Collaborative Filtering with Noisy Ratings

Dongsheng Li* Chao Chen* Zhilin Gong†‡ Tun Lu†‡ Stephen M. Chu* Ning Gu†‡

Abstract

User ratings on items are noisy in real-world recommender systems, which raises challenges to matrix approximation (MA)-based collaborative filtering (CF) algorithms — the learned models will be easily biased to the noisy training data and yield low generalization performance. This paper proposes a noise-resilient matrix approximation (NORMA) method, which can achieve less biased matrix approximation and thus more accurate collaborative filtering. In NORMA, an adaptive weighting strategy is proposed to decrease the gradient updates of noisy ratings, so that the learned MA models will be less prone to the noisy ratings. Theoretical analyses show that NORMA can achieve better generalization performance than standard matrix approximation methods. Experimental studies on real-world datasets demonstrate that NORMA can outperform state-of-the-art matrix approximation-based collaborative filtering methods in recommendation accuracy.

Keywords: collaborative filtering, matrix approximation.

1 Introduction

Collaborative filtering (CF) is one of the most popular algorithms in real-world recommender systems [1, 22]. Recent studies demonstrate that matrix approximation (MA) methods [6, 16, 21, 19, 28, 31] have achieved state-of-the-art accuracy in collaborative filtering on rating prediction task. Generally, MA methods first learn the feature vectors for users and items from a set of observed user-item ratings, then the obtained user/item feature vectors are employed for recommendation on unobserved user-item ratings [16]. Since the MA models are learned from the observed ratings, so that the quality of the observed ratings is critical to the success of recommender systems.

In real-world recommender systems, user-item ratings are very noisy [3, 8, 21]. Studies show that only around 60% of user ratings are kept the same when the users are asked to re-rate items [8], which indicates that a large fraction of observed user-item ratings cannot accurately represent users' true interests. Similar phenomenons have also been observed by other studies [2, 3, 14, 30]. This will introduce challenges to matrix approximation methods, because the true ratings and noisy ratings will be considered equally important during model learning in standard matrix approximation methods. As a result, the learned matrix approximation models will be easily biased towards the noisy training data and thus achieve low generalization performance in practice. To achieve unbiased matrix approximation and deliver high quality recommendations based on noisy ratings, matrix approximation methods should capture the diverse noises in user-item ratings and alleviate overreaction to the noises during model learning.

In this paper, we propose a noise-resilient matrix approximation method, namely NORMA, to achieve less biased matrix approximation and thus more accurate collaborative filtering with noisy user-item ratings. Firstly, each observed user-item rating is modelled as a Gaussian random variable with independent variance, so that the diverse noises in the user-item ratings can be modeled independently. Secondly, an adaptive weighting strategy is proposed to adjust the learning steps based on the estimated levels of noises for each rating, so that the ratings with larger estimated noises can be trained with smaller learning steps to alleviate overreaction. Theoretical analyses show that NORMA can achieve better generalization performance, i.e., sharper generalization error bound and expected risk bound, compared with standard matrix approximation methods. Experimental studies on real-world datasets demonstrate that NORMA can outperform six state-of-the-art matrix approximation-based collaborative filtering methods in recommendation accuracy.

2 Problem Formulation

This section first analyzes the noise issue in the user-item ratings from different perspectives. Then, we model and analyze the rating noises from a probabilistic view.

2.1 Noises in User-Item Ratings Collaborative filtering methods rely on historical user-item ratings to predict users' interests on unseen items, so that the reliability of user-item ratings are crucial to the success of CF methods [23, 3]. However, real-world user-item ratings are noisy, and both *malicious noise* and *natural noise* exist in today's recommender system databases [26]. This paper focuses on *natural noises*, which are harder to detect and model in collaborative filtering [26].

Reasons. The *natural noises* are mainly introduced due to the following reasons. 1) User preferences are hard to

*IBM Research - China. Email: {ldsli, cchao, schu}@cn.ibm.com

†School of Computer Science, Fudan University, China. Email: {17210240091, tunlu, ninggu}@fudan.edu.cn

‡Shanghai Key Laboratory of Data Science, Fudan University, China.

quantify [12]. It is difficult for users to accurately measure their preferences and consistently assign their preferences to ratings. 2) Granularity of rating scales. Users like finer-grained scale the best [8], but most recommender systems do not support continuous rating scale. Therefore, noises arise when mapping user opinions into discrete ratings. 3) Memory Loss. Users store a complex set of feelings about items, and they may not clearly remember all these feelings when they are asked to rate the items [25]. 4) Other complex reasons [3, 12], e.g., mood, context, user rating speed, item orders, etc.

Severity. Cosley et al. [8] showed that users only keep about 60% of ratings the same as their previous ratings when they are asked to re-rate the same movies. The studies of Jones et al. [12] showed that the stability of user ratings is around 63% and the stability of user comparing items is around 82%. These rating noises can affect the accuracies of collaborative filtering methods. Cosley et al. [8] observed statistically significant MAE differences when CF algorithm runs on users' original ratings and their ratings on new scales. Amatriain et al. [2] showed that recommendation accuracy can be affected in different conditions, and the RMSE variations can be as high as 40%.

In summary, noises widely exist in the user-item rating data of today's recommender systems, and the accuracies of collaborative filtering algorithms are affected due to the existence of noises. Therefore, it is necessary to design collaborative filtering algorithms which are resilient to the rating noises.

2.2 A Probabilistic View of Rating Noises The PMF method [29] first describes user-item ratings as Gaussian random variables as follows:

$$R_{i,j} \sim \mathcal{N}(R_{i,j}|\hat{R}_{i,j}, \sigma^2), \tag{2.1}$$

where $R_{i,j}$ is the observed rating of the i-th user on the j-th item and $\hat{R}_{i,j}$ is the predicted rating. σ^2 is the variance of the Gaussian distribution, which is considered the same across all observed user-item ratings during model learning [29]. However, using the same variance for all user-item ratings may not be appropriate, because users are very consistent with extreme ratings, e.g., 1 or 5 on a scale of 1 to 5, but less consistent with moderate ratings, e.g., 2, 3 and 4 [2, 12]. Therefore, we use different variances for different user-item ratings as follows:

$$R_{i,j} \sim \mathcal{N}(R_{i,j}|\hat{R}_{i,j}, \sigma_{i,j}^2), \tag{2.2}$$

where $\sigma_{i,j}^2$ stands for the variance of rating $R_{i,j}$. Therefore, we can assume that $R_{i,j} - \hat{R}_{i,j}$ follows a 0 mean Gaussian distribution as follows:

$$R_{i,j} - \hat{R}_{i,j} \sim \mathcal{N}(R_{i,j} - \hat{R}_{i,j}|0, \sigma_{i,j}^2). \tag{2.3}$$

2.3 Confidence Interval of $\sigma_{i,j}^2$ Since users only rate an item once in most recommender systems, it is difficult to estimate the rating variances directly. Here, we estimate the confidence interval of $\sigma_{i,j}^2$ using the following theorem.

