Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

ERMMA: Expected Risk Minimization for Matrix Approximation-based Recommender Systems

Dongsheng Li[1], **Chao Chen,**[1] **Qin Lv,**[2] **Li Shang,**[2] **Stephen M. Chu,**[1] **Hongyuan Zha**[3]
[1]IBM Research – China, Shanghai, P.R. China, 201203
[2]Univeristy of Colorado Boulder, Boulder, Colorado, USA, 80309
[3]Georgia Institute of Technology, Atlanta, Georgia, USA, 30332
{ldsli, schu}@cn.ibm.com, chench.resch@gmail.com, {qin.lv, li.shang}@colorado.edu, zha@cc.gatech.edu

Abstract

Matrix approximation (MA) is one of the most popular techniques in today's recommender systems. In most MA-based recommender systems, the problem of risk minimization should be defined, and how to achieve minimum expected risk in model learning is one of the most critical problems to recommendation accuracy. This paper addresses the expected risk minimization problem, in which expected risk can be bounded by the sum of optimization error and generalization error. Based on the uniform stability theory, we propose an expected risk minimized matrix approximation method (ERMMA), which is designed to achieve better tradeoff between optimization error and generalization error in order to reduce the expected risk of the learned MA models. Theoretical analysis shows that ERMMA can achieve lower expected risk bound than existing MA methods. Experimental results on the MovieLens and Netflix datasets demonstrate that ERMMA outperforms six state-of-the-art MA-based recommendation methods in both rating prediction problem and item ranking problem.

Introduction

Matrix approximation (MA) is popular among existing techniques for recommender systems mainly due to its high accuracy (Koren et al. 2009; Ekstrand, Riedl, and Konstan 2011). In recommender systems, the observed data in user-item rating matrices are incomplete, and closed-form solutions cannot be easily obtained as in fully observed matrices (Candès and Plan 2010). Therefore, in existing MA methods (Paterek 2007; Koren 2008; Salakhutdinov and Mnih 2008; Yan et al. 2010; Li et al. 2010; Lee et al. 2013), risk functions, typically empirical risks, should be first defined to measure the goodness of MA models on training data. Then, iterative learning methods, e.g., gradient descent-based methods, are adopted to learn MA models with local/global minimum empirical risks. The learned models can then be adopted to predict user interests on unrated items.

A common issue of existing MA methods using gradient descent-based learning is that the learned models may perform well on observed examples but achieve worse results on unobserved examples, i.e., low generalization performance (Ng 2004; Hardt, Recht, and Singer 2015; Li et al. 2016). Recently, Li et al. (2016) proposed a stable matrix approximation method to improve the generalization performance of MA methods. However, low generalization error cannot always guarantee good model performance, because the expected risk, which measures the true risk of a model on both observed and unobserved examples, does not always decrease when generalization error decreases. As pointed out by Hardt et al. (2015), expected risk can be decomposed into three terms: (1) minimum empirical risk, (2) optimization error (the discrepancy between empirical risk and minimum empirical risk), and (3) generalization error (the discrepancy between empirical risk and expected risk). Minimum empirical risk is fixed when the type of model is fixed. Therefore, the goal of expected risk minimization is to reduce the sum of optimization error and generalization error. Generally, we cannot simultaneously reduce optimization error and generalization error. As such, the goal of expected risk minimization is to achieve a good tradeoff between optimization error and generalization error. Although the critical issue of the tradeoff between optimization error and generalization error has been mentioned previously, no clear solution has been articulated in MA methods. To the best of our knowledge, this work represents a first attempt to address this problem and we hope it will draw the attention of researchers to further investigate this key problem in MA-based recommender systems.

This paper proposes an expected risk minimized matrix approximation method (ERMMA), which can achieve a good tradeoff between optimization error and generalization error both theoretically and empirically. ERMMA randomly shrinks the learning step of a fraction of training examples in each epoch during stochastic gradient updates, which can improve generalization performance over standard methods due to lower bound of uniform stability. Our theoretical analysis proves that ERMMA can achieve lower expected risk bound, i.e., a better tradeoff between optimization error and generalization error, compared with classic MA methods. Moreover, based on our theoretical analysis, popular techniques such as regularization and adaptive learning steps, which have been successful in improving model performance, can also be incorporated by ERMMA without violating its key characteristics. Experimental studies using real-world datasets (MovieLens and Netflix) demonstrate that, by introducing proper regularization term and adaptive learning steps, ERMMA can achieve better accuracy compared with

six state-of-the-art MA-based collaborative filtering methods in both rating prediction and item ranking problems, further pushing the state-of-the-art performance for this important class of problems.

Problem Formulation

This section first defines the key concepts of expected risk minimization and then introduces low-rank matrix approximation.

