}^T)_{ij}}{((U_{(:,i)}^o \odot \; W_{(:,j)})^T U^o + u_{m+1}^T (u_{i,1} w_{j,1})) V_{(:,j)}} .(13)$$

We can observed that $(U^o)^T A$ and $(U_{(:,i)}^o \odot W_{(:,j)})^T U^o$ in (12) are fixed numbers during the interactive progress; once computed, they could be stored and don't need to compute again. Furthermore, these two factors could be obtained directly from the original static WNMF model (see (10)), they even don't need to recalculate in the IWNMF model. Hence, the computational complexity of IWNMF for new user problem under (8) and (12) is $O(nk + nk^2)$, i.e. $O(nk^2)$.

2) *New Item enters*

When a new item v_{n+1} with a few ratings signs in, the new rating matrix becomes:

$$A^I = \left[A \quad x_{n+1}\right] ,$$

where x_{n+1} records the rating information of v_{n+1}, and the corresponding weighting vector is w_{n+1}.

The objective function for the dynamic system is:

$$J(U,V) = \sum_{i=1}^{m} \sum_{j=1}^{n+1} W_{ij}^I (A_{ij}^I - (UV)_{ij})^2$$
$$= \sum_{i=1}^{m} \sum_{j=1}^{n} W_{ij} (A_{ij} - (UV_n)_{ij})^2 \quad . \quad (14)$$
$$+ \sum_{i=1}^{m} (w_{n+1})_i ((x_{n+1})_i - (Uv_{n+1})_i)^2$$

Symmetrically, new item problem is similar to new user problem. The entries of a new item don't have the power to significantly affect the distribution of the other items in the feature space. Hence, the objective function becomes:

$$J(U, v_{n+1}) = \sum_{i=1}^{m} \sum_{j=1}^{n} W_{ij} (A_{ij} - (UV^o)_{ij})^2$$
$$+ \sum_{i=1}^{m} (w_{n+1})_i ((x_{n+1})_i - (Uv_{n+1})_i)^2 \quad . \quad (15)$$

Perform the Lagrange Multiplier method and using the Karush-Kuhn-Tucker complementary condition, we can get the multiplicative update rules as follows:

$$u_{ij} \leftarrow u_{ij} \frac{(A(V^o)^T + x_{n+1} v_{n+1}^T)_{ij}}{(((UV^o) \odot W)(V^o)^T + ((Uv_{n+1}) \odot w_{n+1}) u_{n+1}^T)_{ij}} , (16)$$

$$\left(v_{n+1}\right)_a \leftarrow \left(v_{n+1}\right)_a \frac{\left(U^T \left(x_{n+1} \odot w_{n+1}\right)\right)_a}{\left(U^T \left(\left(Uv_{n+1}\right) \odot w_{n+1}\right)\right)_a} . \quad (17)$$

To further reduce the complexity complexity, we also utilize some mathematical tricks to rewrite the updating rules. The updating rule (3) for calculating U in WNMF can be rewritten as:

$$u_{ij} \leftarrow u_{ij} \frac{((A \odot W) V^T)_{ij}}{(((UV) \odot W) V^T)_{ij}} , \quad (18)$$

$$u_{ij} \leftarrow u_{ij} \frac{AV_{ij}^T}{U_{(i,:)} (V (V_{(j,:)} \odot W_{(j,:)})^T} . \quad (19)$$
$$(W \textit{ is a binary matrix})$$

The updating rule (16) for calculating U in IWNMF can be rewritten as:

$$u_{ij} \leftarrow u_{ij} \frac{(A(V^o)^T + x_{n+1} v_{n+1}^T)_{ij}}{(((UV^o) \odot W)(V^o)^T + (Uv_{n+1}) \odot w_{n+1} v_{n+1}^T)_{ij}} , \; (20)$$

$$u_{ij} \leftarrow u_{ij} \frac{(A(V^o)^T + x_{n+1} v_{n+1}^T)_{ij}}{U_{(i,:)} (V^o (V_{(j,:)}^o \odot W_{(i,:)})^T + v_{n+1} (v_{j,1} w_{i,1}))} .(21)$$

In (20), $A(V^o)^T$ and $V^o(V^o_{(j,:)} \odot W_{(i,:)})^T$ are two constant factors, and can be obtained from the original WNMF model (see (18)). Hence, the computational complexity of IWNMF for new item problem under (20) and (17) is $O(mk + mk^2)$, i.e. $O(mk^2)$.

