Joint Capped Norms Minimization for Robust Matrix Recovery

Feiping Nie[1], **Zhouyuan Huo**[2], **Heng Huang**[2*]
[1]School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL),
Northwestern Polytechnical University, Xian 710072, Shaanxi, P. R. China.
[2]Computer Science and Engineering, University of Texas at Arlington, USA.
feipingnie@gmail.com, huozhouyuan@gmail.com, heng@uta.edu

Abstract

The low-rank matrix recovery is an important machine learning research topic with various scientific applications. Most existing low-rank matrix recovery methods relax the rank minimization problem via the trace norm minimization. However, such a relaxation makes the solution seriously deviate from the original one. Meanwhile, most matrix recovery methods minimize the squared prediction errors on the observed entries, which is sensitive to outliers. In this paper, we propose a new robust matrix recovery model to address the above two challenges. The joint capped trace norm and capped ℓ_1-norm are used to tightly approximate the rank minimization and enhance the robustness to outliers. The evaluation experiments are performed on both synthetic data and real world applications in collaborative filtering and social network link prediction. All empirical results show our new method outperforms the existing matrix recovery methods.

1 Introduction

As a challenging machine learning problem, matrix recovery is to impute the missing entries of the given data matrix, and has many scientific applications [Srebro *et al.*, 2004; Rennie and Srebro, 2005; Abernethy *et al.*, 2009], such as friendship prediction in social network, rating value estimation in recommendation system and collaborative filtering, link prediction in protein-protein interaction network. As one emerging technique of compressive sensing, the problem of matrix recovery has been extensively studied on both theory and algorithms [Candès and Recht, 2009; Candes and Tao, 2009; Recht *et al.*, 2010; Mazumder *et al.*, 2009; Cai *et al.*, 2008; Rennie and Srebro, 2005; Nie *et al.*, 2012; 2015].

Most existing matrix recovery methods assume that the values in the data matrix are correlated and the rank of the data matrix is low. The missing entries can be recovered using the observed entries by minimizing the rank of the data matrix, which is an NP hard problem. Instead of solving this NP hard problem, the researchers minimize the trace norm (the sum of the singular values of the data matrix) as the convex relaxation of the rank function. Many recent research has been focusing on solving such trace norm minimization problem [Toh and Yun, 2010; Ji and Ye, 2009; Liu *et al.*, 2009; Ma *et al.*, 2009; Mazumder *et al.*, 2009]. Meanwhile, instead of strictly keeping the values of the observed entries, the recent research work relaxed it to minimize the prediction errors (using squared error function) on the observed entries. Although the trace norm minimization based matrix recovery objective is a convex problem with global solution, the relaxation makes the final solution seriously deviate from the original one. It is desired to solve a better approximation of the rank minimization problem.

In this paper, we propose a novel matrix recovery model using the joint capped norms, which have been recently studied in machine learning community [Zhang, 2008; 2010; Sun *et al.*, 2013]. First, we use the capped trace norm to approximate the rank minimization problem. Because the capped trace norm only minimizes several smallest eigen values, our new objective can approximate the rank minimization better than the trace norm. Moreover, to improve the robustness of matrix recovery method, we introduce the capped ℓ_1-norm error function for the prediction errors on the observed entries. Thus, our new objective minimizes the joint capped trace norm and capped ℓ_1-norm. Our objective is more robust and effective than the standard matrix recovery methods. Although our objective function is not a convex problem, we derive a new efficient optimization algorithm with rigorous convergence analysis. We evaluate our new method using both synthetic and real world data sets. The benchmark data sets from collaborative filtering and social network link prediction applications are utilized in our validations. All empirical results show our new robust matrix recovery method outperforms the standard missing value prediction approaches. In summary, we highlight the main contributions of this paper as follows:

1. We propose a novel objective function for the robust matrix recovery task using the joint capped trace norm and capped ℓ_1-norm.
2. Optimizing the objective function is a non-trivial problem, thus we derive a new optimization algorithm to

*Corresponding Author. This work was partially supported by the U.S. NIH R01 AG049371, NSF IIS 1302675, IIS 1344152, DBI 1356628, IIS 1619308, IIS 1633753.

solve the proposed objective with rigorous convergence analysis.