THEOREM 2.1. *Given one observation X from $\mathcal{N}(0, \sigma^2)$ with σ unknown, for any $0 < \delta < 1$, with a confidence level of at least $1 - \delta$, the confidence interval for σ^2 is $[0, 2X^2/\pi\delta^2)$.*

Proof. Since the probability density of $\mathcal{N}(0, \sigma^2)$ distribution does not exceed $1/\sqrt{2\pi}\sigma$, we have $\Pr(|X| \leq a) \leq 2a/\sqrt{2\pi}\sigma$ for any $a > 0$. Then, for any $\delta \in (0, 1)$, we have

$$\begin{aligned} \delta &\geq \Pr(|X| \leq \sqrt{2\pi}\sigma\delta/2) \\ &= \Pr(X^2 \leq \pi\sigma^2\delta^2/2) = \Pr(\sigma^2 \geq 2X^2/\pi\delta^2). \end{aligned}$$

Thus, we can claim that $\Pr(\sigma^2 < 2X^2/\pi\delta^2) \geq 1 - \delta$.

Based on the above theorem, we can use its confidence level bound to approximately estimate $\sigma_{i,j}^2$ as follows:

$$\sigma_{i,j}^2 \approx \frac{2c(R_{i,j} - \hat{R}_{i,j})^2}{\pi\delta^2}, \tag{2.4}$$

where c is a predefined constant to control the range of $\sigma_{i,j}^2$ and δ is a predefined constant to control the confidence level of estimating $\sigma_{i,j}^2$. One simple mechanism is to choose c as the ratio of $\pi\delta^2$, so that the computation will be easier because we only have one hyper-parameter to tune in estimating $\sigma_{i,j}^2$, i.e., $c' = 2c/\pi\delta^2$.

3 Noise-Resilient Matrix Approximation

In this section, we first define the optimization problem of the proposed NORMA method. Then, we propose how to adaptively optimize the targeted problem considering different levels of noises on different ratings. At last, we analyze the convergence rate of solving the proposed weighted optimization problem using SGD.

3.1 The Optimization Problem The optimization objective of PMF [29] can be described as a least square problem with L_2 regularization. By replacing their universal variance with individual variances, the new optimization objective can be described as follows:

$$\sum_{i,j\in\Omega} \frac{1}{\sigma_{i,j}^2}(R_{i,j} - U_i V_j^T)^2 + \frac{1}{\sigma_U^2}\sum_i ||U_i||^2 + \frac{1}{\sigma_V^2}\sum_j ||V_j||^2.$$

Here, $R \in \mathbb{R}^{m\times n}$, $U \in \mathbb{R}^{m\times r}$, and $V \in \mathbb{R}^{n\times r}$, where m is the number of users, n is the number of items, and r is the rank. Ω is the set of all observed entries in R. U_i (V_j) is the feature vector of i-th user (j-th item). $\sigma_{i,j}^2$ stands for the variance of rating $R_{i,j}$, which can be estimated by Equation 2.4. σ_U^2 and σ_V^2 are the variances for user features

and item features, respectively. Here, we can regard $1/\sigma_{i,j}^2$ as the weight for rating $R_{i,j}$, and regard $1/\sigma_U^2$ and $1/\sigma_V^2$ as part of the L_2 coefficients. Then, the above optimization objective can be converted into a weighted least square problem with L_2 regularization as follows:

$$\sum_{i,j\in\Omega} W_{i,j}(R_{i,j} - U_i V_j^T)^2 + \mu(\sum_i ||U_i||^2 + \sum_j ||V_j||^2).$$

$W_{i,j}$ is the weight for rating $R_{i,j}$, which is related to $1/\sigma_{i,j}^2$. μ is the L_2 regularization coefficient. Then, the user features and item features can be learned by stochastic gradient descent (SGD), in which the $(t+1)$-th update rules with regard to rating $R_{i,j}$ can be described as follows:

$$U_i^{(t+1)} \leftarrow U_i^{(t)} - \lambda(2W_{i,j}^{(t)}(\hat{R}_{i,j}^{(t)} - R_{i,j})V_j^{(t)} + 2\mu U_i^{(t)}),$$
$$V_j^{(t+1)} \leftarrow V_j^{(t)} - \lambda(2W_{i,j}^{(t)}(\hat{R}_{i,j}^{(t)} - R_{i,j})U_i^{(t)} + 2\mu V_j^{(t)}).$$

λ stands for the learning rate in the above update rules.

3.2 The Adaptive Weighting Strategy In this paper, we define $W_{i,j}$ as a function of $\sigma_{i,j}^2$. Two main challenges should be addressed in defining the function: 1) ratings with larger estimated variances will be given smaller weights and vice versa; and 2) $W_{i,j}$ should not be too large or smaller, otherwise the learning process will diverge or converge very slowly. Therefore, we propose to use a sigmoid function with bias to define $W_{i,j}$ as follows:

$$W_{i,j} = \alpha S(-c'(R_{i,j} - \hat{R}_{i,j})^2) + (1-\alpha), \tag{3.5}$$

where c' is a predefined constant to estimate $\sigma_{i,j}^2$ from $(R_{i,j} - \hat{R}_{i,j})^2$. $S(\cdot)$ is the standard sigmoid function, so that the range of $W_{i,j}$ can be bounded when $(R_{i,j} - \hat{R}_{i,j})^2$ varies in $[0, +\infty)$. $\alpha \in (0,1)$ is a scaling coefficient to control to range of $W_{i,j}$. Therefore, the range of $W_{i,j}$ will be bounded by $(1-\alpha, 1-\frac{\alpha}{2}]$, because $S(x) \in (0, \frac{1}{2}]$ when $x \leq 0$.

