11-2013

Second Order Online Collaborative Filtering

Jing Lu

Steven C. H. HOI
Singapore Management University, chhoi@smu.edu.sg

Jialei Wang

Peilin Zhao

Citation

Lu, Jing; HOI, Steven C. H.; Wang, Jialei; and Zhao, Peilin. Second Order Online Collaborative Filtering. (2013). *JMLR: Workshop and Conference Proceedings: ACML 2013, November 13-15, Canberra*. 29, 325-340.

Available at: https://ink.library.smu.edu.sg/sis_research/2288

Second Order Online Collaborative Filtering

Jing Lu JLU010@E.NTU.EDU.SG
Steven Hoi CHHOI@NTU.EDU.SG
Jialei Wang JL.WANG@NTU.EDU.SG
Peilin Zhao ZHAO0106@NTU.EDU.SG
School of Computer Engineering, Nanyang Technological University, Singapore

Editor: Cheng Soon Ong and Tu Bao Ho

Abstract

Collaborative Filtering (CF) is one of the most successful learning techniques in building real-world recommender systems. Traditional CF algorithms are often based on batch machine learning methods which suffer from several critical drawbacks, e.g., extremely expensive model retraining cost whenever new samples arrive, unable to capture the latest change of user preferences over time, and high cost and slow reaction to new users or products extension. Such limitations make batch learning based CF methods unsuitable for real-world online applications where data often arrives sequentially and user preferences may change dynamically and rapidly. To address these limitations, we investigate online collaborative filtering techniques for building live recommender systems where the CF model can evolve on-the-fly over time. Unlike the regular first order CF algorithms (e.g., online gradient descent for CF) that converge slowly, in this paper, we present a new framework of second order online collaborative filtering, i.e., Confidence Weighted Online Collaborative Filtering (CWOCF), which applies the second order online optimization methodology to tackle the online collaborative filtering task. We conduct extensive experiments on several large-scale datasets, in which the encouraging results demonstrate that the proposed algorithms obtain significantly lower errors (both RMSE and MAE) than the state-of-the-art first order CF algorithms when receiving the same amount of training data in the online learning process.

Keywords: Collaborative Filtering, Online Learning, Matrix Factorization, Second Order Optimization

1. Introduction

Collaborative Filtering (CF), an approach that uses known preferences of some users to make predictions to the unknown preferences of other users, has been widely used as one of core learning techniques in building real-world recommender systems, including many commercial websites such as *Amazon*, *Barnes*, *Netflix*, and *eBay*. Consider online e-commerce applications where a user wishes to watch a movie or buy a product, the system offers recommendations using CF techniques in exploiting one's previous preference and that of others. A good recommender system is extremely beneficial to users in accurately predicting their preferences and providing satisfactory recommendations, and consequently benefiting the company. The fundamental assumption of CF is that if two users rate many items similarly, they will be likely to rate other items similarly (Goldberg et al., 2001).

In literature, a variety of CF algorithms have been proposed, which can be generally grouped in two categories: memory-based CF and model-based CF (Su and Khoshgoftaar, 2009). Memory-based CF algorithms were wildly used in some early generation CF systems. The systems measure the similarity between items (or users) based on the ratings in the training dataset and then make prediction to the unknown ratings based on the weighted average of ratings to similar items (or from similar users) (Sarwar et al., 2001). The main limitation of memory-based CF methods is that they often suffer from the data sparsity issue which makes the computation of similarity inaccurate or even impossible. To overcome the drawbacks of memory-based CF, model-based CF methods have been proposed and found encouraging results even in the challenging scenarios of sparse data (Miyahara and Pazzani, 2002; Ungar and Foster, 1998). Among different CF techniques, one of the most successful approaches is the matrix factorization methodology (Koren et al., 2009). Our work also follows the same methodology due to its leading performance in practice.

Despite being studied extensively, most traditional model based CF algorithms are based on batch machine learning techniques which assume all training data are provided prior to the model training (Sarwar et al., 2002; Linden et al., 2003). Such assumption makes them unsuitable and non-scalable for real-world large-scale online applications for a few reasons. First of all, the ratings usually arrive sequentially and periodically in an online application; as a result, batch learning model has to be retrained from scratch whenever new training samples are received, making the training process extremely expensive. Second, whenever a new item or a user is added to the system from time to time, batch learning cannot handle such changes immediately without involving an expensive re-training process. Third, it is common that users preferences could drift rapidly over time in real-world online applications, making the batch learning approaches fail to capture such rapid changes on time. This motivates us to study online/incremental collaborative filtering techniques to address these limitations.

Recent years have witnessed some emerging studies for online collaborative filtering (Abernethy et al., 2007; Ali et al., 2011). Unfortunately, these methods generally follow the first order optimization framework (e.g., online gradient descent (Abernethy et al., 2007)) in finding the optimal solutions of low-rank matrix factorization. Due to the ignorance of second order information, these approaches suffer from slow convergence. To address the weakness of these first order online CF approaches, we propose a novel framework of Confidence Weighted Online Collaborate Filtering (CWOCF), which exploits the confidence information of the low rank matrixes and online second order optimization method. The key idea of CWOCF is to not only update the user and item weight vectors at each round, but also estimate their distribution, i.e., mean and covariance matrix. Because of exploiting the additional confidence information, CWOCF converges significantly faster and thus achieves much lower values of RMSE and MAE than those of the regular first order algorithms when receiving the same amount of rating observations.

The rest of the paper is organized as follows. Section 2 reviews the background and related work. Section 3 proposes the proposed Confidence Weighted Online Collaborate Filtering algorithm. Section 4 discusses the experimental results on several real-world datesets, and Section 5 concludes this work and discusses the future work.

2. Background and Related Work

This section briefly reviews the background and major related work, including matrix factorization CF methods, online collaborative filtering, and second order online optimization.

As stated above, among two major categories of CF algorithms (He et al., 2011; Melville et al., 2002; Ungar and Foster, 1998), model-based CF methods usually outperform than memory-based ones. One of the most successful approaches for model-based CF is matrix factorization, which was used to develop the algorithm that won the Netflix prize (Koren et al., 2009). This algorithm uses a model similar to the linear classification model. It assumes that the rating of an user to a item is determined by some potential features. Thus each user or item can be represented by a feature vector and the rating is the inner product of the user vector and the item vector. Although batch matrix factorization algorithms predict the preference of users relatively actuate, it is not suitable for real world online applications on large scale datesets because of the high computational cost in both time and memory.

