Accelerated Online Learning for Collaborative Filtering and Recommender Systems

LI Yuan-Xiang, LI Zhi-Jie, WANG Feng, KUANG Li
State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, China

***Abstract*—Collaborative filtering (CF) is one of the major approaches to building recommender systems. Traditional batch-trained algorithms for CF suffer from some drawbacks, and online learning algorithms for CF, is a promising tool for attacking the large-scale dynamic problems. However, the low time complexity of online algorithm often be accompanied by low convergence rate, and the convergence rate of current dual-averaging online algorithm is only $O(1/\sqrt{T})$ up to *T*-th iteration. In order to tackle this problem, we propose a novel accelerated online learning framework for CF. Our algorithm has a accelerated capability, and its theoretical convergence rate bound is $O(1/T^2)$. Moreover, the proposed algorithm has low time and memory complexity, and scales linearly with the number of observed ratings. The experimental results on real-world datasets demonstrate the merits of the proposed online learning algorithm for large-scale dynamic collaborative filtering problems.**

Keywords—collaborative filtering; recommender systems; dual-averaging; online probabilistic matrix factorization; accelerated convergence

I. INTRODUCTION

With the emergence of large-scale online user-contributed websites and online shopping websites, such as MovieLens, Yahoo!Music, Amazon, etc., the user and item base expand tremendously. Recommender systems become crucial for service providers. Collaborative filtering (CF), aiming at predicting users' unknown preferences based on observational preferences from some users, are one of the major approaches to recommender systems. CF methods for recommender systems are generally categorized into memory-based and model-based methods [1], [2], [3]. In this paper, we focus on model-based techniques, specifically probabilistic matrix factorization (PMF).

Traditional PMF adopts batch-trained algorithms. These methods suffer from two major drawbacks. First, batch-trained methods for PMF scale poorly. Before training, they require that all data are available. During training, to perform the algorithms, all ratings must be scanned through once at each iteration. This is inefficient when the number of training data is so large that they cannot be loaded into the memory simultaneously. The second drawback of batch-trained algorithms is that they are unsuitable for dynamic ratings. A recommender system may change in one of the following ways: 1) new users joining the system; 2) new items arriving in the system; 3) new rating updating the system. In these cases, to capture the change, the batch-trained PMF methods have to rebuild the model, which is very expensive.

Online learning algorithm, as an alternative to parallel algorithm, become one promising technology to attack the large-scale dynamic problems. Stochastic gradient descent for PMF(SGD-PMF) naturally gives an online algorithm [4], where at each iteration, we make a small change for user u's feature vector U_u and item i's feature vector V_i when a rating (u, i, r) is revealed. These feature vectors move toward the average gradient descent, by a small step controlled by η. SGD-PMF is efficient in terms of time complexity and memory cost. Its convergence rate is $O(1/T)$ [5]. However, SGD-PMF can only search a local optimal value of the objective.

Recently, regularized dual averaging method(RDA) [6] ignited the development of online optimization algorithms. RDA is based on a convex optimization problem for which there is an iterative algorithm which converges to an optimal solution, so RDA is a overall optimization algorithm. At each iteration of this method, the learning variables are adjusted by solving a simple minimization problem that involves the running average of all past subgradients of the loss function and the whole regularization term. Due to overall optimization and efficiency of dual-averaging method, we decide to adopt it to solve the online PMF problem. However, the low time complexity of current dual-averaging online algorithm be accompanied by low convergence rate, and its convergence rate is only $O(1/\sqrt{T})$. In the literature, although there are several tasks [7], [8], [9], [10] investigating online learning for CF, they did not explore convergence problem. The low convergence rate problem has become a bottleneck for applying online PMF efficiently.

In this paper, we study online algorithms for PMF from various aspects to solve the issues facing batch-trained and previous online learning PMF algorithms. We adopt a improved mini-batch accelerated approach in dual-averaging online learning PMF framework [10], and the accelerated online algorithm has optimal convergence rate $O(1/T^2)$. We well study the convergence of the online learning algorithms from theoretical perspective, which guarantees the convergence rate of the proposed algorithm. Due to its low time complexity and optimal convergence rate, this method can improve real-time performance and scalability of realistic recommender system. To our best knowledge, this is the first attempt to develop accelerated online algorithms for PMF methods.