Note that the value of $W_{i,j}$ will vary for different iterations because the estimation of $\sigma_{i,j}^2$, i.e., $c'(R_{i,j} - \hat{R}_{i,j})^2$, will change when the model learns more accurate $\hat{R}_{i,j}$. Therefore, we call this adaptive weighting strategy, which means the weights will adaptively change with the learned model. In addition, the value of $W_{i,j}$ will slightly increase when $(R_{i,j} - \hat{R}_{i,j})^2$ gets smaller with increasing number of epochs, which can naturally address the issue of infinitely small learning rate after large number of epochs in many existing adaptive learning rate methods, e.g., AdaGrad [9].

3.3 Convergence Rate Analysis Minimizing the proposed weighted mean square loss can be regarded as minimizing the standard mean square loss with adaptive learning rates. Therefore, we can analyze the convergence rate of NORMA in the view of adaptive learning rates, then the analysis can be naturally applied to the weighted least square problem of NORMA. Assume that we want to minimize $f(w)$, where $f(\cdot)$ stands for the mean square loss function and w is a model (U or V in our case). Then, we can know that $f(\cdot)$ is strongly convex, i.e., there exists a positive number $l > 0$ such that $f(w) - f(w') \geq \langle \nabla f(w'), w - w' \rangle + \frac{l}{2}||w - w'||^2$. The convergence rate of the NORMA method using adaptive weighting strategy can be bounded in the following Theorem 3.1.

THEOREM 3.1. *Assuming that the gradients of $f(w)$ satisfy that $\mathbb{E}(||g||^2) \leq G^2$ for all w, where $\mathbb{E}(g) = \nabla f(w)$. Choose step size $\eta_t = \frac{1}{(1-\alpha)tl}$ for the t-th iteration. Then, we have $\mathbb{E}(||w_t - w^*||^2) \leq \max\{||w_1 - w^*||^2, \frac{(1-\alpha/2)^2 G^2}{(1-\alpha)^2 l^2}\}/t$.*

Proof. Let W_t be the weight for the t-th iteration. We have

$$\mathbb{E}(||w_{t+1} - w^*||^2) = \mathbb{E}(||w_t - \eta_t W_t g_t - w^*||^2) = \mathbb{E}(||w_t - w^*||^2) - 2\eta_t \mathbb{E}\langle W_t g_t, w_t - w^* \rangle + \eta_t^2 \mathbb{E}(||W_t g_t||^2). \tag{3.6}$$

Since $W_t \in (1-\alpha, 1-\frac{\alpha}{2}]$, we know

$$\mathbb{E}\langle W_t g_t, w_t - w^* \rangle \geq (1-\alpha)\mathbb{E}\langle g_t, w_t - w^* \rangle. \tag{3.7}$$
$$\mathbb{E}(||W_t g_t||^2) \leq (1 - \frac{\alpha}{2})^2 \mathbb{E}(||g_t||^2). \tag{3.8}$$

Based on the strongly convex property of f, we have

$$\langle \nabla f(w_t), w_t - w^* \rangle \geq l||w_t - w^*||^2. \tag{3.9}$$

Then, by combining the above Inequalities 3.7 – 3.9 into Equation 3.6, we have

$$\begin{aligned} &\mathbb{E}(||w_{t+1} - w^*||^2) \\ \leq &(2\alpha - 1)\eta_t l)\mathbb{E}(||w_t - w^*||^2) + \eta_t^2 (1 - \frac{\alpha}{2})^2 G^2 \\ \leq &(1 - \frac{2}{t})\mathbb{E}(||w_t - w^*||^2) + \frac{(1-\frac{\alpha}{2})^2 G^2}{(1-\alpha)^2 l^2 t^2}. \end{aligned} \tag{3.10}$$

From Inequality 3.10, we know that

$$\mathbb{E}(||w_1 - w^*||^2) \leq \frac{\max\{||w_1 - w^*||^2, \frac{(1-\frac{\alpha}{2})^2 G^2}{(1-\alpha)^2 l^2}\}}{1}.$$

We can adopt induction to prove our results, in which we prove that the above inequality holds for $t+1$ if the above inequality holds for t. Let us assume that $C = \max\{||w_1 - w^*||^2, \frac{(1-\frac{\alpha}{2})^2 G^2}{(1-\alpha)^2 l^2}\}$. Assuming that $\mathbb{E}(||w_t - w^*||^2) \leq C/t$, then we have

$$\begin{aligned} &\mathbb{E}(||w_{t+1} - w^*||^2) \\ \leq &(1 - \frac{2}{t})\mathbb{E}(||w_t - w^*||^2) + \frac{(1-\frac{\alpha}{2})^2 G^2}{(1-\alpha)^2 l^2 t^2} \\ \leq &(1 - \frac{2}{t})\frac{C}{t} + \frac{C}{t^2} \leq (\frac{1}{t} - \frac{1}{t^2})C \leq \frac{C}{t+1}. \end{aligned}$$

The above result shows that the inequality also holds for $t+1$ if it holds for t, which completes the proof.

Theorem 3.1 proves that minimizing the proposed weighted mean square loss using SGD can achieve a convergence rate of $O(1/t)$, i.e., the converge rate of SGD will not be affected after weighting.

4 Generalization Performance Analysis

This section theoretically analyzes the generalization performance of the proposed method by deriving the generalization error bound and expected risk bound of NORMA and comparing with those of standard matrix approximation method.

4.1 Preliminaries We adopt the uniform stability theory [5] to analyze the generalization error bound of the proposed method, which can measure how a learning algorithm perturbs with changes in the input samples. Here, we adopt the uniform stability definition from Hardt et al. [10], which is based on randomized learning algorithms.