3. Our proposed new capped norm optimization algorithm is general and can be applied to solve other capped norm based problems.

2 Robust Matrix Recovery via Joint Capped Norms

2.1 Definitions of Capped ℓ_1-Norm and Capped Trace Norm

The ℓ_1-norm of a matrix X can be defined as $\|X\|_1 = \sum_{i,j} |x_{ij}|$ and the trace norm is defined as $\|X\|_* = \sum_i \sigma_i(X)$, where $\sigma_i(X)$ is the i-th singular value of X. Based on these definitions, the capped ℓ_1-norm of a matrix X is defined as:

$$\|X\|_{C_1^\varepsilon} = \sum_{i,j} \min(|x_{ij}|, \varepsilon) \tag{1}$$

and the capped trace norm is defined as:

$$\|X\|_{C_*^\varepsilon} = \sum_i \min(\sigma_i(X), \varepsilon) \tag{2}$$

The capped trace norm is a better approximation to rank minimization than the trace norm. When the largest singular values have large changes, the rank of the matrix could keep the same value but the trace norm will definitely change largely. Because the capped trace norm only minimizes small singular values, the capped trace norm won't have large change or even stays at the same value.

2.2 Proposed Robust Matrix Recovery Model

Denote $X_\Omega = \{x_{ij} \,|(i,j) \in \Omega\}$, and $\|X_\Omega\|_F^2 = \sum_{(i,j)\in\Omega} x_{ij}^2$. Suppose in a matrix T the observed values are $T_\Omega = \{t_{ij} \,|(i,j) \in \Omega\}$, the matrix recovery task is to predict the unobserved values in the matrix T. This task is usually addressed by solving a rank minimization problem as follows:

$$\min_X \|X_\Omega - T_\Omega\|_F^2 + \gamma \text{rank}(X) \tag{3}$$

To make the problem easier to solve, in practice, the rank is relaxed to the trace norm and then we solve the following relaxed problem:

$$\min_X \|X_\Omega - T_\Omega\|_F^2 + \gamma \|X\|_* \tag{4}$$

However, the relaxation makes the solution deviate seriously from the original one. Moreover, the applied squared error is sensitive to outliers.

In this paper, we use the joint capped norms to address these disadvantages in Eq. (4). First, we replace the squared F-norm loss function by the capped ℓ_1-norm. As can be seen in Eq. (1), the capped ℓ_1-norm can be used as a robust loss function since the function value is bounded to ε no matter how the input value deviated from zero. In practice, outliers usually make the input values largely deviate from zero, thus the bounded property in the capped norm will make it very robust to outliers. Second, we replace the trace norm by capped trace norm. As can be seen in Eq. (2), the capped trace norm is a better approximation to the rank function than trace norm does, and the large singular values have less effect in the learning.

Based on the above analysis, we propose to solve the following problem for robust matrix recovery task:

$$\min_X \|X_\Omega - T_\Omega\|_{C_1^{\varepsilon_1}} + \gamma \|X\|_{C_*^{\varepsilon_2}} \tag{5}$$

which can be written as [1]

$$\min_X \sum_{(i,j)\in\Omega} \min(|x_{ij} - t_{ij}|, \varepsilon_1) + \gamma \sum_i \min(\sigma_i(X), \varepsilon_2) \tag{6}$$

This problem looks very difficult to solve, however, inspired by the reweighted method [Nie *et al.*, 2010; 2014; 2017], we will propose a very simple algorithm to solve it, and prove its convergence in the next section.

3 A New Algorithm to Solve Joint Capped Norm Optimization

3.1 Proposed Algorithm

The proposed algorithm to solve problem (6) is described in Alg. 1. Note that the problem (8) in the iteration can be easily solved and has closed form solution. The algorithm is derived from sub-gradient analysis. If the algorithm converges, the converged solution will make zero belongs to the sub-gradient of the problem (6). Therefore, the key problem is to prove the algorithm indeed converges.

Algorithm 1 Algorithm to solve the problem (6).