3) New Rating enters

Without loss of generality, assume that the new rating is given by user m on item n. In order to express the problem well, we partitioned the corresponding matrices. The block form of the new rating matrix A^R and the original rating matrix A in (22) shows the connection between the two matrices.

$$A^R = \begin{bmatrix} A' & x_n \\ x_m^T & x_{mn} \end{bmatrix}, \; A = \begin{bmatrix} A' & x_n \\ x_m^T & 0 \end{bmatrix}, \qquad (22)$$

where

$x_m^T \in R^{(n-1)\times 1}$, $x_n \in R^{(m-1)\times 1}$ are two rating vectors.

w_m^T, w_n are the corresponding weighting vectors.

x_{mn} is the newly graded rating value.

Then the objective function for new rating problem is:

$$\begin{aligned} J(U,V) &= \sum_{i=1}^{m}\sum_{j=1}^{n} W_{ij}^R (A_{ij}^R - (UV)_{ij})^2 \\ &= \sum_{i=1}^{m-1}\sum_{j=1}^{n-1} W'_{ij}(A'_{ij} - (U'V')_{ij})^2 \\ &+ \sum_{i=1}^{m-1} (w_n)_i ((x_n)_i - (U'v_n)_i)^2 \\ &+ \sum_{j=1}^{n-1} (w_m^T)_j ((x_m^T)_j - (u_m^T V')_i)^2 \\ &+ (x_{mn} - u_m^T v_n)^2 \end{aligned} \quad . \qquad (23)$$

Considering the case that a new item doesn't have sufficient power to significantly affect the relationship between different users and items. Therefore, we assume that this new rating only affect the feature representation of user m and item n. In such assumption, U' is equal to the first $(m-1)$ rows of U^o, and V' is equal to the first $(n-1)$ columns of V^o. Hence, U' and V' in (23) are fixed matrices.

Then the objective function becomes:

$$\begin{aligned} J(u_m, v_n) &= \sum_{i=1}^{m-1}\sum_{j=1}^{n-1} W'_{ij}(A'_{ij} - (U'V')_{ij})^2 \\ &+ \sum_{i=1}^{m-1} (w_n)_i ((x_n)_i - (U'v_n)_i)^2 \\ &+ \sum_{j=1}^{n-1} (w_m^T)_j ((x_m^T)_j - (u_m^T V')_i)^2 \\ &+ (x_{mn} - u_m^T v_n)^2 \end{aligned} \quad . \qquad (24)$$

Perform the Lagrange Multiplier method and using the Karush-Kuhn-Tucker complementary condition, we can get the multiplicative update rules as follows:

TABLE II.
COMPUTATIONAL COMPLEXITY

Models	Computational Complexity
WNMF	$O(mnk)$
IWNMF for new user	$O(nk^2)$
IWNMF for new item	$O(mk^2)$
IWNMF for new rating	$O(k(m+n-2))$

TABLE III.
DATASET DESCRIPTION

	MovieLens
Number of Users	943
Number of Products	1,682
Number of Ratings	100,000
The scale of Ratings	1 ~ 5
Density(%)	6.3

$$(u_m)_a \leftarrow (u_m)_a \frac{(V'(x_m \odot w_m) + x_{mn} v_n)_a}{(V'((u_m V')^T \odot w_m) + (u_m^T v_n) v_n)_a} ,(25)$$

$$(v_n)_a \leftarrow (v_n)_a \frac{((U')^T (x_n \odot w_n) + x_{mn} u_m)_a}{((U')^T ((U' v_n) \odot w_n) + (u_m^T v_n) u_m)_a} .(26)$$

B. Summary of Computational Complexity

The computational complexity of both WNMF and IWNMF are summarized in table II. We can tell from the table that the WNMF model is computational expensively, the training time increases linearly by the number of users and items involved in the system. And the IWNMF model has a much lower computational complexity. Furthermore, the computational complexity is independent of the number of users for a new user model; symmetrically, the computational complexity is independent of the number of items for a new item model.

IV. EXPERIMENTAL RESULTS

In this section, we conduct several experiments to compare the prediction accuracy and complexities between WNMF and IWNMF model. All experiments were done on a PC with 2.8GHz processor and 2GB RAM.