DEFINITION 1. *[Uniform Stability [10]] Let S and S' be two samples which differ in at most one example. We say randomized learning algorithm A is ϵ-uniformly stable if we have*

$$\sup_x \mathbb{E}_A(f(A(S);x) - f(A(S');x)) \leq \epsilon.$$

Hardt et al. [10] proved that generalization error can be bounded by uniform stability bound, so that we can analyze the generalization performance of NORMA by analyzing its uniform stability bound. The uniform stability bound of solving the least square problem using standard SGD method can be bounded as follows [10]:

THEOREM 4.1. *Let $f : \Phi \to \mathbb{R}$ be the convex square loss function, and assume that $||\nabla f(\cdot;x)|| \leq L$ (L-Lipschitz) and $||\nabla f(w;x) - \nabla f(w';x)|| \leq \beta||w - w'||$ (β-smooth) for all $x \in X$ and $w, w' \in \Phi$. Suppose that we run SGD on samples with N examples with the t-th step size $\eta_t \leq 2/\beta$ for totally T steps. Then, its uniform stability can be bounded by $\epsilon_{stab} \leq \frac{2L^2}{N}\sum_{t=1}^{T}\eta_t$.*

4.2 Generalization Error Bound The uniform stability bound of NORMA is derived in the following Theorem.

THEOREM 4.2. *Let $f : \Phi \to \mathbb{R}$ be the convex square loss function, and assume that $||\nabla f(\cdot;x)|| \leq L$ (L-Lipschitz) and $||\nabla f(w;x) - \nabla f(w';x)|| \leq \beta||w - w'||$ (β-smooth) for all $x \in X$ and $w, w' \in \Phi$. Suppose that we run SGD on samples with N examples with the t-th step size $W_t\eta_t \leq 2/\beta$ (W_t is the weight) for totally T steps. Then, its uniform stability can be bounded by $\epsilon_{stab} \leq \frac{2L^2}{N}\sum_{t=1}^{T}(1 - \frac{\alpha}{2})\eta_t$.*

Proof. We have $\epsilon_{stab} \leq \frac{2L^2}{N}\sum_{t=1}^{T} W_t\eta_t$ from Theorem 4.1. Since $W_t \leq 1 - \frac{\alpha}{2}$, we have $\epsilon_{stab} \leq \frac{2L^2}{N}\sum_{t=1}^{T}(1 - \frac{\alpha}{2})\eta_t$, which completes the proof.

η_t is the original step size (for the t-th step) in SGD without weighting, so that we can conclude that NORMA has sharper uniform stability bound because $\frac{2L^2}{N}\sum_{t=1}^{T}(1 - \frac{\alpha}{2})\eta_t < \frac{2L^2}{N}\sum_{t=1}^{T}\eta_t$, i.e., NORMA has lower generalization error bound than standard matrix approximation method.

4.3 Expected Risk Bound Generalization error is only part of the true risk, because generalization error will sometimes be traded with optimization error [20]. Therefore, we analyze the expected risk bound of NORMA here, which is obtained by considering the true data distribution and thus can be regarded as true risk [10, 20]. The expected risk of a learned model w by stochastic gradient descent (SGD) on a sample S can be bounded as follows [10, 20]:

$$\mathbb{E}(D(w)) \leq \mathbb{E}(D_S(w_*^S)) + \epsilon_{opt} + \epsilon_{stab}, \tag{4.11}$$

where $D(w)$ is the mean square error, w_*^S is the model with minimum mean square error, ϵ_{opt} is the optimization error, and ϵ_{stab} is the uniform stability bound. Since $\mathbb{E}(D_S(w_*^S))$ can be regarded as a constant, we can bound the expected risk of w by $\epsilon_{opt} + \epsilon_{stab}$.

The following result from Nemirovski et al. [24] is adopted to analyze the expected risk bound, i.e., the bound of $\epsilon_{opt} + \epsilon_{stab}$.

THEOREM 4.3. *[24] Assume we run SGD with a constant step size η on a convex function $R(w) = \mathbb{E}_{x\in X} f(w;x)$, in which $||\nabla f(w;x)|| \leq L$ and $||w_0 - w_*|| \leq D$ ($w_* = \arg\min_w R(w)$). Let $\bar{w}_T$ be the average of T iterations by SGD, then $R(\bar{w}_T) \leq R(w_*) + \frac{D^2}{2\eta T} + \frac{L^2\eta}{2}$.*

The expected risk of solving classic least square problem using standard SGD can be bounded by the following theorem.

THEOREM 4.4. *[10] Let $S = \{x_1, \ldots, x_n\}$ ($|S| = N$). Suppose that we run SGD with totally T steps with constant step size $\gamma \leq 2/\beta$ from a start point w_0, and w_0 satisfies that $||w_0 - w_*|| \leq Q$. Then the average of the T iterations $\bar{w}_T$ satisfies that $\mathbb{E}(D(\bar{w}_T)) \leq \mathbb{E}(D_S(w_*^S)) + \frac{QL}{\sqrt{N}}\sqrt{\frac{N+2T}{T}}$.*

The following theorem shows that NORMA will not increase the expected risk bound of SGD (as shown in the above theorem) by properly choosing α in the weight definition (Equation 3.5). The proof of the theorem below is trivial and thus omitted.

THEOREM 4.5. *Let $S = \{x_1, \ldots, x_n\}$ ($|S| = N$). Suppose that we run SGD with totally T steps with constant step size $\eta \leq 2/\beta$ by properly choosing α in Equation 3.5 from a start point w_0, and w_0 satisfies that $||w_0 - w_*|| \leq Q$. Then the average of the T iterations $\bar{w}_T$ satisfies that $\mathbb{E}(D(\bar{w}_T)) \leq \mathbb{E}(D_S(w_*^S)) + \frac{QL}{\sqrt{N}}\sqrt{\frac{N+2T}{T}}$.*

In addition, the following theorem shows that NORMA can even achieve sharper expected risk bound of SGD by properly choosing α in the weight definition (Equation 3.5), i.e., NORMA can have lower true risk bound if we properly choose α.