978-1-4799-4274-9/14 $31.00 © 2014 IEEE
DOI 10.1109/ICDMW.2014.95

IEEE computer society

II. DUAL-AVERAGING ONLINE PMF

A. Probabilistic Matrix Factorization

Definition 1. Matrix factorization collaborative filtering problem (PMF). Suppose that we are given a set of N users $\{u_1,u_2,\ldots,u_N\}$ and a set of M items $\{i_1,i_2,\ldots,i_M\}$. Users' rating on the items forms an $N\times M$ matrix R, where the element r_{ui} denotes user u's rating on item i, $r_{ui}\in[0,1]$. The problem of matrix factorization collaborative filtering is to learn two low-rank feature matrices, U and V, where U^TV fits R based on the given M, N, R. The user feature matrix U is a $K\times N$ matrix where the column U_u is user u's feature vector. V is a $K\times M$ matrix where the column V_i is item i's feature vector. The latent feature size K is much smaller than N and M.

Currently, there are some methods for solving matrix factorization collaborative filtering problem, such as batch training method for PMF (BT-PMF), stochastic gradient descent method for PMF (SGD-PMF), and dual-averaging method for PMF (DA-PMF) etc.

For BT-PMF method, its objective is equivalent to minimizing a squared loss with regularization,

$$\min_{U,V}\varphi(U,V)=\frac{1}{2}\sum_{u=1}^{N}\sum_{i=1}^{M}I_{ui}\left(r_{ui}-y_{ui}\right)^2+\Omega_{\lambda_U}(U)+\Omega_{\lambda_V}(V) \quad (1)$$

where loss function $l_{ui}=\frac{1}{2}\left(r_{ui}-y_{ui}\right)^2$, y is the logistic function: $y(x)=1/(1+\exp(-x))$. We use y_{ui} to denote $y\left(U_u^TV_i\right)$. I_{ui} is an indicator function which equals 1 if user u have rated item i and 0 otherwise. $\Omega_{\lambda_U}(U)$ and $\Omega_{\lambda_V}(V)$ are regularization terms.

For SGD-PMF method, suppose the new coming rating is $(u, i, r)\in Z$, in Eq. (1), the terms related to this particular rating are,

$$\varphi_{(u,i,r)}=\frac{1}{2}\left(r_{ui}-y_{ui}\right)^2+\frac{\lambda_U}{2}\left\|U_u\right\|_2^2+\frac{\lambda_V}{2}\left\|V_i\right\|_2^2. \quad (2)$$

By adopting the gradient descent method, we obtain the following update equations,

$$U_u\leftarrow U_u-\eta((y_{ui}-r_{ui})y'_{ui}V_i+\lambda_UU_u), \quad (3)$$

$$V_i\leftarrow V_i-\eta((y_{ui}-r_{ui})y'_{ui}U_u+\lambda_VV_i), \quad (4)$$

where η is the step size controlling how much change to make at each step. SGD-PMF is stochastic in the sense that every time we adjust the parameter. This method can be adopted to reach a local minimum of the objective given in Eq.(2), its convergence rate is $O(1/T)$. Since only user feature matrix and item feature matrix have to be stored in memory, the memory cost is $O((N+M)K)$. For each observation $(u, i, r)\in Z$, only $O(K)$ steps are needed to update the model. So SGD-PMF is efficient in terms of time complexity and memory cost.

B. Dual-Averaging Online PMF

In [6], a regularized dual averaging method(RDA) is proposed, where the learning variable is updated based on the average of all past calculated subgradients of the loss functions. At each iteration of dual-averaging method, there are three steps to complete [14]. 1) compute the subgradient of loss function; 2) update the average of all past calculated subgradients; 3) calculate the learning variables of next iteration based on the average subgradient.

For DA-PMF, its objective variables are user feature matrix U and item feature matrix V, respectively [10]. $G_U^t\in\partial l_U^t(),G_V^t\in\partial l_V^t()$ respectively denote the subgradients of loss function over U and V at t-th iteration. When a newly observed rating (u, i, r) is revealed,

$$G_{U_u}^t=\left(y_{ui}-r_{ui}\right)y'_{ui}V_i^{t-1},. \quad (5)$$

$$G_{V_i}^t=\left(y_{ui}-r_{ui}\right)y'_{ui}U_u^{t-1}. \quad (6)$$

We obtain G_U^t,G_V^t by updating G_U^{t-1},G_V^{t-1} using $G_{U_u}^t,G_{V_i}^t$. Then we keep track of $\bar{G}_U^t,\bar{G}_V^t$, the average subgradient of the squared loss with respect to U and V at t-th iteration as follows:

$$\bar{G}_U^t\leftarrow\frac{t-1}{t}\bar{G}_U^{t-1}+\frac{1}{t}G_U^t, \quad (7)$$

$$\bar{G}_V^t\leftarrow\frac{t-1}{t}\bar{G}_V^{t-1}+\frac{1}{t}G_V^t. \quad (8)$$