A. Dataset Description

Our algorithms were evaluated on the famous benchmark collaborative filtering data set MovieLense[1], collected by GroupLens Research Lab at the University of Minnesota. The dataset contains 100,000 movie ratings of 943 users for 1682 movies, on which ratings are made on a 5-star scale, and whole-star ratings only, and 0 indicates that the movie is not rated by the user. Table III summarizes the characteristics of the dataset.

[1] http://www.grouplens.org/node/73

B. Evaluation

We use Root Mean Square Error (RMSE) to measure the predictive accuracy, which is widely used in prediction tasks. RMSE is defined as:

$$RMSE = \sqrt{\frac{\sum_{ij}(r_{ij} - \hat{r}_{ij})^2}{N}}\,, \qquad (27)$$

where r_{ij} denotes the actual ratings given by user i on item j, $\hat{r}_{ij}$ denotes the predicted ratings given by user i on item j, N denotes the number of ratings predicted on testing set.

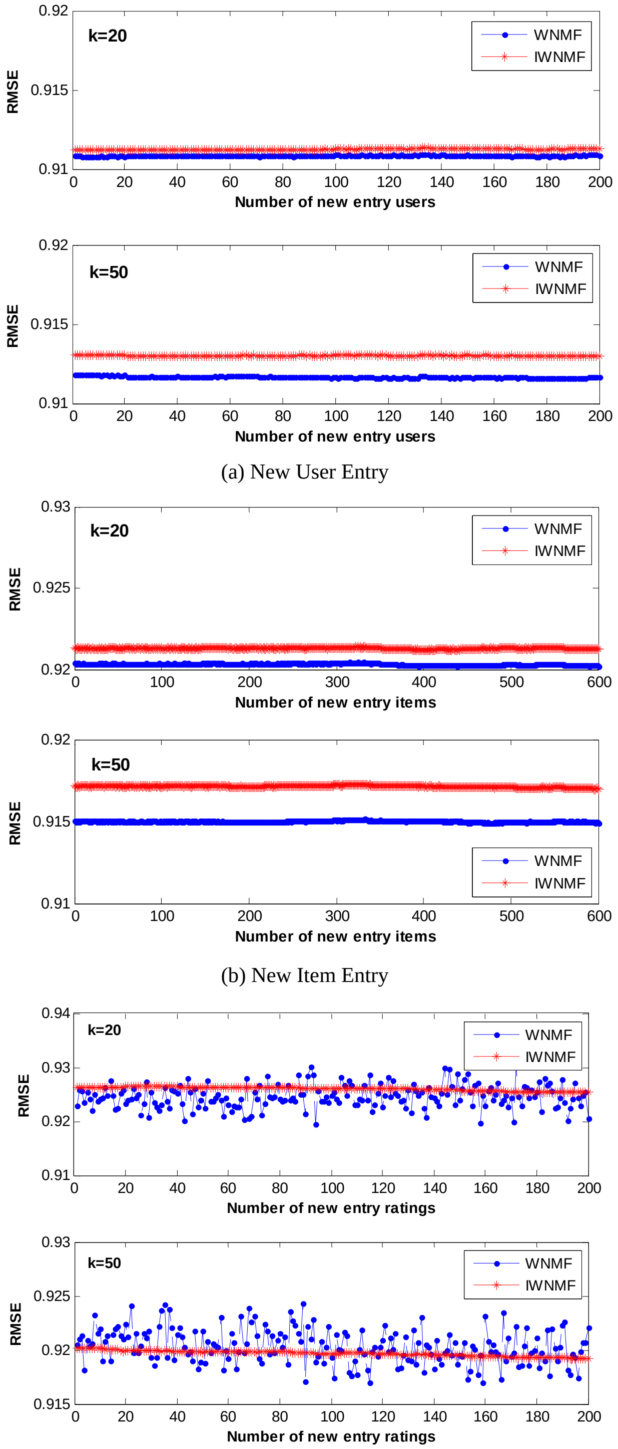
To measure the computational complexity, running time is adopted.

C. Prediction Accuracy: WNMF vs. IWNMF

Fig.3 shows the RMSE obtained by applying WNMF and IWNMF for different dynamic scenarios. The IWNMF models achieve approximately the same prediction accuracy with WNMF model.