Some recent studies have attempted to address Online collaborative filtering (OCF). An early group of studies (Crammer and Singer, 2001; Harrington, 2003) is to apply the idea of online/stochastic gradient descent and matrix factorization to find the optimal low-rank user matrix and item matrix. Another recent approach is to explore probabilistic matrix Factorization by adopting a probabilistic linear model with Gaussian observation noise and exploring the dual averaging optimization method (Nesterov, 2009). Despite simple and efficient, the common limitation of these methods is that they all belong to the first order online optimization method and thus often suffer from relatively slow convergence rate. Besides the model-based approaches, another related work is the Online Evolutionary Collaborative Filtering (OECF), which generally belongs to memory based methods (Liu et al., 2010). The key idea of their work is to treat the timestamp of a rating as an important factor when computing the similarity. Although they provide both batch and online extensions, the key limitation of this work is its extremely high computational cost, which makes it impractical for large-scale applications.

Our work aims to overcome the slow convergence limitation of first order OCF approaches by exploring second order optimization techniques, which have been actively studied for improving learning efficacy of online optimization tasks (Cesa-Bianchi et al., 2005; Wang et al., 2012; Orabona and Crammer, 2010). In particular, instead of exploiting the first order information ("mean") of weight vector during each iteration, second order algorithms attempt to maintain the second order information ("covariance") and update them efficiently. Although second order online learning has been actively studied, the most relevant existing work to our framework is the confidence weighted learning (CW) method (Dredze et al., 2008; Crammer et al., 2009b), which assumes the weight vector is in Gaussian distribution and updates both the mean and the covariance matrix during each iteration. Because of the additional covariance information, CW usually achieves significantly better performance than the first order methods. Although a variety of CW algorithms have been actively studied for online classification tasks (Dredze et al., 2010; Crammer et al., 2009a), to the best of our knowledge, there is no existing study of exploring secord order online optimization for collaborative filtering tasks.

3. Confidence Weighted Online Collaborative Filtering

In this section, we present the proposed framework of Confidence Weighted Online Collaborate Filtering (CWOCF). The key idea is to follow the low-rank matrix factorization framework for online collaborative filtering and exploit confidence weighted online learning techniques in optimizing the low-rank matrixes. In the following, we will first briefly introduce the problem settings, and then present an overview of the main idea of the confidence weighted online optimization, and finally present the proposed CWOCF algorithms and discuss some open issues.

3.1. Problem Settings

Consider a typical collaborate filtering task with a total of n users and m (products) items. We denote by r_{ab} the rating given by user a to item b, e.g., $r_{ab} \in \{1, 2, 3, 4, 5\}$ for some movie recommendation datasets. The collection of users' ratings forms an incomplete rating matrix $R \in \mathbb{R}^{n\times m}$. Note that in an online learning setting, when new items/users are added or old items/users are deleted, the values of m and n could change over time. We would first assume that they are constant and known in advance, and will also discuss the issue of new users or items extension later. In general, the goal of an collaborative filtering task is to predict the unknown ratings in R based on the observed ones.

The proposed collaborative filtering framework follows the principle of matrix factorization techniques for recommender systems (Koren et al., 2009). Specifically, the matrix factorization model maps both users and items into a joint low-dimensional latent factor space with very small dimensionality $k \ll n, m$, i.e., each user a can be represented by a vector $U_a \in \mathbb{R}^k$, and each item b can be represented by a vector $V_b \in \mathbb{R}^k$. Consequently, the rating r_{ab} can then be approximated by the dot product of U_a and V_b, i.e., $r_{ab} = U_a^\top V_b$. To be concise, one can respectively define the user matrix $U \in \mathbb{R}^{k\times n}$ and the item matrix $V \in \mathbb{R}^{k\times m}$, whose columns are the k-dimensional vectors of certain users/items. The goal is to find the optimal matrixes U and V that minimizes the approximation error:

$$\mathop{\arg\min}_{U\in\mathbb{R}^{k\times n}, V\in\mathbb{R}^{k\times m}} ||R - U^\top V||_F^2, \tag{1}$$

where $||X||_F$ is the Frobenius norm of the matrix X, i.e. $\sum_{i,j} x_{ij}^2$. Unfortunately, the rating matrix R is only partly observed and often extremely sparse in a CF task. Thus, traditional matrix factorization methods, such as Singular Value Decomposition (SVD), are not suitable to solve the problem. Alternatively, one can formulate the following objective function which is only related to the sum of prediction loss on all the observed ratings by excluding the unknown ones:

$$\mathop{\arg\min}_{U\in\mathbb{R}^{k\times n}, V\in\mathbb{R}^{k\times m}} \sum_{(i,j)\in C} \ell(U_a, V_b, r_{ab}) \tag{2}$$

where C is the set of all observed ratings and the loss function ℓ can be defined to minimize certain evaluation matric. In this paper, we consider two popular metrics, i.e., Root Mean Square Error (RMSE) and Mean Absolute Error(MAE), defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{|\mathcal{C}|}\sum_{(a,b)\in\mathcal{C}} (r_{a,b} - \hat{r}_{a,b})^2}, \quad \text{MAE} = \frac{1}{|\mathcal{C}|}\sum_{(a,b)\in\mathcal{C}} |r_{a,b} - \hat{r}_{a,b}|$$

where $\hat{r}_{a,b}$ denotes the predicted rating, i.e., $\hat{r}_{a,b} = U_a^\top V_b$ under the matrix factorization setting. Using the definitions, it is straightforward to define the loss function for optimizing the RMSE metric by the following square error function:

$$\ell_1(U_a, V_b, r_{a,b}) = (r_{a,b} - U_a^\top V_b)^2, \tag{3}$$

Similarly, to optimize the MAE metric, one can define the absolute loss function:

$$\ell_2(U_a, V_b, r_{a,b}) = |r_{a,b} - U_a^\top V_b|. \tag{4}$$

3.2. Overview of Confidence Weighted Learning

The Confidence Weighted (CW) algorithm (Dredze et al., 2008) was proposed to solve online linear classification tasks, i.e., finding the optimal classification weight vector w that is able to make accurate prediction of class label $y \in \{0, 1\}$ for a sequence of instances $x \in \mathbb{R}^d$ based on a linear prediction model: $\hat{y} = sign(w \cdot x)$. Unlike the first order algorithms that learn only the first order information (mean) of the weight vector, CW attempts to maintain a Gaussian distribution over the weight vector by assuming $w \sim \mathcal{N}(\mu, \Sigma)$, where $\mu \in \mathbb{R}^d$ is the mean vector and $\Sigma \in \mathbb{R}^{d\times d}$ is the covariance matrix. When making a prediction, the prediction confidence $M = w \cdot x$ also follows a Gaussian distribution: $M \sim \mathcal{N}(\mu_M, \Sigma_M)$, where $\mu_M = \mu \cdot x$ and $\Sigma_M = x^\top \Sigma x$. By formulating it as a constrained optimization, the original CW algorithm gives closed-form (approximate) solutions for optimizing the model parameters μ and Σ. Despite the pioneering study, the original CW algorithm in Dredze et al. (2008) has several drawbacks, e.g., assuming data is linearly separable and difficult to be extended for other learning tasks, which inspired a variety of improved CW algorithms Crammer et al. (2009b); Wang et al. (2012). In this work, we follow the improved CW learning study of Adaptive Regularization of Weight Vectors (AROW) Crammer et al. (2009b), which updates these parameters based on the tradeoff of three desires: (i) to prevent losing the information learnt so far, the algorithm should make the smallest change to the distribution of the weight vector as measured by the KL divergence; (ii) the new distribution must be able to make an accurate prediction to the new sample x_t, i.e., suffering the least classification loss; (iii) the prediction confidence, M, should be stable, i.e., with low variance. Consequently, the update rule is found by minimizing the following unconstrained objective function at each round:

$$\mathcal{C}(\mu, \Sigma) = D_{KL}(\mathcal{N}(\mu, \Sigma)||\mathcal{N}(\mu_{t-1}, \Sigma_{t-1})) + \lambda_1 \ell(y_t, \mu \cdot x_t) + \lambda_2 x_t^\top \Sigma x_t \tag{5}$$

where $\ell(y_t, \mu \cdot x_t) = (max\{0, 1 - y_t(\mu \cdot x_t)\})^2$ is the squared-hinge loss and λ_1 and λ_2 are regularization parameters to control the tradeoff among the three desires.

3.3. Confidence Weighted OCF (CWOCF) Algorithm

We extend the CW learning framework to tackle online collaborative filtering task by proposed the Confidence Weighted Online Collaborate Filtering (CWOCF). In particular, when receiving a rating $(a, b, r_{a,b})$, we assume that both the user vector U_a and the item vector V_b follow a Gaussian distribution, respectively: $U_a \sim \mathcal{N}(\mu_{u_a}, \Sigma_{u_a})$, $V_b \sim \mathcal{N}(\mu_{v_b}, \Sigma_{v_b})$. Because of the complexity of modeling the inner product of two Gaussian distributions, we optimize U_a and V_b separately, and assume one to be fixed when optimizing the other one.

Following the principle of CW learning, confidence weighted online collaborative filtering learns to update the distributions of both the user vector U_a and the item vector V_b by minimizing the following unconstrained objective functions respectively:

$$\mathcal{C}_U(\mu_{u_a},\Sigma_{u_a}) = D_{KL}(\mathcal{N}(\mu_{u_a},\Sigma_{u_a})||\mathcal{N}(\mu_{u_a,t-1},\Sigma_{u_a,t-1})) + \lambda_1\ell(\mu_{u_a},V_b,r_{ab}) + \lambda_2 V_b^\top\Sigma_{u_a}V_b \quad (6)$$

$$\mathcal{C}_V(\mu_{v_b},\Sigma_{v_b}) = D_{KL}(\mathcal{N}(\mu_{v_b},\Sigma_{v_b})||\mathcal{N}(\mu_{v_b,t-1},\Sigma_{v_b,t-1})) + \lambda_1\ell(U_a,\mu_{v_b},r_{ab}) + \lambda_2 U_a^\top\Sigma_{v_b}U_a \quad (7)$$

By rewriting the KL divergence D_{KL} explicitly, we can rewrite the above as follows:

$$\mathcal{C}_U(\mu_{u_a},\Sigma_{u_a}) = \frac{1}{2}\log(\frac{\det\Sigma_{u_a,t-1}}{\det\Sigma_{u_a}})+\frac{1}{2}\mathrm{Tr}(\Sigma_{u_a,t-1}^{-1}\Sigma_{u_a})+\frac{1}{2}(\mu_{u_a,t-1}-\mu_{u_a})^\top\Sigma_{u_a,t-1}^{-1}(\mu_{u_a,t-1}-\mu_{u_a})$$

$$-\frac{k}{2}+\frac{1}{2\alpha_1}\ell(\mu_{u_a},V_b,r_{ab})+\frac{1}{2\alpha_2}V_b^\top\Sigma_{u_a}V_b \quad (8)$$

$$\mathcal{C}_V(\mu_{v_b},\Sigma_{v_b}) = \frac{1}{2}\log(\frac{\det\Sigma_{v_b,t-1}}{\det\Sigma_{v_b}})+\frac{1}{2}\mathrm{Tr}(\Sigma_{v_b,t-1}^{-1}\Sigma_{v_b})+\frac{1}{2}(\mu_{v_b,t-1}-\mu_{v_b})^\top\Sigma_{v_b,t-1}^{-1}(\mu_{v_b,t-1}-\mu_{v_b})$$

$$-\frac{k}{2}+\frac{1}{2\alpha_1}\ell(U_a,\mu_{v_b},r_{ab})+\frac{1}{2\alpha_2}U_a^\top\Sigma_{v_b}U_a \quad (9)$$

where α_1 and α_2 are used for simplicity by setting $\lambda_1 = 1/2\alpha_1$, and $\lambda_2 = 1/2\alpha_2$. It is important to note that the objective function can be decomposed into two decoupled terms: one depending on μ and the other depending on Σ. As a result, we can optimize and update the mean vector and covariance matrix separately, as shown in the following propositions.

Proposition 1 *In confidence weighted online collaborative filtering, given an observed rating pair (a,b,r_{ab}), the updating rules of Σ with respect to both RMSE and MAE are expressed as:*

$$\Sigma_{u_a} = \Sigma_{u_a,t-1} - \frac{\Sigma_{u_a,t-1}V_bV_b^\top\Sigma_{u_a,t-1}}{\alpha_2+V_b^\top\Sigma_{u_a,t-1}V_b} \quad (10)$$

$$\Sigma_{v_b} = \Sigma_{v_b,t-1} - \frac{\Sigma_{v_b,t-1}U_aU_a^\top\Sigma_{v_b,t-1}}{\alpha_2+U_a^\top\Sigma_{v_b,t-1}U_a} \quad (11)$$

The proof to the above mostly follows the similar proof as in Crammer et al. (2009b). We omit the details due to space limitation.