Once we get the average subgradient, we update U and V by:

$$U_t=\arg\min_U\phi_U(U)=\left\{\left\langle\bar{G}_U^t,U\right\rangle+\Omega_{\lambda_U}(U)+h_U(U)\right\}.$$

$$V_t=\arg\min_V\phi_V(V)=\left\{\left\langle\bar{G}_V^t,V\right\rangle+\Omega_{\lambda_V}(V)+h_V(V)\right\}.$$

Where $h_U(U)$, $h_V(V)$ are auxiliary strongly convex functions.

The typical dual-averaging online PMF is summarized in Algorithm 1. $\|\bullet\|_F$ denote Frobenius norm.

In Algorithm 1, the key to solve the DA-PMF efficiently depends on the simplicity of updating of the 3) step. Given $\Omega_{\lambda_U}=\frac{1}{2}\lambda_U\|U\|_F^2,\Omega_{\lambda_V}=\frac{1}{2}\lambda_V\|V\|_F^2$, the solution to Eq. (10) and Eq. (11) can be found analytically by taking the derivative to 0 as follows,

$$U_t\leftarrow\frac{-\bar{G}_U^t}{\lambda_U+1},V_t\leftarrow\frac{-\bar{G}_V^t}{\lambda_V+1}. \quad (9)$$

Obviously, the computation time of Eq. (9) is a constant. The memory cost for DA-PMF includes the user and item feature matrix U, V, approximate average gradient $\bar{G}_U$, $\bar{G}_V$. The total memory cost is still $O((N+M)K)$. The convergence rate of DA-PMF is only $O\left(1/\sqrt{T}\right)$, T is number of iteration.

Algorithm 1: Dual-Averaging Online PMF(DA-PMF)

Input: $h_U(U)=\frac{1}{2}\|U\|_F^2, h_V(V)=\frac{1}{2}\|V\|_F^2$; $\alpha_1=1$.

Initialization: $U_0=V_0=\min_W \frac{1}{2}\|W\|_F^2$, $\overline{G}_U^0=\overline{G}_V^0=0$.

for t =1, 2, 3, ⋯, **do**

1) given $(u, i, r) \in Z$, compute $G_{U_u}^t, G_{V_i}^t$ by (5), (6).

2) update $\overline{G}_U^t, \overline{G}_V^t$ by (7), (8).

3) output U_t, V_t :

$$U_t=\arg\min_U \phi_U(U)=\left\{\left\langle \overline{G}_U^t, U\right\rangle+\Omega_{\lambda_U}(U)+\frac{1}{2}\|U\|_F^2\right\}. \quad (10)$$

$$V_t=\arg\min_V \phi_V(V)=\left\{\left\langle \overline{G}_V^t, V\right\rangle+\Omega_{\lambda_V}(V)+\frac{1}{2}\|V\|_F^2\right\}. \quad (11)$$

end for

III. Accelerated Online PMF

A. Accelerated Online PMF Algorithm

We propose a novel accelerated online PMF where regularized dual averaging method is fused with accelerated techniques. Accelerated online learning framework for PMF is outlined in Algorithm 2.

Comparing Algorithm 2 with Algorithm 1, we observe that two query points P_t and Q_t are added in regularized dual-averaging PMF framework. At each iteration, $\overline{G}_P^t, \overline{G}_Q^t$ are the subgradients of loss function over query points P_t and Q_t, respectively. Query points P_t and Q_t are adjusted by Eq. (15). We adopt input sequence (14) to move P_t and Q_t toward U_t and V_t as soon as possible. It must be pointed out that we change two auxiliary strongly convex functions of Eq. (12) and (13) to $h_U(U)=\frac{1}{2}\|U-P_t\|_F^2$, $h_V(V)=\frac{1}{2}\|V-Q_t\|_F^2$. Since, we not only need to add the convexity of optimization function, and the more important target is that through fusing $\frac{1}{2}\|U-P_t\|_F^2$, $\frac{1}{2}\|V-Q_t\|_F^2$ into the optimization objective Eqs. (12), (13) of Algorithm 2, the optimization variables U_t and V_t can move fasterly toward the query points P_t and Q_t, which is necessory condition to obtain continuous acceleration. So our accelerated technique in Algorithm 2 is improved mini-batch approach, which obtain faster accelerated convergence rate.