THEOREM 4.6. *Let $S = \{x_1, ..., x_n\}$ ($|S| = n$). Loss function f is convex, β-smooth and L-Lipschitz. $R_S(w) = \frac{1}{n}\sum_{x \in S} f(w; x)$ and $w_*^S = \arg\min_w R_S(w)$. Suppose that we run SGD with T steps by a suitable constant step size $(1-s)\eta \leq 2/\beta$ ($0 < s \leq 1 - \frac{n}{2T}$) by properly choosing α in Equation 3.5 and from a start point w_0 satisfying that $||w_0 - w_*|| \leq D$. Then, the expected risk bound of the average of the T iterations $\bar{w}_T$ is sharper than that of Theorem 4.5.*

Proof. From Theorem 4.3, the optimization error bound can be obtained as follows:

$$\epsilon_{opt}(\bar{w}_T) \leq \frac{D^2}{2\eta(1-s)T} + \frac{L^2\eta}{2}.$$

The bound for uniform stability can be obtained by combining Lemma 4.4 and 4.7 in [10] as follows:

$$\epsilon_{stab} \leq \frac{\eta(1-s)^2L^2T}{n}.$$

Combining the two inequalities above, we have

$$\mathbb{E}(R(\bar{w}_T)) \leq \mathbb{E}(R_S(w_*^S)) + \frac{D^2}{2\eta(1-s)T} + \frac{L^2\eta}{2} + \frac{\eta(1-s)^2L^2T}{n}.$$

By choosing $\eta = \frac{D\sqrt{n}}{L\sqrt{(1-s)T(n+2(1-s)^2T)}}$, we can yield the the following expected risk bound:

$$\mathbb{E}(R(\bar{w}_T)) \leq \mathbb{E}(R_S(w_*^S)) + \frac{DL}{\sqrt{n}}\sqrt{\frac{n+2(1-s)^2T}{(1-s)T}}.$$

When $0 < s \leq 1 - \frac{n}{2T}$, we have that $\frac{DL}{\sqrt{n}}\sqrt{\frac{n+2(1-s)^2T}{(1-s)T}} \leq \frac{QL}{\sqrt{N}}\sqrt{\frac{N+2T}{T}}$, which completes the proof.

5 Experiments

In this section, we first present the experimental setup including dataset description, hyperparameter setting and details of the compared methods. Then, we empirically analyze the generalization error of NORMA. After that, we analyze the sensitivity of NORMA with different hyperparameters and the sensitivity of NORMA against manually inserted rating noises. At last, we compare the accuracy of NORMA with state-of-the-art matrix approximation-based collaborative filtering methods.

5.1 Experimental Setup *Dataset Description*. We evaluate the proposed method using three popular real-world datasets: 1) MovieLens 1M ($\sim 10^6$ ratings from 6k users on 4k movies); 2) MovieLens 10M ($\sim 10^7$ ratings from 70k users on 10k movies); and 3) Netflix ($\sim 10^8$ ratings from 480k users on 18k movies). For each experiment, we randomly split the dataset with 90% of data as training set and 10% of data as test set. All the results are reported by averaging over five different random splits.

Hyperparameter Setting. In the experiments, we set the learning rate $\lambda = 0.001$ and regularization coefficient $\mu = 0.02$. The convergence threshold is set to 10^{-5} and the maximum number of iterations is set to 600. The values of α and c' in Equation 3.5 are chosen based on the sensitivity analysis experiments. Note that the sensitivity analysis are conducted on training sets only, so that all comparisons with other methods are fair. The optimal parameters for all compared methods are chosen from their original papers.

Compared Methods. We compare the accuracy of NORMA with the following six state-of-the-art matrix approximation-based collaborative filtering methods: 1) BPMF [28] extends the PMF method [29] by a Baysian treatment to automatically control the model capacity; 2) LLORMA [18] combines a set of local low-rank matrix approximation sub-models using kernel smoothing to improve accuracy; 3) GSMF [31] introduces group-sparsity regularization to matrix approximation, which can model multiple user behaviors to improve accuracy; 4) WEMAREC [7] constructs sub-models by co-clustering-based matrix approximation, and then combines the sub-models by weighted average; 5) SMA [21] is a stable matrix approximation method, which can improve the generalization performance of MA-based CF methods; 6) ERMMA [20] can minimize the expected risk of matrix approximation, which can also improve the accuracy of collaborative filtering. Similar to NORMA, SMA and ERMMA also adaptively/randomly manipulate the gradients of a subset of examples in SGD, but both methods does not consider noise levels of different examples when manipulating the gradients.

We also compare NORMA with two popular robust matrix approximation-based CF methods, which were also proposed to address the noisy rating issue in matrix approximation: 1) Robust MF [23] proposes an iteratively reweighted matrix approximation method to improve recommendation stability and 2) RBMF [17] proposes to use heteroscedastic noise models in the PMF method [29] to improve predictive performance.

Since changing the weights of examples can be equivalently regarded as changing learning rates, we also compare NORMA with three popular adaptive learning rate methods as follows: 1) AdaGrad [9] uses larger learning rates for infrequent parameters and smaller learning rates for frequent parameters; 2) AdaDelta [32] uses a running average of the

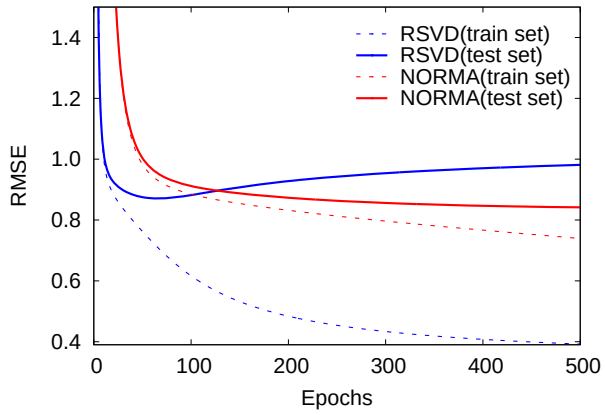

Figure 1: Training and test errors vs. epoch for NORMA and RSVD [27] on the MovieLens 1M dataset. Here, we set the rank $r = 100$, $\alpha = 0.6$ and $c' = 0.5$ for NORMA.

magnitudes of gradients to adjust the learning rates; and 3) Adam [13] considers the first and the second moment of the gradients to adjust the learning rates. Here, the three adaptive learning rate methods are applied in the RSVD [27] method as the baselines.