Proposition 2 *In confidence weighted online collaborative filtering, given an observed rating pair (a,b,r_{ab}), the updating rules of μ with respect to RMSE are expressed as:*

$$\mu_{u_a} = \mu_{u_a,t-1} + \frac{r_{ab}-V_b^\top\mu_{u_a,t-1}}{\alpha_1+V_b^\top\Sigma_{u_a,t-1}V_b}\Sigma_{u_a,t-1}V_b \quad (12)$$

$$\mu_{v_b} = \mu_{v_b,t-1} + \frac{r_{ab}-U_a^\top\mu_{v_b,t-1}}{\alpha_1+U_a^\top\Sigma_{v_b,t-1}U_a}\Sigma_{v_b,t-1}U_a \quad (13)$$

Proof We rewrite equation (8) w.r.t. RMSE and omit the terms not depending on μ_{U_a}

$$\mathcal{C}_U(\mu_{u_a}, \Sigma_{u_a}) = \frac{1}{2}(\mu_{u_a,t-1} - \mu_{u_a})^\top \Sigma^{-1}_{u_a,t-1}(\mu_{u_a,t-1} - \mu_{u_a}) + \frac{1}{2\alpha_1}(\mu_{u_a}^\top V_b - r_{ab})^2 \tag{14}$$

Taking the derivative of $\mathcal{C}_U(\mu_{u_a}, \Sigma_{u_a})$ with respect to μ_{u_a} and setting it to zero leads to

$$\mu_{u_a} = \mu_{u_a,t-1} - \frac{1}{\alpha_1}\Sigma_{u_a} V_b(\mu_{u_a}^\top V_b - r_{ab}) \tag{15}$$

We solve μ_{u_a} by taking the dot product with V_b on each side and substituting it back to (15) to obtain (13). We skip the proof for updating μ_{v_b} since it is identical to that of μ_{u_a}. ■

Proposition 3 *In confidence weighted online collaborative filtering, given an observed rating pair (a, b, r_{ab}), the updating rules of μ with respect to MAE are expressed as:*

$$\mu_{u_a} = \begin{cases} \mu_{u_a,t-1} - \lambda_1 \Sigma_{u_a,t-1} V_b & \text{if } \hat{r}_{a,b} - r_{ab} > \lambda_1 V_b^\top \Sigma_{u_a,t-1} V_b \\ \mu_{u_a,t-1} + \lambda_1 \Sigma_{u_a,t-1} V_b & \text{if } \hat{r}_{a,b} - r_{ab} < -\lambda_1 V_b^\top \Sigma_{u_a,t-1} V_b \\ \mu_{u_a,t-1} & \text{otherwise} \end{cases} \tag{16}$$

$$\mu_{v_b} = \begin{cases} \mu_{v_b,t-1} - \lambda_1 \Sigma_{v_b,t-1} U_a & \text{if } \hat{r}_{a,b} - r_{ab} > \lambda_1 U_a^\top \Sigma_{v_b,t-1} U_a \\ \mu_{v_b,t-1} + \lambda_1 \Sigma_{v_b,t-1} U_a & \text{if } \hat{r}_{a,b} - r_{ab} < -\lambda_1 U_a^\top \Sigma_{v_b,t-1} U_a \\ \mu_{v_b,t-1} & \text{otherwise} \end{cases} \tag{17}$$

Proof We rewrite equation (8) w.r.t. MAE and omit the terms not depending on μ_{U_a}

$$\mathcal{C}_U(\mu_{u_a}, \Sigma_{u_a}) = \frac{1}{2}(\mu_{u_a,t-1} - \mu_{u_a})^\top \Sigma^{-1}_{u_a,t-1}(\mu_{u_a,t-1} - \mu_{u_a}) + \lambda_1 |\mu_{u_a}^\top V_b - r_{ab}| \tag{18}$$

Taking the derivative of $\mathcal{C}_U(\mu_{u_a}, \Sigma_{u_a})$ with respect to μ_{u_a} and setting it to zero leads to

$$\mu_{u_a} = \mu_{u_a,t-1} - \lambda_1 sign(\mu_{u_a}^\top V_b - r_{ab}) \Sigma_{u_a,t-1} V_b \tag{19}$$

In the above, if $\mu_{u_a}^\top V_b - r_{ab} > 0$, we have

$$\mu_{u_a} = \mu_{u_a,t-1} - \lambda_1 \Sigma_{u_a,t-1} V_b.$$

Taking the inner product with V_b on both sides of the equation leads to

$$\mu_{u_a}^\top V_b = \mu_{u_a,t-1}^\top V_b - \lambda_1 V_b^\top \Sigma_{u_a,t-1} V_b > r_{ab}$$

Since $\mu_{u_a,t-1} V_b = \hat{r}_{a,b}$, the first case is proved. Similarly, we can prove the second case. When $|\hat{r}_{a,b} - r_{ab}| < \lambda_1 V_b^\top \Sigma_{u_a,t-1} V_b$, $|\mu_{u_a}^\top V_b - r_{ab}| = 0$. Thus, no update is needed. ■

Finally, Algorithm 1 summarizes the detailed framework of the proposed Confidence Weighted Online Collaborative Filtering (CWOCF) algorithms. Both time and space complexity of the proposed algorithm at each learning round are $O(k^2)$, where k, the dimensionality of the joint latent factor space, is typically a small constant.

Algorithm 1 Confidence Weighted Online Collaborative Filtering (**CWOCF-I**)

```
Input: dimension $k$, a sequence of rating $\{(a_t, b_t, R_{ab}), t = 1, \dots, T\}$
Initialization: Initialize a random matrix for user matrix $U \in \mathbb{R}^{n\times k}$ and item matrix $V \in \mathbb{R}^{m\times k}$, and initialize $n$ covariance matrixes to be $I$
for $t = 1, 2, \dots, T$ do
    receive rating prediction request of user $a_t$ on item $b_t$
    make prediction $\hat{r}_{a_t,b_t} = U_{a_t} V_{b_t}^\top$
    the true rating $r_{a_t,b_t}$ is revealed
    the algorithm suffers a loss $\ell(U_a, V_b, r_{a,b})$
    update $U_{a_t}$ and $V_{b_t}$ byProposition 2 (RMSE) or 3 (MAE)
    update $\Sigma_{u_a}$ and$\Sigma_{v_b}$ Proposition 1
end for
```

3.4. Efficient Update for Large-scale Applications

Although CWOCF is able to achieve significantly fast convergence rate, the computational cost in both time and space could be potentially high when k is large, making it inefficient for large-scale applications. To address the tradeoff between efficiency and efficacy, one solution is to update only the diagonals of the covariance matrixes, thus reducing both the space and time complexity to $O(k)$.