Empirical Risk vs. Expected Risk

Consider a sample $S = \{x_1, x_2, \ldots, x_n\}$ ($x_i \in X$) drawn i.i.d. from some unknown distribution $\mathcal{D}$. The general goal of empirical risk minimization is to find a model w that can minimize the following empirical risk:

$$R_S(w) = \frac{1}{n} \sum_{x_i \in S} f(w; x_i) \tag{1}$$

where f is a given loss function and $f(w; x)$ measures the loss of model w over example x. Similarly, the true risk of a given model w can be measured as follows:

$$R(w) = \mathbb{E}_{x \sim \mathcal{D}} f(w; x) \tag{2}$$

Then, given a randomized learning algorithm A and $w = A(S)$, the general goal of the learning algorithm A is to find a w that can minimize the discrepancy between $R(w)$ and $R_S(w)$. Here, we adopt the expected generalization error (Hardt, Recht, and Singer 2015) to measure this discrepancy as follows:

$$\epsilon_{gen} = \mathbb{E}_{A,S}(R_S(A(S)) - R(A(S))) \tag{3}$$

where the expectation is taken over the randomness of A and S. To bound ϵ_{gen}, uniform stability on randomized learning algorithms can be employed.

Definition 1. *[Uniform Stability (Bousquet and Elisseeff 2001)] A randomized learning algorithm A is ϵ-uniformly stable if for any two samples S and S' satisfying that S and S' differ in at most one example, we have*

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

Let ϵ_{stab} be the smallest upper bound in the definition above, Hardt et al. (2015) pointed out that the expected risk of a learned model w by stochastic gradient descent (SGD) on a sample S can be bounded as follows:

$$\mathbb{E}(R(w)) \leq \mathbb{E}(R_S(w_*^S)) + \epsilon_{opt} + \epsilon_{stab} \tag{4}$$

where $w_*^S = \arg\min_w R_S(w)$ is the model with minimum empirical risk, ϵ_{opt} is the expected gap between empirical risk and minimum empirical risk, and ϵ_{stab} is the uniform stability bound. Generally, ϵ_{opt} will decrease with the number of iterations in SGD while ϵ_{stab} will increase. Therefore, a tradeoff between ϵ_{opt} and ϵ_{stab} is needed to achieve lower $\epsilon_{opt} + \epsilon_{stab}$, i.e., lower expected risk bound.

Low-Rank Matrix Approximation

Given the general framework above, we now focus on the specific problem of matrix approximation. Consider a user-item rating matrix $M \in \mathbb{R}^{m \times n}$, where m and n stand for the numbers of users and items, respectively. The general goal of matrix approximation is to determine two feature matrices $U \in \mathbb{R}^{m \times r}$ and $V \in \mathbb{R}^{n \times r}$, such that $M \approx \hat{M} = UV^T$. In some cases, good performance can be achieved when the rank of U and V $r \ll \min\{m, n\}$. As such, this type of matrix approximation is also referred to as low-rank matrix approximation (LRMA). To determine appropriate U and V, the problem of empirical risk minimization is usually defined and solved as follows (Paterek 2007; Koren 2008; Salakhutdinov and Mnih 2008):

$$U, V = \arg\min_{U', V'} R_S(U', V') \tag{5}$$

where S is the set of observed user-item ratings in M.

Typically, mean square loss is adopted in the above problem, i.e., $f(U, V; M_{i,j}) = (M_{i,j} - U_i V_j^T)^2$. However, the low-rank assumption makes the square loss non-convex. Recently, a convex relaxation of LRMA is proposed by Mazumder et al. (2010):

$$f(Z; M_{i,j}) = (M_{i,j} - Z_{i,j})^2 + \mu ||Z||_* \tag{6}$$

where nuclear norm $||Z||_*$ is the sum of the singular values of Z (a convex relaxation of the rank). Recent theoretical analysis (Candès and Recht 2009; Mazumder, Hastie, and Tibshirani 2010) showed that the above f is convex in Z.

Solving LRMA Problems

Many recent works (Gemulla et al. 2011; Chen et al. 2015; Li et al. 2016) adopted stochastic gradient descent (SGD) to solve the LRMA problem due to its low computation cost (Mazumder, Hastie, and Tibshirani 2010) and high generalization performance (Hardt, Recht, and Singer 2015). In SGD, loss functions can be iteratively optimized as follows:

$$w \leftarrow w - \alpha \nabla f(w; x) \tag{7}$$

where w is a model and x is a randomly selected example. Hardt et al. (2015) proved that the SGD method can achieve good generalization performance. Moreover, the generalization performance of SGD can be improved by setting a fraction of the gradient weights to zero (Hardt, Recht, and Singer 2015):

$$w \leftarrow w - \alpha d_s \nabla f(w; x) \tag{8}$$

where d_s is 0 with probability s and 1 with probability $1 - s$. However, the method above cannot guarantee minimum expected risk because setting gradients to zero means those examples are not chosen in training and unchosen examples will sacrifice training accuracy, which can not be easily compensated by the improvement in generalization performance. This will be further analyzed in the next section.

ERMMA: Expected Risk Minimized Matrix Approximation

In this section, we first analyze the generalization performance improvement of ERMMA and then prove that ERMMA can also achieve better tradeoff between ϵ_{opt} and ϵ_{stab}, i.e., lower expected risk bound.