5.2 Generalization Error Analysis Figure 1 shows the trends of training and test errors with different number of epochs in NORMA and RSVD [27] on the MovieLens 1M dataset. The only difference between NORMA and RSVD is that NORMA adopts the adaptive weighting strategy proposed in this paper but RSVD weighs all ratings equally. We can see from the results that the training and test RMSEs of NORMA are much closer than those of RSVD. Meanwhile, it is noticeable that NORMA does not show any signs of overfitting even with 500 epochs. This indicates that NORMA achieves lower generalization error than RSVD, i.e., NORMA is less prone to overfitting than RSVD. Meanwhile, the lowest test RMSE of NORMA is around 0.8310, which is much smaller than that of RSVD — 0.8461. This means that NORMA achieves lower test error than RSVD, which is because less biased matrix approximation models has stronger predictive ability.

5.3 Sensitivity Analysis Here, we analyze the sensitivity of NORMA with different hyperparameters and noises. We compare NORMA with AdaGrad [9], AdaDelta [32] and Adam [13] in the experiments, and the results show that NORMA can outperform the three popular adaptive learning rate methods in recommendation accuracy.

5.3.1 Sensitivity with α Figure 2 shows how the RMSE of NORMA changes with α increasing from 0.1 to 0.9. We can see from the results that the test RMSE of NORMA first decreases with increasing α when $\alpha < 0.7$ and then increases afterwards. Since the variance term plays more important role in the weighting function as α increases, which indicates that introducing the variance in the adaptive weighting can be beneficial to the model learning. However, when α is too large, e.g., 0.8 or 0.9, the learning rate will become too small, so that the learning process will stop before reaching local minimum when the gain is too small. Since $\alpha = 0.6$ can achieve near optimal accuracy, we choose $\alpha = 0.6$ in the following experiments.

5.3.2 Sensitivity with c' Figure 2 shows how the RMSE of NORMA changes with c' increasing from 0.1 to 1. We can see from the results that the test RMSE of NORMA first decreases as c' increases when $c' < 0.5$ and then slightly increases afterwards. c' actually controls the confidence interval of the estimated variance, so that too small confidence interval (small c') or too large confidence interval (large c') will cause bad variance estimation. From this experiment, we know that $c' = 0.4$ achieves near optimal accuracy.

5.3.3 Sensitivity with Manual Noises Since NORMA can give noisy ratings lower weights in training, it is desirable that NORMA can be more robust with rating noises than standard matrix approximation methods. In this experiment, for each noise scale x, we first randomly choose 20% of ratings in the training data, and increase the ratings for half of the chosen data by x and decrease the ratings by x for the other half of the chosen data.

Figure 4 analyzes how NORMA performs with different level of manually inserted rating noises. As we can see from the results, the test RMSE variation of NORMA is smaller than 0.009 when the noise scale increases from 0.1 to 1, but the test RMSE variations of the other three adaptive learning rate methods are larger than 0.011 with the same setting. This experiment indicates that NORMA is more stable than RSVD with adaptive learning rate methods on noisy ratings, which confirms that NORMA can indeed achieve robust recommendation on noisy ratings.

5.4 Accuracy Comparison Table 1 compares the recommendation accuracy of NORMA with six state-of-the-art matrix approximation-based CF methods. Among the compared methods, LLORMA [18] and WEMAREC [7] are ensemble methods, which are empirically more accurate than standalone methods. However, NORMA statistically significantly outperforms all the compared methods on both datasets with at least 95% confidence level. The main reasons are: 1) NORMA can learn less biased models than the other methods due to giving lower weights to more noisy ratings, i.e., NORMA can achieve better generalization performance and 2) the adaptive weighting strategy can adjust learning steps in SGD to achieve better convergence, which is similar to the adaptive learning rate methods in SDG [9, 13, 32].

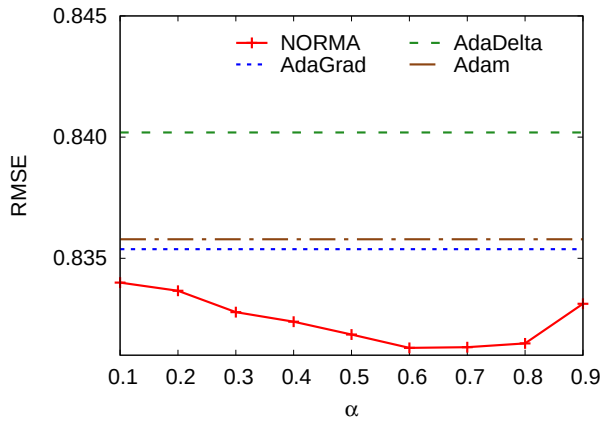

Figure 2: Sensitivity analysis of NORMA with different α values on the MovieLens 1M dataset. We set the rank $r = 100$ and $c' = 0.5$ in the experiment.

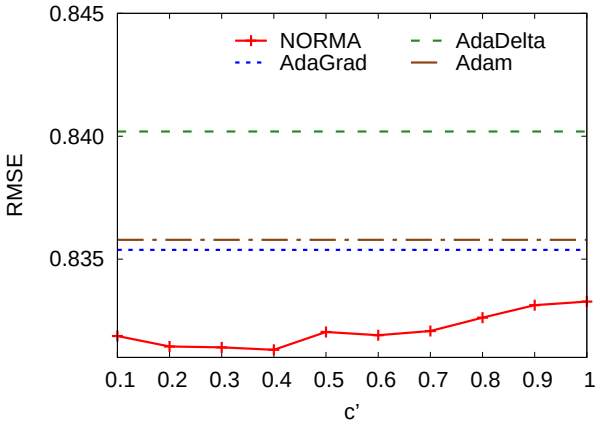

Figure 3: Sensitivity analysis of NORMA with different c' values on the MovieLens 1M dataset. We set the rank $r = 100$ and $\alpha = 0.6$ in the experiment.

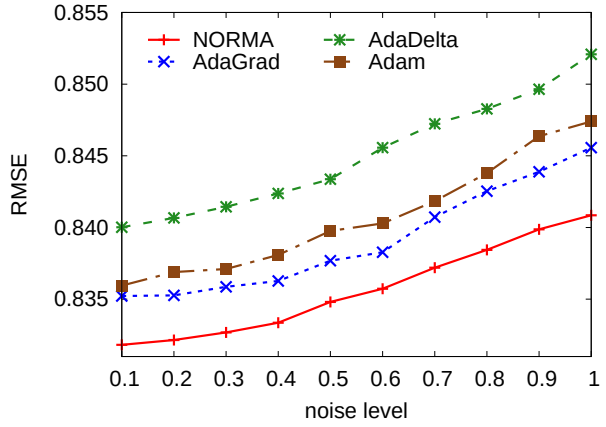

Figure 4: Sensitivity analysis of NORMA with different manual noises on the MovieLens 1M dataset. We set the rank $r = 100$, $\alpha = 0.6$ and $c' = 0.5$ in the experiment.