Similar to Algorithm 1, the key to solve Algorithm 2 efficiently depends on the simplicity of updating U_t, V_t in 3) step. Here, we introduce: $\Omega_{\lambda_U}=\frac{1}{2}\lambda_U\|U\|_F^2, \Omega_{\lambda_V}=\frac{1}{2}\lambda_V\|V\|_F^2$. Then the solution to Eq. (12) and Eq. (13) can be found analytically by taking the derivative to 0 as follows,

$$U_t \leftarrow \frac{P_t-\overline{G}_P^t}{\lambda_U+1}, V_t \leftarrow \frac{Q_t-\overline{G}_Q^t}{\lambda_V+1}.$$

For Algorithm 2, the user and item feature matrix U, V, approximate average gradient $\overline{G}_U$, $\overline{G}_V$ have to be stored in memory, and total memory cost is $O((N+M)K)$. K is generally on the scale of tens even for a very big dataset. For each observation (u, i, r), only $O(K)$ steps are needed to update the model. So Algorithm 2 scales linearly with the number of observed ratings, and it can be effective to large-scale datasets. ADA-PMF not only has low time complexity and memory cost, but also has optimal convergence rate $O(1/T^2)$.

Algorithm 2: Accelerated online PMF (ADA-PMF)

Input: $h_U(U)=\frac{1}{2}\|U-P_1\|_F^2, h_V(V)=\frac{1}{2}\|V-Q_1\|_F^2$; $\alpha_1=1$.

Initialization: $U_0=V_0=P_1=Q_1=\min_W \frac{1}{2}\|W\|_F^2$, $\overline{G}_U^0=\overline{G}_V^0=0$.

for t =1, 2, 3, ⋯, **do**

1) given $(u, i, r) \in Z$, compute $G_{P_u}^t, G_{Q_i}^t$ by (5), (6).

2) update $\overline{G}_P^t, \overline{G}_Q^t$ by (7), (8).

3) output U_t, V_t :

$$U_t=\arg\min_U \phi_U(U)=\left\{\left\langle \overline{G}_P^t, U\right\rangle+\Omega_{\lambda_U}(U)+\frac{1}{2}\|U-P_t\|_F^2\right\}. \quad (12)$$

$$V_t=\arg\min_V \phi_V(V)=\left\{\left\langle \overline{G}_Q^t, V\right\rangle+\Omega_{\lambda_V}(V)+\frac{1}{2}\|V-Q_t\|_F^2\right\}. \quad (13)$$

4) update α_t sequence:

$$\alpha_{t+1}=\frac{1+\sqrt{1+4\alpha_t^2}}{2}. \quad (14)$$

5) compute the query point:

$$P_{t+1}=U_t+\left(\frac{\alpha_t-1}{\alpha_{t+1}}\right)(U_t-U_{t-1}), Q_{t+1}=V_t+\left(\frac{\alpha_t-1}{\alpha_{t+1}}\right)(V_t-V_{t-1}). \quad (15)$$

end for

B. Convergence Analysis

A main issue to guarantee the online learning Algorithm 2 is to provide theoretical analysis for convergence rate. The objective of regularized stochastic learning problem for online PMF are of the following form,

$$\min_W \varphi(W)=E_Z l(W,Z)+\Omega_\lambda(W), \quad (16)$$

where Z is stochasticly observed triplet set, W is the optimization matrix which is equivalent to U or V, and $\varphi(W)$ is the objective value of online learning for PMF.

Definition 2. Convergence rate of online learning for PMF. The objective value $\varphi(W)$ of online learning for PMF is defined as Eq. (16). Here, W is the optimization matrix which is equivalent to U or V. Suppose there exists an optimal solution W^* for the problem of (16), then up to the T- step, the convergence rate of online learning for PMF is s_T as follows,

$$s_T = \varphi(W_T) - \varphi(W^*). \tag{17}$$

THEOREM 1. Suppose there exists an optimal solution W^* for Eq. (17) in Algorithm 2, W is the optimization matrix which is equivalent to U or V. Then we have the following properties in Algorithm 2: for each $T \geq 1$, the convergence rate up to T-th step is bounded by,

$$s_T = \varphi(W_T) - \varphi(W^*) \leq \frac{2\|W_0 - W^*\|_F^2}{(T+1)^2} = O\left(\frac{1}{T^2}\right). \tag{18}$$