Optimization Problem of ERMMA

Instead of setting a fraction of the gradient weights to zero, ERMMA shrinks a fraction of the gradient weights by a parameter $0 \leq \lambda \leq 1$ in SGD, so that there is less sacrifice in optimization accuracy and generalization performance can still be improved. To do so, it is equivalent to optimizing the following problem using standard SGD:

$$\min_w \lambda R_S(w) + (1-\lambda) R_{S_d}(w) \tag{9}$$

where $R_S(w)$ is empirical risk (Equation 1) and $R_{S_d}(w) = \frac{1}{n_d}\sum_{x_i \in S_d} f(w; x_i)$ is the empirical risk on S_d. Here, S_d is the set of examples that are randomly selected from S with probability $1-s$, and n_d is the number of examples in S_d. Note that solving the problem above using SGD is equivalent to Equation 7 if we set $\lambda = 1$ and equivalent to Equation 8 if we set $\lambda = 0$.

Generalization Error Analysis

Hardt et al. (2015) have analyzed the generalization error bound of solving classic empirical risk minimization problem using standard SGD method, in which uniform stability can be bounded as follows:

Theorem 1. *Given a loss function $f : \Omega \to \mathbb{R}$, assuming $f(\cdot; x)$ is convex, $||\nabla f(\cdot; x)|| \leq L$ (L-Lipschitz) and $||\nabla f(w; x) - \nabla f(w'; x)|| \leq \beta ||w - w'||$ (β-smooth) for all $x \in X$ and $w, w' \in \Omega$. Suppose that SGD is run with the t-th step size $\alpha_t \leq 2/\beta$ for totally T steps. Then, SGD satisfies uniform stability on samples with n examples by $\epsilon_{stab} \leq \frac{2L^2}{n}\sum_{t=1}^{T}\alpha_t$.*

The following theorem proves that solving the problem defined in Equation 9 using SGD has lower uniform stability bound, i.e., better generalization performance.

Theorem 2. *Given a loss function $f : \Omega \to \mathbb{R}$, assuming that $f(\cdot; x)$ is convex, $||\nabla f(\cdot; x)|| \leq L$ (L-Lipschitz) and $||\nabla f(w; x) - \nabla f(w'; x)|| \leq \beta ||w - w'||$ (β-smooth) for all $x \in X$ and $w, w' \in \Omega$. Suppose that we run SGD to solve the problem defined in Equation 9 with the t-th step size $\alpha_t \leq 2/\beta$ for totally T steps, then this satisfies uniform stability on samples with n examples by $\epsilon_{stab} \leq \frac{2(1-s+s\lambda)^2 L^2}{n}\sum_{t=1}^{T}\alpha_t$.*

Proof. Proof can be found in supplementary material. □

Remark 1. Since s and λ are in $[0, 1]$, we know that $1 - s + s\lambda \leq 1$, i.e., $\frac{2(1-s+s\lambda)^2 L^2}{n}\sum_{t=1}^{T}\alpha_t \leq \frac{2L^2}{n}\sum_{t=1}^{T}\alpha_t$. Therefore, we can conclude that solving ERMMA using SGD is more stable than solving classic empirical risk minimization problems using SGD. In other words, the models learned from ERMMA generalize better when we adopt the same number of iterations T and learning step α_t for both methods.

Expected Risk Analysis

Here, we analyze the expected risk of solving the problem defined in Equation 9 using SGD and prove that it can yield lower expected risk bound, i.e., lower $\epsilon_{opt} + \epsilon_{stab}$, than solving classic empirical risk minimization problem using Equation 7 and Equation 8 in certain circumstances. The result from Nemirovski and Yudin (Nemirovsky and Yudin 1983) are adopted to analyze the bound of $\epsilon_{opt} + \epsilon_{stab}$.

Theorem 3. *(Nemirovsky and Yudin 1983) Assume we run SGD with constant step size α 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 $R(\bar{w}_T) \leq R(w_*) + \frac{D^2}{2\alpha T} + \frac{L^2\alpha}{2}$.*

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

Theorem 4. *(Hardt, Recht, and Singer 2015) Let $S = \{x_1, \ldots, x_n\}$ ($|S| = n$). 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 constant step size $\alpha \leq 2/\beta$ and from a start point w_0 satisfying that $||w_0 - w_*|| \leq D$. Then the average of the T iterations $\bar{w}_T$ satisfies that $\mathbb{E}(R(\bar{w}_T)) \leq \mathbb{E}(R_S(w_*^S)) + \frac{DL}{\sqrt{n}}\sqrt{\frac{n+2T}{T}}$.*

The expected risk of solving classic empirical risk minimization problem using Equation 8 can be bounded by the following theorem.

Theorem 5. *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 Equation 8 with T steps by a suitable constant step size $\alpha \leq 2/\beta$ and from a start point w_0 satisfying that $||w_0 - w_*|| \leq D$. Then the average of the T iterations $\bar{w}_T$ satisfies that $\mathbb{E}(R(\bar{w}_T)) \leq \mathbb{E}(R_S(w_*^S)) + \frac{DL}{\sqrt{n}}\sqrt{\frac{n+2(1-s)^2T}{(1-s)T}}$.*

Proof. Proof can be found in supplementary material. □

The expected risk of solving ERMMA using SGD can be bounded as follows.