Initialize X.

while not converge **do**

1. For each $(i,j) \in \Omega$, update the (i,j)-th element of S by

$$s_{ij} = \begin{cases} \frac{1}{2|x_{ij} - t_{ij}|}, & |x_{ij} - t_{ij}| < \varepsilon_1 \\ 0, & otherwise \end{cases}. \tag{7}$$

2. Compute the SVD of $X = U\Sigma V^T$. Without loss of generality, suppose the singular values σ_i are sorted from the smallest one to the largest one, and there are k singular values smaller than ε_2. Compute $D = \frac{1}{2}\sum_{i=1}^k \sigma_i^{-1} u_i u_i^T$.

3. Update X by the optimal solution to the problem

$$\min_X \sum_{(i,j)\in\Omega} s_{ij}(x_{ij} - t_{ij})^2 + \gamma Tr(X^T D X). \tag{8}$$

end while

[1] In practice, to make the problem derivable, $|x|$ and σ are replaced by $\sqrt{x^2 + \varepsilon}$ and $\sqrt{\sigma^2 + \varepsilon}$, respectively. It can be seen when $\varepsilon \to 0$, it approximates the original problem.

3.2 Proof of Convergence

Before proving the convergence of the algorithm, we prepare the following lemmas.

Lemma 1 ([Marshall *et al.*, 2009], pages: 340) *For any two hermitian matrices $A, B \in \mathbb{R}^{n\times n}$, suppose the eigenvalues $\lambda_i(A)$ are sorted with same order, then we have the following inequality:*

$$\sum_{i=1}^{n}\lambda_i(A)\lambda_{n-i+1}(B) \le tr(AB) \le \sum_{i=1}^{n}\lambda_i(A)\lambda_i(B)$$

Lemma 2 *Given two matrices X and $\tilde{X}$, the SVD of X and $\tilde{X}$ are $X = U\Sigma V^T$ and $\tilde{X} = \tilde{U}\tilde{\Sigma}\tilde{V}^T$, respectively. Without loss of generality, suppose the singular values σ_i, $\tilde{\sigma}_i$ are sorted from the smallest one to the largest one respectively, and there are k singular values σ_i, $\tilde{k}$ singular values $\tilde{\sigma}_i$ smaller than ε_2 respectively, then we have:*

$$\begin{aligned}&\sum_i \min(\tilde{\sigma}_i(\tilde{X}),\varepsilon_2) - \tfrac{1}{2}Tr(\sum_{i=1}^{k}\sigma_i^{-1}u_iu_i^T\tilde{X}\tilde{X}^T)\\ &\le \sum_i \min(\sigma_i(X),\varepsilon_2) - \tfrac{1}{2}Tr(\sum_{i=1}^{k}\sigma_i^{-1}u_iu_i^TXX^T)\end{aligned} \tag{9}$$

Proof: Since there are $\tilde{k}$ singular values $\tilde{\sigma}_i$ smaller than ε_2, it can be easily verified that for any k and $\tilde{k}$ ($k \ge \tilde{k}$ or $k < \tilde{k}$), we have $\sum_{i=1}^{\tilde{k}}(\tilde{\sigma}_i - \varepsilon_2) \le \sum_{i=1}^{k}(\tilde{\sigma}_i - \varepsilon_2)$, which is equal to

$$\sum_{i=1}^{\tilde{k}}\tilde{\sigma}_i - \tilde{k}\varepsilon_2 \le \sum_{i=1}^{k}\tilde{\sigma}_i - k\varepsilon_2 \tag{10}$$

Starting from $(\sigma_i - \tilde{\sigma}_i)^2 \ge 0$, we have

$$\begin{aligned}&\sigma_i^2 - 2\sigma_i\tilde{\sigma}_i + \tilde{\sigma}_i^2 \ge 0\\ &\Rightarrow \tilde{\sigma}_i - \frac{1}{2}\sigma_i^{-1}\tilde{\sigma}_i^2 \le \frac{1}{2}\sigma_i\\ &\Rightarrow \sum_{i=1}^{k}\tilde{\sigma}_i - \sum_{i=1}^{k}\frac{1}{2}\sigma_i^{-1}\tilde{\sigma}_i^2 \le \frac{1}{2}\sum_{i=1}^{k}\sigma_i\end{aligned} \tag{11}$$