Proof. In Algorithm 2, there are two symmetrical objective variables U and V. To avoid repetition, we use unitedly W which is equivalent to U or V, $\phi(W, O_t)$ equivalent to $\phi_U(U, P_t)$ or $\phi_V(V, Q_t)$, $\Omega_\lambda(W)$ equivalent to $\Omega_{\lambda_U}(U)$ or $\Omega_{\lambda_V}(V)$. Then, it follows from Eq. (12) or Eq. (13) in Algorithm 2,

$$W_t = \arg\min_W \phi(W, O_t) = \left\{ \left\langle \bar{G}_O^t, W \right\rangle + \Omega_\lambda(W) + \frac{1}{2}\|W - O_t\|_F^2 \right\}.$$

Obviously, it follows from the above equation,

$W_t = \arg\min_W \phi(W, O_t) = \mu(O_t)$, where W_t is equivalent to U_t or V_t, and O_t is equivalent to P_t or Q_t.

Since both the loss function l and the regularization term Ω_λ in Algorithm 2 are convex, then for any matrix variable, $\forall X, Y \in dom\Omega_\lambda$, we have,

$$\begin{aligned} &l(X) \geq l(Y) + < X - Y, \nabla l(Y) >, \\ &\Omega_\lambda(X) \geq \Omega_\lambda(\mu(Y)) + < X - \mu(Y), g(\mu(Y)) >. \end{aligned}$$

Where $g(\mu(Y)) \in \partial\Omega_\lambda(\mu(Y))$. Summing up the above two inequalities we obtain that

$$\begin{aligned} &\varphi(X) \geq l(Y) + < X - Y, \nabla l(Y) > \\ &+\Omega_\lambda(\mu(Y)) + < X - \mu(Y), g(\mu(Y)) > \end{aligned} \tag{19}$$

Since $\phi(\mu(Y), Y) = \varphi(\mu(Y)) + \frac{1}{2}\| \mu(Y) - Y \|_F^2$,

we have that

$$\begin{aligned} &\phi(\mu(Y), Y) = l(Y) + < \mu(Y) - Y, \nabla l(Y) > \\ &+\Omega_\lambda(\mu(Y)) + \frac{1}{2}\| \mu(Y) - Y \|_F^2. \\ &\partial\phi = \nabla l(Y) + (\mu(Y) - Y) + g(\mu(Y)) = 0. \end{aligned} \tag{20}$$

By combining the Eq. (19) and Eq. (20), we obtain that

$$\begin{aligned} &\varphi(X) - \varphi(\mu(Y)) \geq \varphi(X) - \phi(\mu(Y), Y) \\ &\geq < X - \mu(Y), \nabla l(Y) + g(\mu(Y)) > - \frac{1}{2}\| \mu(Y) - Y \|_F^2 \\ &= < Y - X, \mu(Y) - Y > + \frac{1}{2}\| \mu(Y) - Y \|_F^2. \end{aligned}$$

i.e.
$$\begin{aligned} &\varphi(X) - \varphi(\mu(Y)) \geq \\ &< Y - X, \mu(Y) - Y > + \frac{1}{2}\| \mu(Y) - Y \|_F^2 \end{aligned} \tag{21}$$

Let us denote

$$\begin{aligned} s_t &= \varphi(W_t) - \varphi(W^*), \\ J_t &= \alpha_t W_t - (\alpha_t - 1) W_{t-1} - W^*. \end{aligned} \tag{22}$$

Applying Eq.(21) with $X = W_t$, $Y = O_{t+1}$ and $X = W^*$, $Y = O_{t+1}$, respectively, we obtain the following two inequalities:

$$2(s_t - s_{t+1}) \geq \| W_{t+1} - O_{t+1} \|_F^2 + 2 < W_{t+1} - O_{t+1}, O_{t+1} - W_t > \tag{23}$$

$$-2s_{t+1} \geq \| W_{t+1} - O_{t+1} \|_F^2 + 2 < W_{t+1} - O_{t+1}, O_{t+1} - W^* > \tag{24}$$