Proposition 4 *In confidence weighted online collaborative filtering, given an observed rating pair (a, b, r_{ab}), the updating rule of Σ in diagonal update setting with respect to both RMSE and MAE are expressed as:*

$$\Sigma_{u_a} = \Sigma_{u_a,t-1} - \frac{\Sigma_{u_a,t-1} \odot V_b \odot V_b \odot \Sigma_{u_a,t-1}}{\alpha_2 + V_b^\top(\Sigma_{u_a,t-1} \odot V_b)} \tag{20}$$

$$\Sigma_{v_b} = \Sigma_{v_b,t-1} - \frac{\Sigma_{v_b,t-1} \odot U_a \odot U_a \odot \Sigma_{v_b,t-1}}{\alpha_2 + U_a^\top(\Sigma_{v_b,t-1} \odot U_a)} \tag{21}$$

where $\odot$ denotes element-wise product.

To save the space cost in practice, the Σ_{u_a} and Σ_{v_b} in the diagonal update setting can be implemented by using k dimensional column vectors.

Proposition 5 *In confidence weighted online collaborative filtering, given an observed rating pair (a, b, r_{ab}), the updating rules of μ in diagonal update setting with respect to RMSE are expressed as follows:*

$$\mu_{u_a} = \mu_{u_a,t-1} + \frac{r_{ab} - V_b^\top \mu_{u_a,t-1}}{\alpha_1 + V_b^\top(\Sigma_{u_a,t-1} \odot V_b)}(\Sigma_{u_a,t-1} \odot V_b) \tag{22}$$

$$\mu_{v_b} = \mu_{v_b,t-1} + \frac{r_{ab} - U_a^\top \mu_{v_b,t-1}}{\alpha_1 + U_a^\top(\Sigma_{v_b,t-1} \odot U_a)}(\Sigma_{v_b,t-1} \odot U_a) \tag{23}$$

Algorithm 2 Fast Confidence Weighted OCF with Novel Sample Extension (**CWOCF-II**)

```
Input: a sequence of rating pairs $\{(a_t, b_t, r_{ab}), t = 1, \ldots, T\}$,
Initialization: Initialize $U = V = []$, $\Sigma_u = \Sigma_v = []$.
for $t = 1, 2, \ldots, T$ do
  receive rating prediction request of user $a_t$ on item $b_t$
  if user $a_t$ is new then
    initialize $U_{a_t}$ as a random vector.
    expand the user matrix $U$ as $U = [U; U_{a_t}]$
    expand the covariance matrix as $\Sigma_u = [\Sigma_u; \mathbf{1}]$
  end if
  if item $b_t$ is new then
    initialize $V_{b_t}$ as a random vector.
    expand the item matrix $V$ as $V = [V; V_{b_t}]$
    expand the covariance matrix as $\Sigma_v = [\Sigma_v; \mathbf{1}]$
  end if
  make prediction $\hat{r}_{a_t,b_t} = U_{a_t} V_{b_t}^\top$
  the true rating $r_{a_t,b_t}$ is revealed
  the algorithm suffers a loss $\ell(U_a, V_b, r_{a,b})$
  update $U_{a_t}$, $V_{b_t}$, $\Sigma_{u_a}$ and $\Sigma_{v_b}$ by Proposition 4, 5(RMSE) or 6 (MAE)
end for
```

Proposition 6 *In confidence weighted online collaborative filtering, given an observed rating pair (a, b, r_{ab}), the updating rules of μ with respect to MAE are expressed as:*

$$\mu_{u_a} = \begin{cases} \mu_{u_a,t-1} - \lambda_1 \Sigma_{u_a,t-1} \odot V_b & \textit{if } \; \hat{r}_{a,b} - r_{ab} > \lambda_1 V_b^\top (\Sigma_{u_a,t-1} \odot V_b) \\ \mu_{u_a,t-1} + \lambda_1 \Sigma_{u_a,t-1} \odot V_b & \textit{if } \; \hat{r}_{a,b} - r_{ab} < -\lambda_1 V_b^\top (\Sigma_{u_a,t-1} \odot V_b) \\ \mu_{u_a,t-1} & \textit{otherwise} \end{cases} \tag{24}$$

$$\mu_{v_b} = \begin{cases} \mu_{v_b,t-1} - \lambda_1 \Sigma_{v_b,t-1} \odot U_a & \textit{if } \; \hat{r}_{a,b} - r_{ab} > \lambda_1 U_a^\top (\Sigma_{v_b,t-1} \odot U_a) \\ \mu_{v_b,t-1} + \lambda_1 \Sigma_{v_b,t-1} \odot U_a & \textit{if } \; \hat{r}_{a,b} - r_{ab} < -\lambda_1 U_a^\top (\Sigma_{v_b,t-1} \odot U_a) \\ \mu_{v_b,t-1} & \textit{otherwise} \end{cases} \tag{25}$$

The proofs to the above propositions are straightforward by simplifying the Proposition 1 to 3 from full matrix update to diagonal update. We omits the details due to space limitation.

3.5. Novel User/Item Extension

In the previous discussion, we assume that the number of users n and the number of items m are known in advance and fixed during the online learning process. This however is not realistic in a real-world online application, where new users or new products could be added to the system at any time. In this part, we extend the previous CWOCF algorithm to handle novel user/item extension. Due to the advantage of online algorithms, the novel sample extension can be done in an easy manner by expanding the parameters of the distributions learned so far. We summarize the detailed algorithm together with the diagonal update in Algorithm 2.

4. Experimental Result

In this section, we evaluate the empirical performance of the proposed CWOCF algorithm for online collaborative filtering tasks. In our experiments, we adapt the similar evaluation protocol of online learning tasks for collaborative filtering.

4.1. Compared Algorithms

We compare the proposed CWOCF algorithms with two model-based online collaborative filtering algorithms as follows:

- "OCF": the Online Collaborative Filtering (Abernethy et al., 2007) by online gradient descent method;
- "DA-OCF": the Dual-Averaging optimization method for online collaborative filtering (Ling et al., 2012);
- "CWOCF-I": the proposed Confidence Weighted Online Collaborate Filtering algorithm with full covariance matrix update;
- "CWOCF-II": the proposed Confidence Weighted Online Collaborate Filtering algorithm with diagonal covariance matrix update and novel user or item extension.