Sum over Eq. (10) and Eq. (11) in two sides, we have

$$\sum_{i=1}^{\tilde{k}}\tilde{\sigma}_i - \tilde{k}\varepsilon_2 - \frac{1}{2}\sum_{i=1}^{k}\sigma_i^{-1}\tilde{\sigma}_i^2 \le \frac{1}{2}\sum_{i=1}^{k}\sigma_i - k\varepsilon_2 \tag{12}$$

which is equal to

$$\sum_{i=1}^{\tilde{k}}\tilde{\sigma}_i + (n-\tilde{k})\varepsilon_2 - \frac{1}{2}\sum_{i=1}^{k}\sigma_i^{-1}\tilde{\sigma}_i^2 \le \frac{1}{2}\sum_{i=1}^{k}\sigma_i + (n-k)\varepsilon_2$$

According to Lemma 1, we have

$$\begin{aligned}&\frac{1}{2}Tr(\sum_{i=1}^{k}\sigma_i^{-1}u_iu_i^T\tilde{X}\tilde{X}^T) = \frac{1}{2}Tr(U\Sigma^{-1}U^T\tilde{U}\tilde{\Sigma}^2\tilde{U}^T)\\ &\ge \frac{1}{2}\sum_{i=1}^{k}\sigma_i^{-1}\tilde{\sigma}_i^2\end{aligned} \tag{13}$$

which is equal to

$$-\frac{1}{2}Tr(\sum_{i=1}^{k}\sigma_i^{-1}u_iu_i^T\tilde{X}\tilde{X}^T) \le -\frac{1}{2}\sum_{i=1}^{k}\sigma_i^{-1}\tilde{\sigma}_i^2 \tag{14}$$

Sum over Eq. (13) and Eq. (14) in two sides, we have

$$\begin{aligned}&\sum_{i=1}^{\tilde{k}}\tilde{\sigma}_i + (n-\tilde{k})\varepsilon_2 - \frac{1}{2}Tr(\sum_{i=1}^{k}\sigma_i^{-1}u_iu_i^T\tilde{X}\tilde{X}^T)\\ &\le \frac{1}{2}\sum_{i=1}^{k}\sigma_i + (n-k)\varepsilon_2\end{aligned} \tag{15}$$

Note that

$$\begin{aligned}&\frac{1}{2}Tr(\sum_{i=1}^{k}\sigma_i^{-1}u_iu_i^TXX^T)\\ &= \frac{1}{2}Tr(U\Sigma^{-1}U^TU\Sigma^2U^T) = \frac{1}{2}\sum_{i=1}^{k}\sigma_i\end{aligned} \tag{16}$$

According to Eq. (15) and Eq. (16), we arrive at

$$\begin{aligned}&\sum_{i=1}^{\tilde{k}}\tilde{\sigma}_i + (n-\tilde{k})\varepsilon_2 - \tfrac{1}{2}Tr(\sum_{i=1}^{k}\sigma_i^{-1}u_iu_i^T\tilde{X}\tilde{X}^T) \le\\ &\sum_{i=1}^{k}\sigma_i + (n-k)\varepsilon_2 - \tfrac{1}{2}Tr(\sum_{i=1}^{k}\sigma_i^{-1}u_iu_i^TXX^T)\end{aligned} \tag{17}$$

which is equal to Eq. (9). □

Lemma 3 *Given $s = \begin{cases}\frac{1}{2|e|}, & |e| < \varepsilon_1\\ 0, & otherwise\end{cases}$, we have the following inequality:*

$$\min(|\tilde{e}|,\varepsilon_1) - s\tilde{e}^2 \le \min(|e|,\varepsilon_1) - se^2. \tag{18}$$

Proof: If $|e| < \varepsilon_1$, then $s = \frac{1}{2}|e|^{-1}$. According to Lemma 2 (in the case that X and $\tilde{X}$ are scalar), we have $\min(|\tilde{e}|,\varepsilon_1) - \frac{1}{2}|e|^{-1}\tilde{e}^2 \le \min(|e|,\varepsilon_1) - \frac{1}{2}|e|^{-1}e^2$, thus Eq. (18) holds.

If $|e| \ge \varepsilon_1$, then $s = 0$. Obviously, $\min(|\tilde{e}|,\varepsilon_1) \le \varepsilon_1 = \min(|e|,\varepsilon_1)$ in this case, thus Eq. (18) also holds. Therefore, Eq. (18) holds in any cases. □

Now we are ready to prove the following theorem, which is the main result of this paper.