Multiplying both sides of Eq. (23) by (α_{t+1}-1) and adding it to Eq. (24), we get

$$\begin{aligned} &2\left((\alpha_{t+1} - 1)s_t - \alpha_{t+1}s_{t+1}\right) \geq \alpha_{t+1}\|W_{t+1} - O_{t+1}\|_F^2 \\ &+2\left\langle W_{t+1} - O_{t+1}, \alpha_{t+1}O_{t+1} - (\alpha_{t+1} - 1)W_t - W^* \right\rangle \end{aligned}$$

Multiplying the above inequality by α_{t+1} and making use of the equality $\alpha_t^2 = \alpha_{t+1}^2 - \alpha_{t+1}$ derived from Eq. (14), we get

$$\begin{aligned} &2(\alpha_t^2 s_t - \alpha_{t+1}^2 s_{t+1}) \geq \| \alpha_{t+1}(W_{t+1} - O_{t+1}) \|_F^2 \\ &+2\alpha_{t+1} < W_{t+1} - O_{t+1}, \alpha_{t+1}O_{t+1} - (\alpha_{t+1} - 1)W_t - W^* > \end{aligned} \tag{25}$$

Making use of the fact that for any three matrices A, B, and C of the same size,

$$\| B - A \|_F^2 + 2 < B - A, A - C > = \| B - C \|_F^2 - \| A - C \|_F^2$$

we obtain that

$$\begin{aligned} &2(\alpha_t^2 s_t - \alpha_{t+1}^2 s_{t+1}) \geq \| \alpha_{t+1}W_{t+1} - (\alpha_{t+1} - 1)W_t - W^* \|_F^2 \\ &- \| \alpha_{t+1}O_{t+1} - (\alpha_{t+1} - 1)W_t - W^* \|_F^2. \end{aligned}$$

It follows from Eq. (15) and the definition of J_t in Eq. (22) that

$$2\alpha_t^2 s_t - 2\alpha_{t+1}^2 s_{t+1} \geq \| J_{t+1} \|_F^2 - \| J_t \|_F^2. \tag{26}$$

Summing the above inequality over $t = 1, 2, \ldots, t-1$, we get

$$2s_1 - 2\alpha_t^2 s_t \geq \| J_t \|_F^2 - \| J_1 \|_F^2 . \quad (27)$$

Applying Eq. (21) with $X=W^*$, $Y=O_1$, we get

$$-2s_1 \geq 2\left\langle O_1 - W^*, W_1 - O_1 \right\rangle + \left\| W_1 - O_1 \right\|_F^2 .$$

Making use of the fact,

$$\| B - A \|_F^2 + 2 < B - A, A - C > = \| B - C \|_F^2 - \| A - C \|_F^2 ,$$

We obtain,

$$2s_1 \leq \| O_1 - W^* \|_F^2 - \| W_1 - W^* \|_F^2 . \quad (28)$$

By combining Eq. (27), Eq. (28), we get:

$$\| O_1 - W^* \|_F^2 - 2\alpha_t^2 s_t \geq \| J_t \|_F^2 \geq 0.$$

Since $\alpha_t \geq (t+1)/2$, we obtain that

$$s_t \leq \frac{2 \| W_0 - W^* \|_F^2}{(t+1)^2}, \text{ i.e., } s_T = O\left(\frac{1}{T^2}\right).$$

This completes the proof of the theorem 1.

IV. Experiments

In this section, we conduct experiments to compare the performances of four PMF algorithms: accelerated dual-averaging PMF(ADA-PMF), dual-averaging PMF(DA-PMF), stochastic gradient descent PMF(SGD-PMF) and batch-trained PMF(BT-PMF). The experiments are performed on a PC with 2.80GHz Dual CPU, 3.93GB RAM. We choose data set MovieLens (http://www.cs.umn.edu/Research/Group-Lens) and Yahoo!Music (http://kddcup.yahoo.com) to study empirical performances of our algorithms. Table 1 shows the basic statistics of each dataset. Data stream sampling adopt the ring circular sliding window algorithm [16]. We use gridregression.py function of LIBSVM tool to optimize the parameters of algorithms. All algorithms run on Matlab.

TABLE I Statistics Of Datasets

	MovieLens	Yahoo!Music
No. ratings	1,000,209	252,800,275
No. users	6,040	1,000,990
No. items	3,952	624,961
Rating range	[1,5]	[0,100]

A. Model Performance

In order to evaluate scalability performance of online algorithms, we conduct experiments with the following three settings: **1) T1:** Randomly choose 10% of all (u, i, r) triplets for training, and use remaining 90% for evaluation; **2) T5:** Randomly choose 50% of all (u, i, r) triplets for training, and use remaining 50% for evaluation; **3) T9:** Randomly choose 90% of all (u, i, r) triplets for training, and use remaining 10% for evaluation.