We note that we cannot compare with many batch collaborative filtering algorithms since they are not designed for online CF tasks, making them non-scalable and impractical under online learning settings.

4.2. Experimental Testbed and Setup

To comprehensively examine the empirical performance, we conduct the experiments on four publicly available datasets, which have been widely used for benchmark evaluation of collaborative filtering in literature. All of the datasets can be downloaded from the GroupLens Research webset [1]. We first test the compared algorithms on the two relatively smaller datasets, "Movielens 100k" and "HetRec 2011", for evaluation on latent factor dimensionality, time efficiency, and prediction accuracy. We then evaluate the large-scale experiments on the rest two larger datasets, i.e., "Movielens 1M" and "Movielens 10M". Table 1 summarizes the statistics of the datasets, where the "rating matrix density" is defined as the fraction of observed ratings out of the total number of elements $(m \cdot n)$ in the rating matrix.

Table 1: Statistics of the Datasets Used in Experiments

Datasets	# Ratings	#Items	#Users	Rating Scale	rating matrix density
Movielens 100k	100,000	1,682	943	1-5	6.3%
HetRec 2011	855,598	10,109	2,113	1-5	4.0%
Movielens 1M	1,000,209	3,900	6,040	1-5	4.2%
Movielens 10M	10,000,054	10,681	71,567	1-5	1.3%

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

For parameter settings, to enable fair comparisons of different algorithms, we follow a standard approach of parameter selection for online learning experiments. In particular, all of the parameters including the λ and τ in OCF, the α_1 and α_2 in CWOCF and the λ_U and λ_V in DA-OCF were found automatically by searching from a single experiment on a random permutation of each dataset, except that the latent factor dimensionality is fixed to a constant (5 or 10) for all the algorithms. The search range for λ, τ, λ_U and λ_V is from 10^{-5} to 10^{-1} and the range for α_1 and α_2 is from 1 to 100. After all the parameters are chosen, all the experiments are conducted over 20 runs of different random permutations for each dataset. All the experimental results are reported by averaging over these 20 runs. For performance metrics, we evaluate the performance of online collaborative filtering algorithms by measuring their scores of online Root Mean Square Error (RMSE) and online Mean Absolute Error (MAE).

4.3. Evaluation on Medium-Scale Datasets

We first evaluate the algorithms on two medium-scale datasets under two settings of the latent factor dimensionality ($k = 5$ and $k = 10$). Table 2 summarizes the average performance of the compared algorithms. We can draw some observations as follows.

Table 2: Performance Evaluation, where k is the rank parameter for matrix U and V. The **bold** elements indicate the best performance for each setting.

ML	k=5				k=10			
100k	RMSE	time	MAE	time	RMSE	time	MAE	time
OCF	1.1151±0.0014	0.33	0.9463±0.0015	0.35	1.0461±0.0007	0.33	0.8558±0.0008	0.35
DAOCF	1.2427±0.0160	0.67	0.9815±0.0170	0.67	1.2231±0.0100	0.67	0.9631±0.0075	0.70
CWOCFI	1.0439±0.0010	6.40	0.8397±0.0014	5.44	**1.0103**±0.0005	7.27	0.8091±0.0006	6.12
CWOCFII	**1.0314**±0.0009	1.58	**0.8172**±0.0010	1.92	1.0106±0.0011	1.65	**0.8045**±0.0010	2.07
Het	k=5				k=10			
Rec	RMSE	time	MAE	time	RMSE	time	MAE	time
OCF	0.9234±0.0003	2.96	0.7484±0.0003	3.02	0.8803±0.0001	3.13	0.6891±0.0002	3.20
DAOCF	1.0462±0.0053	6.06	0.7527±0.0065	6.01	1.0871±0.0053	6.20	0.7659±0.0034	6.27
CWOCFI	0.8736±0.0015	55.9	0.6686±0.0010	47.4	**0.8473**±0.0013	65.1	**0.6499**±0.0009	55.3
CWOCFII	**0.8732**±0.0002	14.9	**0.6656**±0.0002	17.8	0.8603±0.0003	18.9	0.6582±0.0002	20.3

First of all, compared with the existing OCF and DA-OCF approaches, we observe that the proposed CWOCF algorithms achieve significantly better performance of smaller RMSE and MAE values for all the cases. This shows that the proposed learning strategy is more effective than the existing first order online gradient descent approaches.

Second, when examining the standard deviation results, we observe that CWOCF is almost the most stable algorithm to different random permutations of training samples, while the DA-OCF is relatively sensitive since it only learns the mean of the weight vector at each iteration and thus loses lots of information. This is an important strength of

CWOCF since it learns the Gaussian distribution by capturing both first order and second order information, leading to more precise and stable update in the entire online learning process.

Third, the performance gap between CWOCF and the first order algorithms in low dimensional space tends to be more significant than that in high dimensional space. This indicates that OCWOCF is able to learn effectively even in very low dimensional space while the first order algorithms would suffer significantly when the dimensionality is too small.

Fourth, when comparing the two variants of the proposed CWOCF algorithms, i.e., the full confidence matrix update and the diagonal update, we found that they generally have very comparable performance, while CWOCF-II is significantly faster, only a few times slower than the first order algorithms. The gain of CWOCF-II is mainly because it only updates the diagonals of confidence matrixes with time complexity $O(k)$ instead of $O(k^2)$. In addition, we found that the performance of CWOCF-II could even exceed that of CWOCF-I. We conjecture that this is primarily because CWOCF-I might suffer from the overfitting if the model is over-complex. Thus, we believe CWOCF-II is the more applicable solution for solving large-scale applications.