Theorem 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 solve ERMMA using SGD (with rate s and shrinkage coefficient λ) with T steps by a suitable constant step size $\alpha \leq 2/\beta$ and from a start point w_0 satisfying that $||w_0 - w_*|| \leq D$. Then the average of the T iterations $\bar{w}_T$ satisfies that $\mathbb{E}(R(\bar{w}_T)) \leq \mathbb{E}(R_S(w_*^S)) + \frac{DL}{\sqrt{n}}\sqrt{\frac{\lambda n+2(1-s+\lambda s)^2T}{\lambda T}}$.*

Proof. Proof can be found in supplementary material. □

From Theorem 4 and 6, we know that the expected risk of ERMMA is more sharply bounded than that of classic empirical risk minimization problem if $\frac{DL}{\sqrt{n}}\sqrt{\frac{\lambda n+2(1-s+\lambda s)^2T}{\lambda T}} \leq \frac{DL}{\sqrt{n}}\sqrt{\frac{n+2T}{T}}$, i.e., $\frac{(1-s+\lambda s)^2}{\lambda} \leq 1$. However, it is nontrivial to directly compare the expected risk bounds between Theorem 5 and Theorem 6. Instead, we compare the lower bounds of these two theorems. Based on the arithmetic-mean inequality, we know that for Equation 8 $\frac{DL}{\sqrt{n}}\sqrt{\frac{n+2(1-s)^2T}{(1-s)T}} = \frac{DL}{\sqrt{n}}\sqrt{\frac{n/(1-s)+2(1-s)T}{T}} \geq \frac{DL}{\sqrt{n}}\sqrt{\frac{2\sqrt{2nT}}{T}}$. For ERMMA, we have $\frac{DL}{\sqrt{n}}\sqrt{\frac{\lambda n+2(1-s+\lambda s)^2T}{\lambda T}} = \frac{DL}{\sqrt{n}}\sqrt{\frac{n+2(1-s+\lambda s)^2T/\lambda}{T}} \geq$

$\frac{DL}{\sqrt{n}}\sqrt{\frac{2\sqrt{2nT(1-s+\lambda s)^2/\lambda}}{T}}$. Again, we can conclude that if $(1-s+\lambda s)^2/\lambda \leq 1$ then the lower bound of the expected risk bound of ERMMA will be smaller than that of solving empirical risk minimization problem using Equation 8.

Remark 2. ERMMA can achieve sharper bound of expected risk than solving classic empirical risk minimization problem using Equation 7 and Equation 8 if $\frac{(1-s+\lambda s)^2}{\lambda} \leq 1$. Solving the inequality above, we can conclude that if $\lambda \in [\frac{(1-s)^2}{s^2}, 1]$ and $s \in [\frac{1}{2}, 1]$ then ERMMA will have sharper expected risk bound than solving empirical risk minimization problem using Equation 7 and Equation 8.

Extension of ERMMA

This section extends ERMMA to more general and practical settings, so that popular "tricks" of SGD, e.g., regularization, adaptive learning step, etc., can also be adopted by ERMMA.

Regularization

Regularization is popular among today's machine learning techniques. Here, we analyze the error bounds of ERMMA with one of the most popular regularization method — L_2-regularization (Ng 2004). By introducing L_2-regularization term $\frac{\mu}{2}||w||^2$, the gradient update rule of ERMMA should be modified as follows:

$$g(w) = w - \alpha\mu w - \alpha(\lambda \nabla f(w;x) + (1-\lambda) d_s(x) \nabla f(w;x))$$

where $d_s(x) = 0$ if x is selected with probability s and $d_s(x) = 1$ otherwise. Then, we analyze how expansive the new update rule $g(w)$ is. For any w and w', we have

$$\begin{aligned}
&||g(w) - g(w')|| \\
\leq &(1-\alpha\mu)||w-w'|| + \alpha(\lambda + d_s(x)(1-\lambda))||\nabla f(w) - \nabla f(w')|| \\
\leq &(1-\alpha\mu)||w-w'|| + \alpha(\lambda + d_s(x)(1-\lambda))\beta||w-w'|| \\
\leq &(1+\alpha(\beta-\mu))||w-w'||
\end{aligned}$$

where the first inequality holds due to triangle inequality, the second inequality holds because f is β-smooth, and the third inequality holds because $d_s(x) \leq 1$. Thus, if we choose $\mu < \beta$ and $\alpha \leq 2/(\beta - \mu)$, the update rule $g(w)$ will still be 1-expansive, so that all the theorems in the paper can be proved by replacing β with $\beta - \mu$. This indicates that all the characteristics of ERMMA can hold when L_2-regularization is introduced.

Adaptive Learning Step

In many real-world problems, SGD will converge with a small number of epochs, e.g., tens of epochs. If we randomly choose S_d in Equation 9 for each epoch, some examples may be chosen many times while some other examples may not be chosen at all by the end of training. Therefore, underfitting or overfitting issues may appear on some examples. Learning rate adaption have been proposed to address such kind of issue (Duchi, Hazan, and Singer 2011; Zeiler 2012), which can adaptively set learning rates to improve robustness of SGD. This idea can be adopted in ERMMA by setting adaptive learning steps for different examples, e.g., the new learning step can be defined as $\alpha'_t = l(t,x)\alpha_t$, where $l(t,x)$ is the adaptive weight for example x at step t. Next, we analyze the stability of ERMMA with such adaptive learning steps.