Theorem 1 *Algorithm 1 will decrease the objective value of problem (6) in each iteration until it converges.*

Proof: Suppose the updated X is $\tilde{X}$ in step 3 of Algorithm 1. Denote the SVD of X and $\tilde{X}$ are $X = U\Sigma V^T$ and $\tilde{X} = \tilde{U}\tilde{\Sigma}\tilde{V}^T$, respectively. Without loss of generality, suppose the singular values σ_i, $\tilde{\sigma}_i$ are sorted from the smallest one to the largest one respectively, and there are k singular values σ_i, $\tilde{k}$ singular values $\tilde{\sigma}_i$ smaller than ε_2 respectively.

Since $\tilde{X}$ is the optimal solution to Eq. (8), we have

$$\begin{aligned}&\sum_{(i,j)\in\Omega}s_{ij}(\tilde{x}_{ij}-t_{ij})^2 + \tfrac{\gamma}{2}Tr(D\tilde{X}\tilde{X}^T)\\ &\le \sum_{(i,j)\in\Omega}s_{ij}(x_{ij}-t_{ij})^2 + \tfrac{\gamma}{2}Tr(DXX^T)\end{aligned} \tag{19}$$

According to the definition of D in step 2 of Algorithm 1, Eq.(19) can be written as

$$\sum_{(i,j)\in\Omega} s_{ij}(\tilde{x}_{ij}-t_{ij})^2 + \frac{\gamma}{2}Tr(\sum_{i=1}^{k}\sigma_i^{-1}u_i u_i^T \tilde{X}\tilde{X}^T) \leq \sum_{(i,j)\in\Omega} s_{ij}(x_{ij}-t_{ij})^2 + \frac{\gamma}{2}Tr(\sum_{i=1}^{k}\sigma_i^{-1}u_i u_i^T XX^T) \tag{20}$$

According to the definition of s_{ij} in step 1 of Algorithm 1, and according to Lemma 3, we have

$$\sum_{(i,j)\in\Omega} \min(|\tilde{x}_{ij}-t_{ij}|,\varepsilon_1) - \sum_{(i,j)\in\Omega} s_{ij}(\tilde{x}_{ij}-t_{ij})^2 \leq \sum_{(i,j)\in\Omega} \min(|x_{ij}-t_{ij}|,\varepsilon_1) - \sum_{(i,j)\in\Omega} s_{ij}(x_{ij}-t_{ij})^2 \tag{21}$$

According to Lemma 2, we have

$$\gamma\sum_i \min(\tilde{\sigma}_i(\tilde{X}),\varepsilon_2) - \frac{\gamma}{2}Tr(\sum_{i=1}^{k}\sigma_i^{-1}u_i u_i^T \tilde{X}\tilde{X}^T) \leq \gamma\sum_i \min(\sigma_i(X),\varepsilon_2) - \frac{\gamma}{2}Tr(\sum_{i=1}^{k}\sigma_i^{-1}u_i u_i^T XX^T) \tag{22}$$

Sum over Eq. (20-22) in two sides, we arrive at

$$\sum_{(i,j)\in\Omega} \min(|\tilde{x}_{ij}-t_{ij}|,\varepsilon_1) + \gamma\sum_i \min(\tilde{\sigma}_i(\tilde{X}),\varepsilon_2) \leq \sum_{(i,j)\in\Omega} \min(|x_{ij}-t_{ij}|,\varepsilon_1) + \gamma\sum_i \min(\sigma_i(X),\varepsilon_2)$$

Since the objective value of problem (6) has a lower bound 0, the Algorithm 1 will converge, and in each iteration the objective value of problem (6) will decreased before the convergence. □

It can be checked that when the algorithm converges, the solution will satisfy the KKT conditions (derivative is zero). In the experiments we observed that the derived algorithm is very fast to converge, usually within 5-10 iterations.