D. Computational Complexity: WNMF vs. IWNMF

To investigate the computational complexity, we record the runtime for iteration in seconds for different models. From Fig.4, we observe that 1) In a scenario of a new user entering, the IWNMF model learns faster than the conventional WNMF, and the running time remain almost the same along with the number of new users increase; 2) In a scenario of a new item added, the IWNMF model learns faster than the conventional WNMF, and the running time remain almost the same along with the number of new items increase; 3) In a scenario of a new rating given, the IWNMF model updates faster; 4) The computational cost grows nearly linear for WNMF as the size of the system increase. The experimental results show the consistence with the theoretical analysis.



(a) New User Entry

(b) New Item Entry

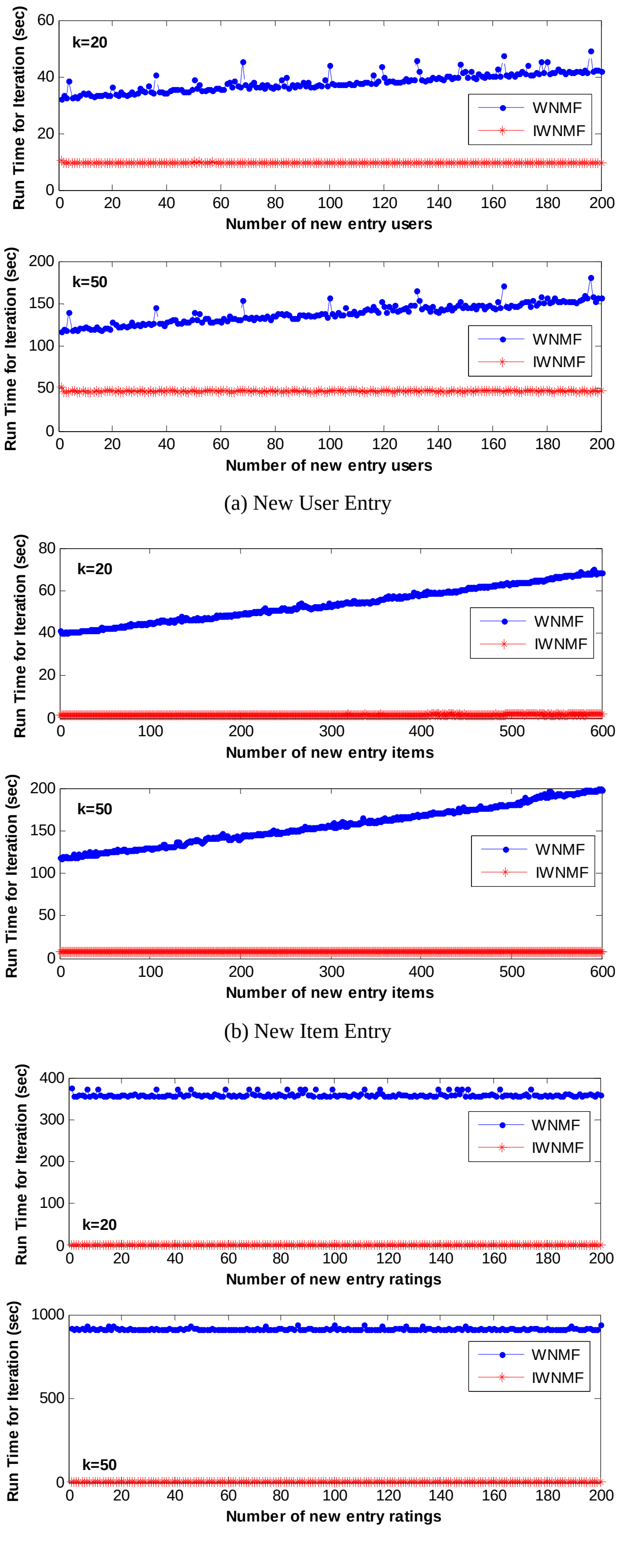
(c) New Rating Entry

Figure 3. Prediction Accuracy: WNMF vs. IWNMF

V. Related Work

Collaborative Filtering (CF) algorithms are the main and widely used technology in recommendation systems. The key idea is to predict further the interests of a user (ratings) based on the available rating information from many users. The assumption behind collaborative filtering is that users should be interested in those items that are similar with items users have selected before or have been selected by users who share similar tastes with the target user. In the literature [14] [15] [16], CF algorithms can be divided into two categories: memory-based methods and model-based methods. Memory-based methods apply some heuristic rules over all historical ratings to form a specific formulation, and directly generate predictions. The main part of memory-based methods is similarity measurements and k-nearest neighbors calculation [14] [17] [18]. Model-based methods use the historical ratings to learn a model, and then apply it for prediction. Examples of model-based methods include clustering model [19], aspect models [20], the flexible mixture model [21] and latent factor models [5] [7] [22] [16].