Lemma 1. *Assuming that loss function f is convex and β-smooth, if the t-th learning step of solving ERMMA using SGD is $\alpha'_t = l(t,x)\alpha_t$, then the gradient update of ERMMA will be 1-expansive if $l(t,x)\alpha_t \leq 2/\beta$.*

Proof. Proof can be found in supplementary material. □

Then, all the theorems in the paper can be similarly proved by replacing α_t with $\alpha'_t = l(t,x)\alpha_t$. By introducing L_2-regularization and adaptive learning step to ERMMA, we can extend Equation 9 as follows:

$$\min_w \frac{\lambda}{n} \sum_{x_i \in S} l_1(t,x) f(w;x_i) + \frac{\lambda'}{n_d} \sum_{x_i \in S_d} l_2(t,x) f(w;x_i) + \frac{\mu}{2}||w||^2 \tag{10}$$

where $l(t,x)$ is the adaptive weight to adjust the learning rate for example x at the t-th step and $\lambda' = 1 - \lambda$. n and n_d are the numbers of examples in S and S_d, respectively. Note that, l_1 and l_2 in Equation 10 can be different weighting functions as long as Lemma 1 holds.

Experiments

In this section, we verify the performance of ERMMA in two popular scenarios of recommender systems: 1) rating prediction, in which ERMMA predicts how users will rate unseen items; and 2) item ranking, in which ERMMA predicts how users will rank different unseen items.

Based on previous analysis, it is desirable to choose the loss function as follows: $f(U,V;M_{i,j}) = (M_{i,j} - Z_{i,j})^2 + \mu_1||Z||_* + \mu_2||U||^2 + \mu_3||V||^2$, where U and V are user and item feature matrices, resp., M is the targeted user-item rating matrix, and $Z = UV^T$ is the approximation of M by U and V. However, it is non-trivial to minimize nuclear norm using iterative methods. Therefore, we try to minimize an upper bound of the above loss function based on the property that $||Z||_* \leq \min_{U,V} \frac{1}{2}(||U||^2 + ||V||^2)$ (Srebro and Shraibman 2005) as follows:

$$f(U,V;M_{i,j}) = (M_{i,j} - Z_{i,j})^2 + \mu(||U||^2 + ||V||^2).$$

Moreover, we define $l_1(t,x_i)$ and $l_2(t,x_i)$ in Equation 10 as $(\sum_{x_i \in S} f(w;x_i)/|S|)^{-\frac{1}{2}}$ and $(\sum_{x_i \in S_d} f(w;x_i)/|S_d|)^{-\frac{1}{2}}$, resp., then the optimization problem defined in Equation 10 can be converted to a classic recommendation problem, i.e., the root mean square error minimization problem.

Three popular datasets are adopted in the experiments: MovieLens 1M dataset (6,040 users, 3,706 items, $\sim 10^6$ ratings). MovieLens 10M ($\sim$70k users, 10k items, 10^7 ratings) and Netflix ($\sim$480k users, 18k items, 10^8 ratings). For each dataset, we randomly split it into training and test sets and keep the ratio of training set to test set as 9:1.

For ERMMA, we consider all the options including s and λ, and use learning rate $v = 0.001$ for stochastic gradient decent, $\mu = 0.06$ for regularization coefficient, $\epsilon = 0.0001$ for gradient descent convergence threshold, and $T = 250$ for maximum number of iterations. For RSVD, BPMF, we use the same parameter values provided in the

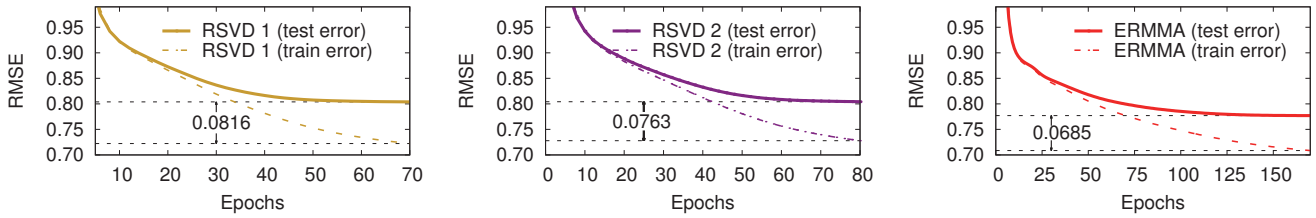

Figure 1: Training error vs. test error with varying number of epochs for three different methods: 1) RSVD 1: RSVD trained with standard SGD (left); 2) RSVD 2: RSVD trained with SGD by randomly setting 10% of gradient updates to 0 (middle); and 3) ERMMA trained with standard SGD by randomly shrinking 10% of gradient updates with 0.5 (right) on MovieLens 10M dataset.