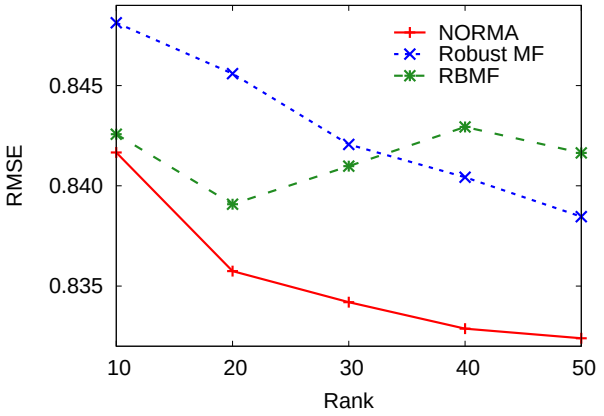

Figure 5: Test RMSE comparison between NORMA and two robust matrix approximation methods: Robust MF [23] and RBMF [17] on the MovieLens 1M dataset.

5.5 Comparison with Robust MF Methods Robust collaborative filtering methods have been proposed to achieve robust recommendation against noisy or even malicious user ratings. Here, we compare the performance of NORMA with two of the state-of-the-art robust matrix approximation-based CF methods — robust matrix factorization (Robust MF) method [23] and robust Bayesian matrix factorization (RBMF) method [17]. Figure 5 compares the test RMSE of NORMA with the two robust matrix approximation methods on MovieLens 1M dataset with rank increasing from 10 to 50. As we can see from the results, NORMA can achieve much lower test RMSEs than both methods with all rank values. Since NORMA and the other two robust methods only differ in how to model the noises in learning, the superior performance of NORMA indicates that the proposed adaptive weighting strategy is more desirable in addressing the noisy rating issue in collaborative filtering applications.

6 Related Work

Collaborative filtering is the most important class of methods in many real-world recommender systems [1]. Among existing collaborative filtering methods, matrix approximation-based methods have achieved state-of-the-art accuracy in rating prediction tasks recently. Billsus et al. [4] first applied singular value decomposition on user-item rating matrix for collaborative filtering. Paterek [27] proposed an improved method for regularized SVD method and achieved significantly better accuracy than the baseline method of Net-

Table 1: Root mean square error (RMSE) comparison between NORMA (rank = 300) and six state-of-the-art matrix approximation-based collaborative filtering methods — BPMF [28], LLORMA [18], GSMF [31], WEMAREC [7], SMA [21], ERMMA [20] on the MovieLens 10M and Netflix datasets. Note that NORMA statistically significantly outperforms the other methods with 95% confidence level on both datasets.

Method	MovieLens (10M)	Netflix
BPMF	0.8197 ± 0.0004	0.8421 ± 0.0002
GSMF	0.8012 ± 0.0011	0.8420 ± 0.0006
LLORMA	0.7855 ± 0.0002	0.8275 ± 0.0004
WEMAREC	0.7775 ± 0.0007	0.8143 ± 0.0001
SMA	0.7682 ± 0.0003	0.8036 ± 0.0004
ERMMA	0.7670 ± 0.0007	0.8018 ± 0.0001
NORMA	$\mathbf{0.7641 \pm 0.0008}$	$\mathbf{0.7986 \pm 0.0002}$

flix prize challenge — the Cinematch method. Salakhutdinov and Minh [29] first proposed the probabilistic matrix factorization (PMF) method, which achieves matrix approximations from a probabilistic view. Then, they improved the PMF method by a Baysian treatment and proposed the Bayesian probabilistic matrix factorization (BPMF) method [28], which can automatically control the model capacity. Koren [15] combined the implicit feedback into the matrix approximation methods and proposed the SVD++ method. Li et al. [21] proposed a stable matrix approximation method, which can improve the generalization performance of matrix approximation-based collaborative filtering. Later, they extended the stable matrix approximation method and proposed ERMMA [20], which can achieve lower expected risk in learning matrix approximation models using SGD by randomly shrinking the gradients of a subset of training examples. However, these methods do not consider the noises in user-item ratings and treat all ratings equally important in model training, which will make the learned MA models be biased when facing with noisy ratings.

The noisy rating issue in collaborative filtering has also been studied recently. Hill et al. [11] observed a correlation of 83% when users are asked to re-rate the movies that they rated 6 weeks ago. The study of Cosley et al. [8] showed that around 60% of user ratings are kept the same when users are asked to re-rate movies that they rated before. Amatriain et al. [2] also studied users' consistency when rating movies and found that users are inconsistent even when determining whether they have seen a movie or not. Their findings reveal that user ratings are indeed noisy and unreliable, so that matrix approximation models will be biased if ratings with different levels of noises are treated equally.

Several methods have been recently proposed to address the noisy rating issue in collaborative filtering. Amatriain et al. [3] proposed to let users re-rate items that they rated before, which can help to remove the natural noises. They observed around 10% improvements in terms of RMSE after denoising. However, their method is not scalable in real applications, because users pay a lot of effort when rating items [30]. Another type of methods tried to design robust collaborative filtering methods to address the rating noise issue [23, 17]. Mehta et al. [23] found that M-estimators cannot significantly improve the stability of collaborative filtering, and they proposed an iteratively reweighted matrix approximation method to improve recommendation stability on noisy ratings. Lakshminarayanan et al. [17] studied the noisy model in the PMF method, and compared different noise models and prior distributions in RBMF. These robust collaborative filtering methods can improve stability of recommendation, but cannot achieve similarly significant improvement in recommendation accuracy compared to the proposed method as shown in the experiments.