Due to its ability to efficiently handle large scale problems and extract latent features, Matrix Factorization has become one of the powerful CF models. It belongs to the model-based category. By using low rank approximations, it tries to capture a number of factors that really influence tastes and how each factor applies to each user. The main objective is to learn a model that



(a) New User Entry

(b) New Item Entry

(c) New Rating Entry

Figure 4. Computational Complexity: WNMF vs. IWNMF

maximizes the log-likelihood of the observed rating. In the literature, a positive regularization item or a nonnegative constraint is usually adopted to enhance the linear factor model. Takács et al. [5] proposed a model that minimizes the total squared training error on the known element, and a gradient decent algorithm is applied to solve the model. To prevent over-fitting, the model also introduce a regularization item with factor λ. The model, named as Regularized Matrix Factorization (RMF), is also mentioned in Wu's work [4]. The non-negativity constraint is enforced to guarantee the interpretability of the matrixes factorization model, which restrict that all the elements in the factor matrix U and V should be non-negative. The WNMF method proposed by Zhang et al. [7] uses the least squares error as the cost function to guide the NMF performance while Wu [4] use the generalized Kullback-Leibler divergence as the measurement. Multiplicative rules can be deduced based on the work represented by Lee and Seung [23]. Yoo and Choi [24] proposed a co-tri-factorization model which simultaneously factor several matrixes into 3-factor decompositions to effectively handle the cold-start cases. Chen et al. [25] presented an ensemble model to hybrid nonnegative matrix tri-factorization and memory-based method to improve prediction accuracy and solve the sparsity and scalability problems.

All models mentioned above are static. Once been trained, the prediction model remains fixed. Though provide good predictive accuracy, they are not capable of handling the situation where new users sign in, new items added and new ratings given over time in a real-word recommender systems. An on-line processing strategy that could make the efficient update possible is crucial for the dynamic recommend applications.

Incremental learning models based on Singular Value Decomposition (SVD) for recommender systems have been studied by Brand [26] and Sarwar et al. [27]. They do help recommender systems achieve high scalability but may suffer from imputation and over-fitting. Furthermore, a SVD requires complete data matrix, it is not the best choice for recommender systems with a highly sparse data matrix. In contrast with the SVD-based models, our work intends to present incremental learning for WNMF.

VI. Conclusion

In this paper, we propose an incremental learning framework for dynamic collaborative filtering, which is established based on the Weighted NMF model. It well addresses the time-consuming online updating problem by utilizing partially the optimization information of the original system, which leads to a very fast performance. Experiments on benchmark dataset show that the IWNMF scheme for different dynamic scenarios greatly lower the computational cost without degrading the prediction accuracy.

References

[1] Òscar Celma and Paul Lamere, "If you like the beatles you might like...: a tutorial on music recommendation," in ACM Multimedia, October 2008, pp. 1157–1158.

[2] J. Bennett and S. Lanning, "The netflix prize," in Proceedings of KDD Cup and Workshop. Citeseer, 2007.

[3] H. Ma, I. King, and M. R. Lyu, "Effective missing data prediction for collaborative filtering," in SIGIR, W. Kraaij,

A. P. de Vries, C. L. A. Clarke, N. Fuhr, and N. Kando, Eds. ACM, 2007, pp. 39–46.

[4] M. Wu, “Collaborative filtering via ensembles of matrix factorization,” in KDD Cup and Workshop, August 2007, pp. 43–47.

[5] Gábor Takács and István Pilászy and Bottyán Németh and Domonkos Tikk, “Major components of the gravity recommendation system,” SIGKDD Explorations, vol. 9, no. 2, pp. 80–83, 2007.

[6] R. Salakhutdinov and A. Mnih, “Probabilistic matrix factorization,” in NIPS, J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, Eds. MIT Press, 2007.

[7] S. Zhang, W. Wang, J. Ford, and F. Makedon, “Learning from incomplete ratings using non-negative matrix factorization,” in SDM, J. Ghosh, D. Lambert, D. B. Skillicorn, and J. Srivastava, Eds. SIAM, 2006.