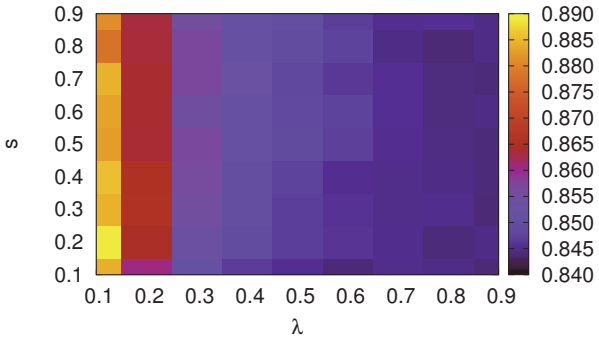

Figure 2: The performance of ERMMA with different s and λ on MovieLens 1M dataset.

original papers (Paterek 2007; Salakhutdinov and Mnih 2008; Li et al. 2016). For SMA, all parameters were set to default values in their implementation [1]. For GSMF, we select $\alpha = 1.0$, $\beta = 70$, $\lambda = 0.05$ and rank $r = 20$. For LLORMA, we choose learning rate $v = 0.001$ and regularization coefficient $\mu = 0.01$, and the number of local models $z = 50$. This is a slight modification of the original LLORMA experimental setup, where better performance can be achieved. For WEMAREC, we adopt learning rate $v = 0.002$ and regularization coefficient $\mu = 0.01$, and default values provided in the source code for unstated parameters (such as the ensemble weights and the maximum number of iterations in clustering).

Generalization Error and Expected Risk Analysis

Figure 1 compares the trends of training and test errors with the number of epochs on a classic matrix approximation method — RSVD (Paterek 2007) and ERMMA. More specifically, we compare three cases: 1) RSVD trained by standard SGD; 2) RSVD trained with SGD by randomly setting a fraction of gradient updates to 0 ($s = 0.1$); and 3) ERMMA trained with standard SGD ($s = 0.1$, $\lambda = 0.5$). As we can see from Figure 1, RSVD 2 can indeed decrease the discrepancy between training and test errors compared with RSVD 1, but the test accuracy has negligible improvement due to the increase in training error. In contrast, ERMMA can a) better reduce the discrepancy between training and test errors and b) yield lower test error than the other two methods. This confirms that ERRMA can achieve both lower generalization error and lower expected risk than the other two methods.

[1] https://github.com/ldscc/StableMA.git

Table 1: Root mean square error (RMSE) comparison between ERMMA (rank = 250) and six state-of-the-art matrix approximation-based collaborative filtering methods — RSVD (Paterek 2007), BPMF (Salakhutdinov and Mnih 2008), GSMF (Yuan et al. 2014), LLORMA (Lee et al. 2013), WEMAREC (Chen et al. 2015) and SMA (Li et al. 2016). Note that, ERMMA statistically significantly outperforms the other methods with 95% confidence level.

Method	MovieLens (10M)	Netflix
RSVD	0.8256 ± 0.0006	0.8534 ± 0.0001
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	$\mathbf{0.7670 \pm 0.0007}$	$\mathbf{0.8018 \pm 0.0001}$

Sensitivity Analysis

The expected risk bound of ERMMA depends on two key parameters: 1) the ratio of randomly selected examples whose gradient updates are shrank — s; and 2) the shrinkage coefficient for the selected examples — λ. Here, we show how ERMMA performs with different combinations of s and λ. As shown in Figure 2, ERMMA achieves good accuracy when s and λ are greater than 0.5. Moreover, ERMMA achieves almost optimal accuracy when s, λ is around $(0.8, 0.8)$, so that we adopt $s = 0.8$ and $\lambda = 0.8$ for the following experiments.

Note that, regularization term and adaptive learning steps also have impacts on ERMMA's performance, and their effectiveness are analyzed in the supplementary material.

Performance Comparison in Rating Prediction

Table 1 compares the recommendation accuracy between the proposed method and six state-of-the-art MA-based collaborative filtering methods in rating prediction task, and root mean square error (RMSE) is adopted to measure the performance. Note that, LLORMA (Lee et al. 2013) and WEMAREC (Chen et al. 2015) are matrix approximation-based ensemble methods, which have been proved to be more accurate than single matrix approximation methods. However,

Table 2: AP and NDCG@10 comparisons between ERMMA and six state-of-the-art matrix approximation-based collaborative filtering methods on Movielens 1M and Movielens 10M datasets. Note that, ERMMA statistically significantly outperforms the other methods with 95% confidence level.