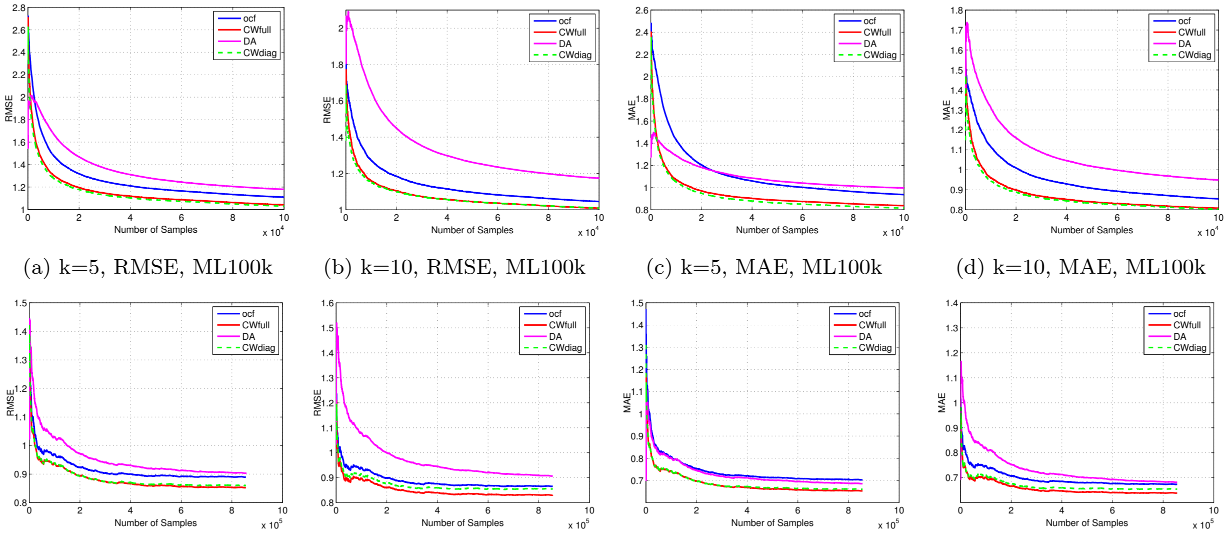
To further inspect the details of online CF performance, Figure 1 also shows the online performance convergence of all the compared algorithms in the entire online learning process. The result again shows that CWOCF algorithms outperform the first order algorithms especially at the beginning of the online learning process. We note that is extremely important and beneficial to many real-world online recommender systems because the amount of collected ratings could be very scarce at the beginning of the deployment of the recommender system. Thus, a highly effective online updating algorithm is particularly important to tackle the sparsity challenge of online collaborative filtering tasks.

4.4. Evaluation on the Latent Factor Dimensionality

The second experiment is to examine how the latent factor dimensionality k affects the learning performance of the proposed algorithms. Figure 2 summarizes the performance of the three compared online algorithms on a series of different latent factor dimensions k ranging from 4 to 20. Several interesting observations can be drawn from the experimental results.

First of all, we found when the latent factor dimensionality k is very small (e.g., smaller than 5), increasing the value of k leads to the improvement of the learning performance. This is reasonable as a model with too small value of k could suffer from *underfitting*, i.e., the model is too simple to make powerful prediction. Second, we found that when k is large than some threshold (e.g., larger than 12 on the ML100 and HetRec datasets), increasing the value of k could lead to the degrade of the overall performance. This is because the model with a large value of k could suffer from *overfitting* from insufficient training data, i.e., the model is too complex to learn from the highly sparse rating matrix.

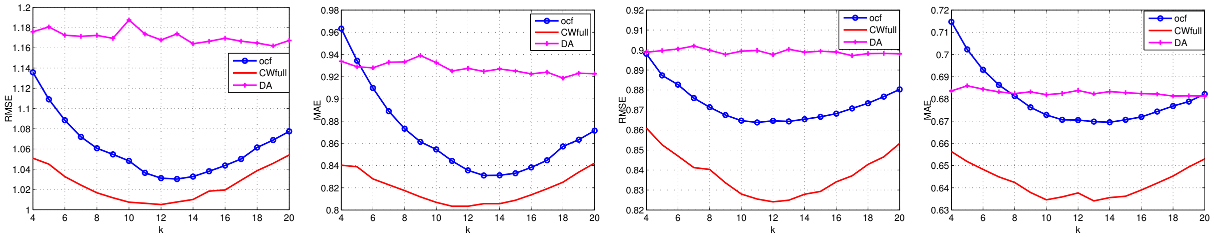
The above observations indicate that choosing the appropriate value of latent factor dimensionality is essentially a tradeoff between underfitting and overfitting with respect to the provided dataset.



(a) k=5, RMSE, ML100k (b) k=10, RMSE, ML100k (c) k=5, MAE, ML100k (d) k=10, MAE, ML100k

(a) k=5, RMSE, HetRec (b) k=10, RMSE, HetRec (c) k=5, MAE, HetRec (d) k=10, MAE, HetRec

Figure 1: Convergence evaluation of online collaborative filtering algorithms (best viewed in color).



(a) RMSE, ML100k (b) MAE, ML100k (c) RMSE, HetRec (d) MAE,HetRec

Figure 2: Performance evaluation on different values of latent factor dimensionality k.

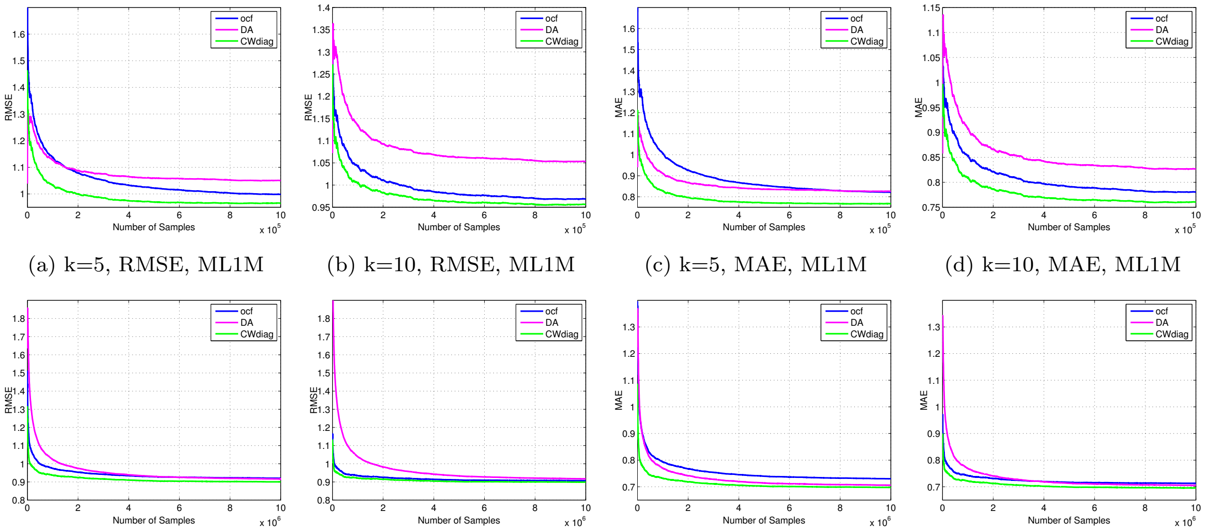
4.5. Evaluation on Large-scale Datesets

Our last experiment is to evaluate the proposed CWOCF-II algorithm on two large-scale datesets: “Movielens 1M” and “Movielens 10M”. The empirical evaluation results are summarized in Table 3 and Figure 3. From the results, we can observe that CWOCF-II converges considerably faster, obtains significantly lower values of RMSE and MAE, while runs only a few times slower than the first order algorithms. These encouraging results again demonstrate that the proposed algorithm can scale well for large-scale datasets.