Data stream sampling adopt the ring circular sliding window algorithm [16]. We adopt Root Mean Square Error (RMSE) to evaluate rating-oriented algorithms. RMSE evaluates the root of average square error between true rating and predicted rating:

$$RMSE = \sqrt{\sum_{(u,i,r) \in Z} \left(\hat{r}_{u,i} - r \right)^2 / |Z|} .$$

Table 2 reports the best performance of online and batch mode PMF under different settings. We have following observations:

1) Overall, online PMF algorithms perform as well as batch-trained algorithm on different datasets.
2) Under **T1**, online algorithms even outperforms batched-trained algorithm a little bit. This may be due to the fact that under the scenario of few training samples, online learning algorithms are less likely to be trapped in a local optimum.
3) Due to the huge size of Yahoo!Music dataset, we were unable to perform batch-trained PMF using **T9** setting. Under **T5**, batch-trained PMF takes more than 6 hours to finish 120 iterations to converge. Our ADA-PMF algorithm take only about 1 minute to finish processing all 180 million ratings to reach a similar performance. The time saving is phenomenal. So our ADA-PMF adapt to various settings **T1**, **T5**, **T9** very easily. Due to superior scalability performance, ADA-PMF is suitable to large-scale stream dataset of recommender systems.

TABLE II Rmses Of PMF Algorithms On Different Settings

	MovieLens			Yahoo!Music	
	T1	**T5**	**T9**	**T1**	**T5**
BT-PMF	1.002	0.902	**0.868**	28.88	23.54
SGD-PMF	**0.992**	0.895	0.869	29.24	24.02
DA-PMF	0.998	0.899	0.882	28.41	22.86
ADA-PMF	0.997	**0.893**	0.876	**28.35**	**22.82**

B. Convergence Rate

To evaluate the convergence of the proposed formulations, we report the convergence time of the online learning algorithms[15]. For all methods, we terminate the algorithms when the relative changes in the objective is below 10^{-8}. We call it convergence time of the algorithm. The lower the convergence time, the faster the convergence procedure of this algorithm. The averaged convergence time on Yahoo!Music over ten random trials for each method is reported in Figure 1. We can observe that ADA-PMF is by far the most efficient method in all settings. This is because that according to theoretical analysis results for ADA-PMF, SGD-PMF, DA-PMF, their convergence rates are $O(1/T^2), O(1/T), O(1/\sqrt{T})$, respectively.

In order to futher investigate the convergence behaviors of three online algorithms, we plot the convergence of RMSEs of these methods on MovieLens using **T5** setting in Figure 2. We

can observe that ADA-PMF converge much faster than other two online algorithms, especially at early iterations. This is consistent with our theoretical results and confirms that the proposed accelerated scheme can reach the optimal value rapidly.

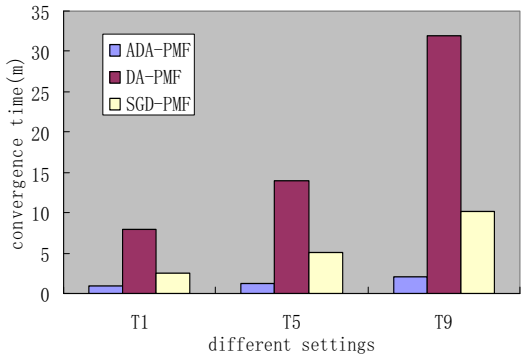

Figure 1. Comparison of ADA-PMF, SGD-PMF, DA-PMF in terms of the convergence time on Yahoo!Music using different settings.

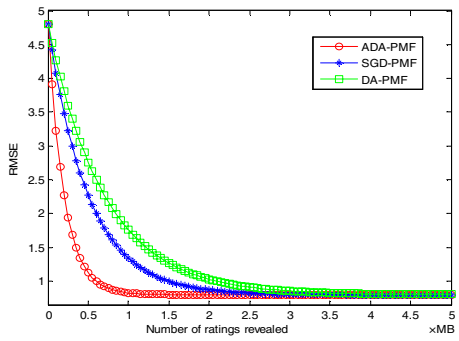

Figure 2. Comparison of three online algorithms in terms of the convergence of RMSEs on MovieLens using **T5** setting.