Metric		Average Precision			NDCG@10		
Data	Method	N=5	N=20	N=50	N=5	N=20	N=50
Movielens 1M	RSVD	0.7473 ± 0.0005	0.7230 ± 0.0008	0.7207 ± 0.0009	0.6423 ± 0.0024	0.6456 ± 0.0010	0.6348 ± 0.0025
	BPMF	0.6352 ± 0.0011	0.6356 ± 0.0009	0.6719 ± 0.0103	0.5006 ± 0.0030	0.4915 ± 0.0016	0.5467 ± 0.0169
	GSMF	0.7041 ± 0.0008	0.7087 ± 0.0018	0.7382 ± 0.0017	0.6031 ± 0.0034	0.6236 ± 0.0016	0.6522 ± 0.0011
	LLORMA	0.7446 ± 0.0008	0.7784 ± 0.0008	0.7916 ± 0.0010	0.6200 ± 0.0015	0.7247 ± 0.0031	0.7554 ± 0.0019
	WEMAREC	0.7492 ± 0.0021	0.7821 ± 0.0008	0.7849 ± 0.0011	0.6527 ± 0.0047	0.6927 ± 0.0016	0.6948 ± 0.0016
	SMA	0.7672 ± 0.0007	0.7850 ± 0.0011	0.7914 ± 0.0005	0.6618 ± 0.0041	0.7378 ± 0.0020	0.7560 ± 0.0015
	ERMMA	$\mathbf{0.7689 \pm 0.0007}$	$\mathbf{0.7864 \pm 0.0007}$	$\mathbf{0.7925 \pm 0.0012}$	$\mathbf{0.6639 \pm 0.0027}$	$\mathbf{0.7419 \pm 0.0012}$	$\mathbf{0.7574 \pm 0.0005}$
Movielens 10M	RSVD	0.6804 ± 0.0007	0.6518 ± 0.0004	0.6901 ± 0.0011	0.6157 ± 0.0008	0.5863 ± 0.0005	0.6088 ± 0.0014
	BPMF	0.5707 ± 0.0001	0.5715 ± 0.0002	0.5618 ± 0.0010	0.5007 ± 0.0003	0.5084 ± 0.0007	0.4895 ± 0.0018
	GSMF	0.6053 ± 0.0033	0.6655 ± 0.0005	0.7273 ± 0.0016	0.5512 ± 0.0024	0.5943 ± 0.0004	0.6439 ± 0.0023
	LLORMA	0.7199 ± 0.0002	0.7407 ± 0.0005	0.7612 ± 0.0002	0.6358 ± 0.0005	0.6676 ± 0.0009	0.6924 ± 0.0004
	WEMAREC	0.7162 ± 0.0019	0.7407 ± 0.0002	0.7535 ± 0.0010	0.6367 ± 0.0023	0.6794 ± 0.0004	0.6934 ± 0.0008
	SMA	0.7284 ± 0.0004	0.7452 ± 0.0004	0.7536 ± 0.0006	0.6562 ± 0.0010	0.6790 ± 0.0009	0.6856 ± 0.0007
	ERMMA	$\mathbf{0.7298 \pm 0.0005}$	$\mathbf{0.7492 \pm 0.0004}$	$\mathbf{0.7632 \pm 0.0003}$	$\mathbf{0.6575 \pm 0.0003}$	$\mathbf{0.6834 \pm 0.0009}$	$\mathbf{0.6976 \pm 0.0005}$

the proposed method statistically significantly outperforms all the six methods. This confirms that ERMMA can achieve lower expected risk than all the other methods. Meanwhile, the computational complexity of ERMMA is the same as RSVD and SMA, i.e., $O(rmn)$ per-iteration where m, n is the number of users and items respectively, r is the rank.

Performance Comparison in Item Ranking

Table 2 compares the accuracy of ERMMA with the other six MA-based collaborative filtering methods in item ranking task. Average precision (AP) and normalized discounted cumulative gain (NDCG) are adopted to measure the performance. Note that, we only compare the methods on Movielens 1M and Movielens 10M due to the memory limitation of our server on Netflix. In this experiment, we fix the numbers of ratings as $N = 5, 20, 50$ to form different training sets and keep the rest ratings in the test sets. As shown in Table 2, ERMMA achieves better accuracy in both AP and NDGC@10 compared with all the other methods. This further confirms that ERMMA can achieve lower expected risk than the other methods.

Related Work

Uniform stability was first proposed by Bousquet and Elisseeff (2001), which can be adopted to obtain bounds on generalization error of learning algorithms by their stability properties. Prior works have shown that a variety of learning algorithms possess the uniform stability property, e.g., regularization networks (Bousquet and Elisseeff 2001), learning to rank algorithms (Agarwal and Niyogi 2009; Lan et al. 2008), etc. Recently, Hardt et al. proved that parametric models trained using SGD with limited iterations have vanishing generalization error (Hardt, Recht, and Singer 2015). However, lower generalization error cannot guarantee lower expected risk bound because generalization error is only part of the expected risk bound. Therefore, it is more desirable to design learning algorithms which can achieve low expected risk bounds rather than only low generalization error bounds. Different from their works, this paper focuses on the expected risk minimization problem and proposes expected risk minimized matrix approximation method, which has the potential to achieve lower expected risk based on our theoretical analysis and empirical studies.

Matrix approximation methods have been extensively studied in the context of recommender systems. Paterek (2007) applied regularized singular value decomposition (RSVD) in the Netflix Prize contest. Later, Koren (2008) novelly proposed a more accurate model — SVD++, which can effectively combine matrix factorization and neighborhood model. On the other side, Salakhutdinov and Mnih (2007) proposed Probabilistic Matrix Factorization (PMF) by viewing matrix factorization from a probabilistic perspective. Based on this, they further proposed Bayesian Probabilistic Matrix Factorization (BPMF) (Salakhutdinov and Mnih 2008), in which fully Bayesian treatment is given to PMF. In addition, some recent works also attempted to train a singleton model integrated with multiple types of relations by using multi-task feature learning techniques (Yuan et al. 2014; Chen et al. 2016). The above methods tried to solve the classic empirical risk minimization problems in model training, so that generalization error or expected risk cannot be directly minimized in those methods.

Recently, Srebro et al. (2004) analyzed the generalization error bounds of collaborative prediction with low-rank matrix approximation for binary recommendation problem. Li et al. (2016) proposed the stable matrix approximation method, which can improve the generalization performance of matrix approximation by increasing the stability of MA methods. Meanwhile, ensemble matrix approximation methods have also been proposed to improve generalization performance of MA methods by ensemble learning, e.g., DFC (Mackey, Jordan, and Talwalkar 2011), LLORMA (Lee et al. 2013), WEMAREC (Chen et al. 2015) etc. The above works only consider how to achieve better generalization performance. However, as analyzed in this paper, minimizing generalization error cannot ensure minimum expected risk because optimization error will typically increase when minimizing generalization error. Therefore, minimizing expected risk in

ERMMA should be more desirable than only minimizing generalization error in the above works.