[8] N. Srebro and T. Jaakkola, “Weighted low-rank approximations,” in ICML, T. Fawcett and N. Mishra, Eds. AAAI Press, 2003, pp. 720–727.

[9] S. S. Bucak, B. Günsel and O. Gursoy, “Incremental nonnegative matrix factorization for dynamic background modelling,” in PRIS, A. L. N. Fred and A. K. Jain, Eds. INSTICC PRESS, 2007, pp. 107–116.

[10] S. S. Bucak and B. Günsel, “Incremental subspace learning via non-negative matrix factorization,” Pattern Recognition, vol. 42, no. 5, pp. 788–797, 2009.

[11] C. Wen-Sheng, P. Binbin, F. Bin, L. Ming, and T. Jianliang, “Incremental nonnegative matrix factorization for face recognition,” Mathematical Problems in Engineering, vol.2008, 2008.

[12] S. Han, Y. Yang, and W. Liu, “PPNMF: Improving Weighted Nonnegative Matrix Factorization with Prior Information,” in ATAI, 2010.

[13] Q. Gu, J. Zhou, and C. Ding, “Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs,” in SDM, 2010, pp. 199–210.

[14] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” in WWW, 2001, pp. 285–295.

[15] B. Marlin, “Modeling user rating profiles for collaborative filtering,” in NIPS, S. Thrun, L. K. Saul, and B. Schölkopf, Eds. MIT Press, 2003.

[16] A. Paterek, “Improving regularized singular value decomposition for collaborative filtering,” in KDD Cup and workshop, August 2007, pp. 39–42.

[17] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, “Grouplens: An open architecture for collaborative filtering of netnews,” in CSCW, 1994, pp. 175–186.

[18] M. Deshpande and G. Karypis, “Item-based top-n recommendation algorithms,” ACM Trans. Inf. Syst., vol. 22, no. 1, pp. 143–177, 2004.

[19] L. H. Ungar and D. P. Foster, “Clustering methods for collaborative filtering,” in Workshop on Recommendation Systems at the 15th National Conference on Artificial Intelligence, August 1998, pp. 114–129.

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

[21] L. Si and R. Jin, “Flexible mixture model for collaborative filtering,” in ICML, T. Fawcett and N. Mishra, Eds. AAAI Press, 2003, pp. 704–711.

[22] N. Srebro, J. D. M. Rennie, and T. Jaakkola, “Maximum-margin matrix factorization,” in NIPS, 2004.

[23] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” in NIPS, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds. MIT Press, 2000, pp. 556–562.

[24] J. Yoo and S. Choi, “Weighted nonnegative matrix cotri-factorization for collaborative prediction,” in ACML, ser. Lecture Notes in Computer Science, Z.-H. Zhou and T. Washio, Eds., vol. 5828. Springer, 2009, pp. 396–411.

[25] G. Chen, F. Wang, and C. Zhang, “Collaborative filtering using orthogonal nonnegative matrix tri-factorization,” Inf. Process. Manage., vol. 45, no. 3, pp. 368–379, 2009.

[26] M. Brand, “Fast online svd revisions for lightweight recommender systems,” in SDM, 2003.

[27] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Incremental singular value decomposition algorithms for highly scalable recommender systems,” in Proceedings of the 5th International Conference in Computers and Information Technology. Citeseer, 2002.

Shuli Han is currently a MS candidate in Control Science and Engineering at Tsinghua University, P.R.China. She received her BS degrees in the depart-ment of Automation from Tsinghua University, China, in 2008.

Her research interests include recommender system and data mining.

Yujiu Yang received his PhD degree in Pattern Recognition from the Institute of Automation, Chinese Academy of Sciences.

His current research interests include statistical learning theory, Web Data mining and machine learning. He is also interested in applications in information retrieval and recommender system.

He is currently a Lecturer of Graduate School at Shenzhen, Tsinghua University. He is a member of the IEEE Computer Society and CCF.

Wenhuang Liu received his B.S. degree in Automation from Tsinghua University in 1970, and he was the visiting scholar Colorado University, USA, from 1986 to 1988.

He is currently a professor of the department of Automation at Tsinghua University. He was a deputy director of the National CIMS Engineering Research Center. His current research interests include data mining, and health informatics.

Prof. Liu is a recipient of the National Prize for Progress in Science and Technology.