In addition, by comparing the performance of the CWOCF algorithms across different datasets, we found that when the dimensionality k increases from 5 to 10, the values of RMSE and MAE on the denser dataset ML100k can decrease for about 2%, while they only decrease for 0.7% in the sparse dataset of ML10M. This again shows that CWOCF benefits less from increasing the dimensionality on sparser datasets, which again validates our analysis in last section.

Table 3: Performance Evaluation on Large-Scale Datasets. The **bold** element indicates the best performance for each setting.

ML	k=5				k=10			
1m	RMSE	time	MAE	time	RMSE	time	MAE	time
OCF	1.0328±0.0002	3.42	0.8640±0.0003	3.58	0.9774±0.0001	3.59	0.7897±0.0002	3.75
DAOCF	1.1136±0.0061	7.35	0.8652±0.0037	6.96	1.1068±0.0030	7.43	0.8576±0.0021	7.32
CWOCFII	**0.9653**±0.0003	16.1	**0.7631**±0.0002	19.3	**0.9580**±0.0003	17.5	**0.7609**±0.0003	20.6
ML	k=5				k=10			
10M	RMSE	time	MAE	time	RMSE	time	MAE	time
OCF	0.9476±0.0001	36.8	0.7591±0.0001	37.1	0.9192±0.0001	39.3	0.7239±0.0001	39.8
DAOCF	1.0764±0.0049	74.2	0.7638±0.0013	74.6	1.0774±0.0032	79.7	0.7632±0.0013	80.5
CWOCFII	**0.9096**±0.0001	321	**0.7043**±0.0001	343	**0.9033**±0.0001	515	**0.7014**±0.0001	540



(a) k=5, RMSE, ML1M (b) k=10, RMSE, ML1M (c) k=5, MAE, ML1M (d) k=10, MAE, ML1M

(a) k=5, RMSE, ML10M (b) k=10, RMSE, ML10M (c) k=5, MAE, ML10M (d) k=10, MAE, ML10M

Figure 3: Convergence Evaluation on Large Scale Datesets (best viewed in color).

5. Conclusions

In this paper, we proposed a novel framework of Second Order Online Collaborative Filtering in building realistic solutions for online recommender systems. In particular, we presented the Confidence Weighted Online Collaborative Filtering (CWOCF) method, which attempts to maintain the distributions of both the user and item vectors by updating both the first order and the second order information of the model in the online learning process. In contrast to the existing first order algorithms which only exploit the first order information, our algorithms converge significantly faster and thus achieve much lower values of RMSE and MAE. We conducted extensive experiments on four different data sets of different sizes and sparsity levels, in which the promising results validate the effectiveness of the proposed algorithms. Future work will address the theoretical analysis of the CWOCF algorithms and exploring other second order algorithms for improving the performance.

Acknowledgments

This research was supported by MOE tier 1 grant (RG33/11).

References

Jacob Abernethy, Kevin Canini, John Langford, and Alex Simma. Online collaborative filtering. *University of California at Berkeley, Tech. Rep*, 2007.

Muqeet Ali, Christopher C Johnson, and Alex K Tang. Parallel collaborative filtering for streaming data. *University of Texas Austin, Tech. Rep*, 2011.

Nicolò Cesa-Bianchi, Alex Conconi, and Claudio Gentile. A second-order perceptron algorithm. *SIAM Journal on Computing*, 34(3):640–668, 2005.

Koby Crammer and Yoram Singer. Pranking with ranking. *Advances in neural information processing systems*, 14:641–647, 2001.

Koby Crammer, Mark Dredze, and Alex Kulesza. Multi-class confidence weighted algorithms. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 496–504. Association for Computational Linguistics, 2009a.

Koby Crammer, Alex Kulesza, Mark Dredze, et al. Adaptive regularization of weight vectors. *Advances in Neural Information Processing Systems*, 22:414–422, 2009b.

Mark Dredze, Koby Crammer, and Fernando Pereira. Confidence-weighted linear classification. In *Proceedings of the 25th international conference on Machine learning*, pages 264–271. ACM, 2008.

Mark Dredze, Alex Kulesza, and Koby Crammer. Multi-domain learning by confidence-weighted parameter combination. *Machine Learning*, 79(1-2):123–149, 2010.

Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2):133–151, 2001.

Edward F Harrington. Online ranking/collaborative filtering using the perceptron algorithm. In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, volume 20, page 250, 2003.

Luheng He, Nathan N Liu, and Qiang Yang. Active dual collaborative filtering with both item and attribute feedback. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76–80, 2003.

Guang Ling, Haiqin Yang, Irwin King, and Michael R Lyu. Online learning for collaborative filtering. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pages 1–8. IEEE, 2012.

Nathan N Liu, Min Zhao, Evan Xiang, and Qiang Yang. Online evolutionary collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 95–102. ACM, 2010.

Prem Melville, Raymod J Mooney, and Ramadass Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *Proceedings of the National Conference on Artificial Intelligence*, pages 187–192. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2002.

Koji Miyahara and Michael J Pazzani. Improvement of collaborative filtering with the simple bayesian classifier. *IPSJ Journal*, 43(11):3429–3437, 2002.

Yurii Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical programming*, 120(1):221–259, 2009.

Francesco Orabona and Koby Crammer. New adaptive algorithms for online classification. 2010.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM, 2001.

Badrul M Sarwar, George Karypis, Joseph Konstan, and John Riedl. Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. In *Proceedings of the fifth international conference on computer and information technology*, volume 1, 2002.

Xiaoyuan Su and Taghi M Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009:4, 2009.

Lyle H Ungar and Dean P Foster. Clustering methods for collaborative filtering. In *AAAI Workshop on Recommendation Systems*, number 1, 1998.

Jialei Wang, Peilin Zhao, and Steven CH Hoi. Exact soft confidence-weighted learning. *arXiv preprint arXiv:1206.4612*, 2012.