Conclusion

This paper proposes a new matrix approximation method — ERMMA, which can minimize the expected risk of matrix approximation models by achieving better tradeoffs between optimization error and generalization error. Theoretical analysis shows that ERMMA can yield lower expected risk bound compared with other methods. Empirical studies on real-world datasets also demonstrate that ERMMA can achieve better performance than minimizing empirical risk in classic matrix approximation methods. Furthermore, by introducing proper regularization term and adaptive learning steps, ERMMA can achieve better accuracy than state-of-the-art matrix approximation methods in both the rating prediction task and the item ranking task.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grant No. 61233016, and the National Science Foundation of USA under Grant Nos. 1251257, 1334351, 1442971, 1620345, and 1639792.

References

Agarwal, S., and Niyogi, P. 2009. Generalization bounds for ranking algorithms via algorithmic stability. *Journal of Machine Learning Research* 10:441–474.

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

Candès, E. J., and Plan, Y. 2010. Matrix completion with noise. *Proceedings of the IEEE* 98(6):925–936.

Candès, E. J., and Recht, B. 2009. Exact matrix completion via convex optimization. *Foundations of Computational mathematics* 9(6):717–772.

Chen, C.; Li, D.; Zhao, Y.; Lv, Q.; and Shang, L. 2015. 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 (SIGIR '15)*, 303–312.

Chen, C.; Li, D.; Lv, Q.; Yan, J.; Chu, S. M.; and Shang, L. 2016. MPMA: Mixture probabilistic matrix approximation for collaborative filtering. In *Proceedings of The 25th International Joint Conference on Artificial Intelligence (IJCAI '16)*, 1382–1388.

Duchi, J.; Hazan, E.; and Singer, Y. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research* 12:2121–2159.

Ekstrand, M. D.; Riedl, J. T.; and Konstan, J. A. 2011. Collaborative filtering recommender systems. *Foundations and Trends in Human-Computer Interaction* 4(2):81–173.

Gemulla, R.; Nijkamp, E.; Haas, P. J.; and Sismanis, Y. 2011. Large-scale matrix factorization with distributed stochastic gradient descent. In *Proceedings of the 17th ACM international conference on Knowledge discovery and data mining ((SIGKDD '11))*, 69–77. ACM.

Hardt, M.; Recht, B.; and Singer, Y. 2015. Train faster, generalize better: Stability of stochastic gradient descent. arXiv:1509.01240.

Koren, Y.; Bell, R.; Volinsky, C.; et al. 2009. Matrix factorization techniques for recommender systems. *Computer* 42(8):30–37.

Koren, Y. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM international conference on Knowledge discovery and data mining (SIGKDD '08)*, 426–434.

Lan, Y.; Liu, T.-Y.; Qin, T.; Ma, Z.; and Li, H. 2008. Query-level stability and generalization in learning to rank. In *Proceedings of the 25th international conference on Machine learning (ICML' 08)*, 512–519. ACM.

Lee, J.; Kim, S.; Lebanon, G.; and Singer, Y. 2013. Local low-rank matrix approximation. In *Proceedings of The 30th International Conference on Machine Learning (ICML '13)*, 82–90.

Li, Y.; Zhou, Y.; Yan, J.; Yang, J.; and He, X. 2010. Tensor error correction for corrupted values in visual data. In *IEEE International Conference on Image Processing (ICIP '10)*, 2321–2324.

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

Mackey, L. W.; Jordan, M. I.; and Talwalkar, A. 2011. Divide-and-conquer matrix factorization. In *Advances in Neural Information Processing Systems*, 1134–1142.

Mazumder, R.; Hastie, T.; and Tibshirani, R. 2010. Spectral regularization algorithms for learning large incomplete matrices. *Journal of machine learning research* 11(Aug):2287–2322.

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

Ng, A. Y. 2004. Feature selection, L1 vs. L2 regularization, and rotational invariance. In *Proceedings of the 21st International Conference on Machine Learning (ICML '04)*, 78–85. ACM.

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

Salakhutdinov, R., and Mnih, A. 2007. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, 1257–1264.

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

Srebro, N.; Alon, N.; and Jaakkola, T. S. 2004. Generalization error bounds for collaborative prediction with low-rank matrices. In *Advances in Neural Information Processing Systems*, 1321–1328.

Srebro, N., and Shraibman, A. 2005. Rank, trace-norm and max-norm. In *International Conference on Computational Learning Theory*, 545–560. Springer.

Yan, J.; Zhu, M.; Liu, H.; and Liu, Y. 2010. Visual saliency detection via sparsity pursuit. *IEEE Signal Processing Letters* 17(8):739–742.

Yuan, T.; Cheng, J.; Zhang, X.; Qiu, S.; and Lu, H. 2014. Recommendation by mining multiple user behaviors with group sparsity. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI '14)*, 222–228.

Zeiler, M. D. 2012. Adadelta: an adaptive learning rate method. arXiv:1212.5701.