Another type of related work is the adaptive learning rate method in SGD, e.g., AdaGrad [9], AdaDelta [32] and Adam [13], etc., because the proposed adaptive weighting strategy can be regarded as a variant of adaptive learning rate method. The merit of these methods is that frequent parameters are given smaller updates and infrequent parameters are given larger updates to improve model convergence. However, these methods are not perfect for matrix approximation-based collaborative filtering on noisy ratings, because both frequent and infrequent parameters can appear on noisy ratings. In the contrary, NORMA can give larger gradient updates on less noisy ratings and give smaller gradient updates to noisy ratings, so that the learned MA models will be less biased to noisy ratings.

7 Conclusion and Future Work

Collaborative filtering is important in today's recommender systems, but the unavoidable noises in the user-item ratings raise challenges to matrix approximation-based CF methods. This paper views the matrix approximation on noisy ratings as a weighted matrix approximation method. An adaptive weighting strategy is proposed to decrease learning steps on noisy ratings, so that the learned MA models will be less sensitive to noises. Theoretical and empirical analyses show that the proposed method can achieve better generalization performance than existing method. Empirical studies on real-world datasets demonstrate that the proposed method can outperform six state-of-the-art matrix approximation-based collaborative filtering methods in recommendation accuracy. One possible extension of this work is to apply the idea of NORMA to other adaptive learning rate methods, e.g., AdaGrad [9], to further improve performance.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 61332008 and U1630115.

References

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[2] X. Amatriain, J. M. Pujol, and N. Oliver. I like it... i like it not: Evaluating user ratings noise in recommender systems. In *The 17th International Conference on User Modeling, Adaptation, and Personalization*, pages 247–258, 2009.

[3] X. Amatriain, J. M. Pujol, N. Tintarev, and N. Oliver. Rate it again: Increasing recommendation accuracy by user re-rating. In *Proceedings of the Third ACM Conference on Recommender Systems*, pages 173–180. ACM, 2009.

[4] D. Billsus and M. J. Pazzani. Learning collaborative information filters. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 46–54, 1998.

[5] O. Bousquet and A. Elisseeff. Algorithmic stability and generalization performance. In *Advances in Neural Information Processing Systems*, pages 196–202, 2001.

[6] C. Chen, D. Li, Q. Lv, J. Yan, L. Shang, and S. Chu. GLOMA: embedding global information in local matrix approximation models for collaborative filtering. In *The 31st AAAI Conference on Artificial Intelligence*, pages 1295–1301, 2017.

[7] C. Chen, D. Li, Y. Zhao, Q. Lv, and L. Shang. WEMAREC: Accurate and scalable recommendation through weighted and ensemble matrix approximation. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 303–312, 2015.

[8] D. Cosley, S. K. Lam, I. Albert, J. A. Konstan, and J. Riedl. Is seeing believing?: How recommender system interfaces affect users' opinions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 585–592, 2003.

[9] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[10] M. Hardt, B. Recht, and Y. Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *Proceedings of the 33rd International Conference on Machine Learning*, ICML'16, pages 1225–1234, 2016.

[11] W. Hill, L. Stead, M. Rosenstein, and G. Furnas. Recommending and evaluating choices in a virtual community of use. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 194–201. ACM, 1995.

[12] N. Jones, A. Brun, and A. Boyer. Comparisons instead of ratings: Towards more stable preferences. In *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, WI-IAT '11, pages 451–456. IEEE, 2011.

[13] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] D. Kluver, T. T. Nguyen, M. Ekstrand, S. Sen, and J. Riedl. How many bits per rating? In *Proceedings of the Sixth ACM Conference on Recommender Systems*, pages 99–106, 2012.

[15] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434. ACM, 2008.

[16] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[17] B. Lakshminarayanan, G. Bouchard, and C. Archambeau. Robust Bayesian matrix factorisation. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pages 425–433, 2011.

[18] J. Lee, S. Kim, G. Lebanon, and Y. Singer. Local low-rank matrix approximation. In *The 30th International Conference on Machine Learning*, pages 82–90, 2013.

[19] D. Li, C. Chen, W. Liu, T. Lu, N. Gu, and S. Chu. Mixture-rank matrix approximation for collaborative filtering. In *Advances in Neural Information Processing Systems*, pages 477–485, 2017.

[20] D. Li, C. Chen, Q. Lv, L. Shang, S. Chu, and H. Zha. ER-MMA: expected risk minimization for matrix approximation-based recommender systems. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pages 1403–1409, 2017.

[21] D. Li, C. Chen, Q. Lv, J. Yan, L. Shang, and S. M. Chu. Low-rank matrix approximation with stability. In *Proceedings of the 33rd International Conference on Machine Learning (ICML '16)*, pages 295–303, 2016.

[22] D. Li, Q. Lv, X. Xie, L. Shang, H. Xia, T. Lu, and N. Gu. Interest-based real-time content recommendation in online social communities. *Knowledge-Based Systems*, 28:1–12, 2012.

[23] B. Mehta, T. Hofmann, and W. Nejdl. Robust collaborative filtering. In *Proceedings of the 2007 ACM Conference on Recommender Systems*, pages 49–56. ACM, 2007.

[24] A. Nemirovsky and D. Yudin. *Problem complexity and method efficiency in optimization*. John Wiley, 1983.

[25] T. Nguyen, D. Kluver, T. Wang, P. Hui, M. Ekstrand, M. Willemsen, and J. Riedl. Rating support interfaces to improve user experience and recommender accuracy. In *ACM Conference on Recommender Systems*, pages 149–156, 2013.

[26] M. P. O'Mahony, N. J. Hurley, and G. C. Silvestre. Detecting noise in recommender system databases. In *Proceedings of the 11th International Conference on Intelligent User Interfaces*, IUI '06, pages 109–115. ACM, 2006.

[27] A. Paterek. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD cup and workshop*, volume 2007, pages 5–8, 2007.

[28] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th international conference on Machine learning (ICML '08)*, pages 880–887. ACM, 2008.

[29] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2008.

[30] E. I. Sparling and S. Sen. Rating: How difficult is it? In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 149–156. ACM, 2011.

[31] T. Yuan, J. Cheng, X. Zhang, S. Qiu, and H. Lu. Recommendation by mining multiple user behaviors with group sparsity. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pages 222–228, 2014.

[32] M. D. Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.