Due to accelerated convergence and low time complexity, ADA-PMF has superior real time peformance. Suppose that a newly coming data stream must be processed and its result be returned below T time, we call it real-time T [17][18]. We observe that while real-time T is fewer than threshold, such as 0.5s, SGD-PMF, especially DA-PMF, the real averaging rating stream processing rate P obviously decend. However, ADA-PMF almost unchange, and its real averaging rating stream processing rate P is about 100%. The superior real time performance of ADA-PMF assure it can adapt to stochastic dynamic collaborative filtering scenario.The superior real time performance of ADA-PMF assure it can adapt to stochastic dynamic collaborative filtering scenario.

V. CONCLUSION

In this paper, we have thoroughly investigated the online learning algorithms for rating-oriented collaborative filtering model, PMF. Different from previous online PMF, our proposed algorithm has the accelerated capability, and its theoretical convergence rate bound is $O\left(1/T^2\right)$. Based on dual-averaging framework, ADA-PMF is a overall optimization method. Moreover, it has low time and space complexity. Experimental results show that our online algorithm achieve comparable performance as batch-trained and other online algorithms while dramatically boosting efficiency. ADA-PMF can improve real-time performance and scalability performance of recommender system. It is suitable to large-scale dynamic collaborative filtering scenario.

There are several directions worthy of considering for future study. 1) One direction is to explore the online optimization framework with different types of regularizations to achieve solutions with different properties. 2) Multi-kernel learning is one of basic problems for machine learning. How to design accelerated online multi-kernel learning method for PMF, is next goal of our study works.

REFERENCES

[1] Deshpande, M, and Karypis, G. Item-based top-n recommendation algorithms. ACM Trans Inf Syst., 2004, 22 (1): 143-177.

[2] Liu, NN, and Yang, Q. Eigenrank: a ranking-oriented approach to collaborative filtering. In SIGIR, 2008, 83-90.

[3] Liu, NN, Zhao, M, and Yang, Q. Probabilistic latent preference analysis for collaborative filtering. In CIKM, 2009, 759-766.

[4] Koren, Y. Collaborative filtering with temporal dynamics. In KDD, 2009, 447-456.

[5] Shapiro, A, and Wardi, Y. Convergence analysis of gradient descent stochastic algorithms. Journal of Optimization Theory and Applications, 1996, 91(6): 439-454.

[6] Xiao L. Dual averaging method for regularized stochastic learning and online optimization. Journal of Machine Learning Research, 2010, 11(4): 2543-2596.

[7] Abernethy, J, Canini, K, Langford, J, and Simma, A. Online collaborative filtering. Technical Report, University of California at Berkeley, 2007.

[8] Liu, NN, Zhao, M, Xiang, EW, and Yang, Q. Online evolutionary collaborative filtering. In RecSys, 2010, 95-102.

[9] Mairal, J, Bach, F, Ponce, J, and Sapiro, G. Online learning for matrix factorization and sparse coding. Machine Learning Research, 2010, 11(3): 19-60.

[10] Ling, G, Yang, HQ, King I, Lyu, MR. Online learning for collaborative filtering. In IJCNN, 2012.

[11] Salakhutdinov, R, and Mnih, A. Probabilistic matrix factorization. In NIPS, 2007.

[12] Salakhutdinov, R, and Mnih, A. Bayesian probabilistic matrix factorization using markov chain monte carlo. In ICML, 2008, 759-766.

[13] Shi, Y, Larson, M, and Hanjalic A. List-wise learning to rank with matrix factorization for collaborative filtering. In RecSys, 2010, 269-272.

[14] Nesterov, Y. Primal-dual subgradient methods for convex problems. Mathematical Programming, 2009, 120(1): 221-259.

[15] Argyriou, A, Evgeniou, T, Pontil, M. Convex multi-task feature learning. Machine Learning, 2008, 73(7): 243-272. DOI 10.1007/s10994-007-5040-8.

[16] Oboxinski, G, Taskar, B, Jordan, MI. Joint covariate selection and joint subspace selection for multiple classification problems. Statistics and Computing, 2009.

[17] Zhan, Y, Wu, CM, Wang BJ. An algorithm for data stream sampling based on ring circular sliding window tightly coupled with buffer. Acta Electronica Sinica, 2011, 39(4): 894-898. (in chinese)

[18] Zhan, Y, Wu, CM, Wang BJ. An algorithm for data stream speed anomaly detection based on ring circular sliding window. Acta Electronica Sinica, 2012, 40(4): 674-680. (in chinese)