4 Experiments

We evaluate our method on the following data sets: The Jester Jokes data sets [Goldberg *et al.*, 2001][1], Wikipedia [Leskovec *et al.*, 2009] and Epinions [Massa and Avesani, 2006] data sets[2] and Sweetrs data set[3]. There are 7 compared matrix recovery methods in total: SVD, SVT [Cai *et al.*, 2008], RRMC [Huang *et al.*, 2013], ALM [Lin *et al.*, 2010], GROUSE [Balzano *et al.*, 2010], OPTSPACE [Keshavan and Oh, 2009] and our method. In our experiments, all the parameters are selected via 5-fold cross validation to guarantee their best performance. Normalized root mean square error (nRMSE) and normalized mean absolute error (nMAE) are used as evaluation metrics.

$$nRMSE = \frac{\sqrt{mean(|X_{ij}-T_{ij}|^2)}}{t_{max}-t_{min}} \tag{23}$$

$$nMAE = \frac{mean(|X_{ij}-T_{ij}|)}{t_{max}-t_{min}} \tag{24}$$

[1]http://eigentaste.berkeley.edu/dataset/

[2]http://snap.stanford.edu/data/

[3]http://sweetrs.org

4.1 Experiments on Real World Data Sets

Jester Jokes Rating Prediction

Firstly, we present experiment results on Jester Jokes data sets. To make this problem easy to evaluate, we select a part of the most active users from the original data. The sizes of matrix for these three data sets are 10000×100, 13109×140 and 13449×140. In the experiments, we assume 5%, 10%, 15% and 20% ratings of each matrix are known respectively. In each case, training data are selected randomly, and the whole procedure is repeated for 5 times randomly to compute the average performance.

The results of matrix recovery task on Jester jokes data are shown in Table 1. In real life, users are very likely to read just some of the jokes and to rate even fewer. From Table 1, obviously our method could make use of sparse training data, and outperforms other methods in all different cases.

In reference to our objective function, there are three parameters to tune, ε_1, ε_2 and γ. Fig. 1 shows the relationship between ε_1, ε_2 and γ with performance evaluation metric nMAE. In the experiment, we fix two parameters and change another one. Fig. 1a shows that in this occasion, ε_1 plays a insignificant role when there are few outliers. Fig. 1b shows that ε_2 is the most important parameter in this model, for it determines the rank of the matrix indirectly. In our experiments, we input a hypothetical rank value, and the value of the ε_2 is selected based on the singular values of the matrix in the first 5 iterations. After that, ε_2 is fixed to be a constant. In this way, our model can learn a proper value of ε_2 automatically in different situations. It is also very interesting to find out through Fig. 1c that the value of γ also doesn't matter too much on the performance of the model in Jester Jokes data sets.

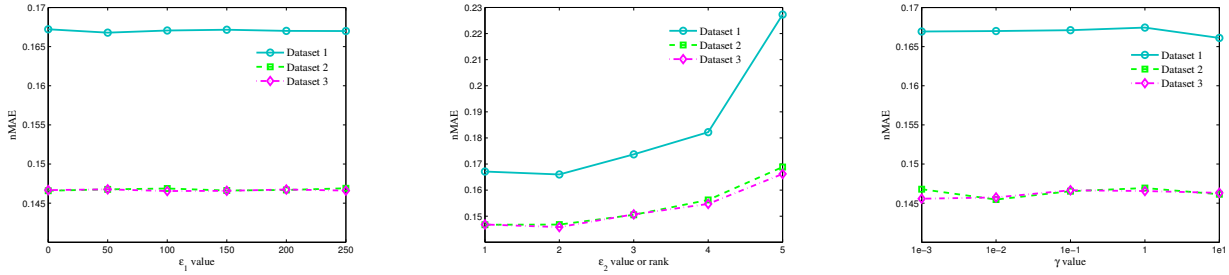
Social Network Linking Prediction

In this section, we show the results of matrix recovery algorithms in predicting social network linking data, Wikipedia and Epinions. In this experiment, in order to alleviate the data skewness and keep the computation manageable, we select top 2,000 highest degree users from each data set. In these experiments, we assume 10% entries of each matrix are known, and the other 90% information are used as testing data.

From Table 2, we can observe that our cRMC method outperforms others in terms of nRMSE and nMAE on both of the two social network trust matrix. Then, we can make use of the prediction matrix to make recommendations. A better recommendation always means better user experience, and it may lead to much profit for these internet enterprises.

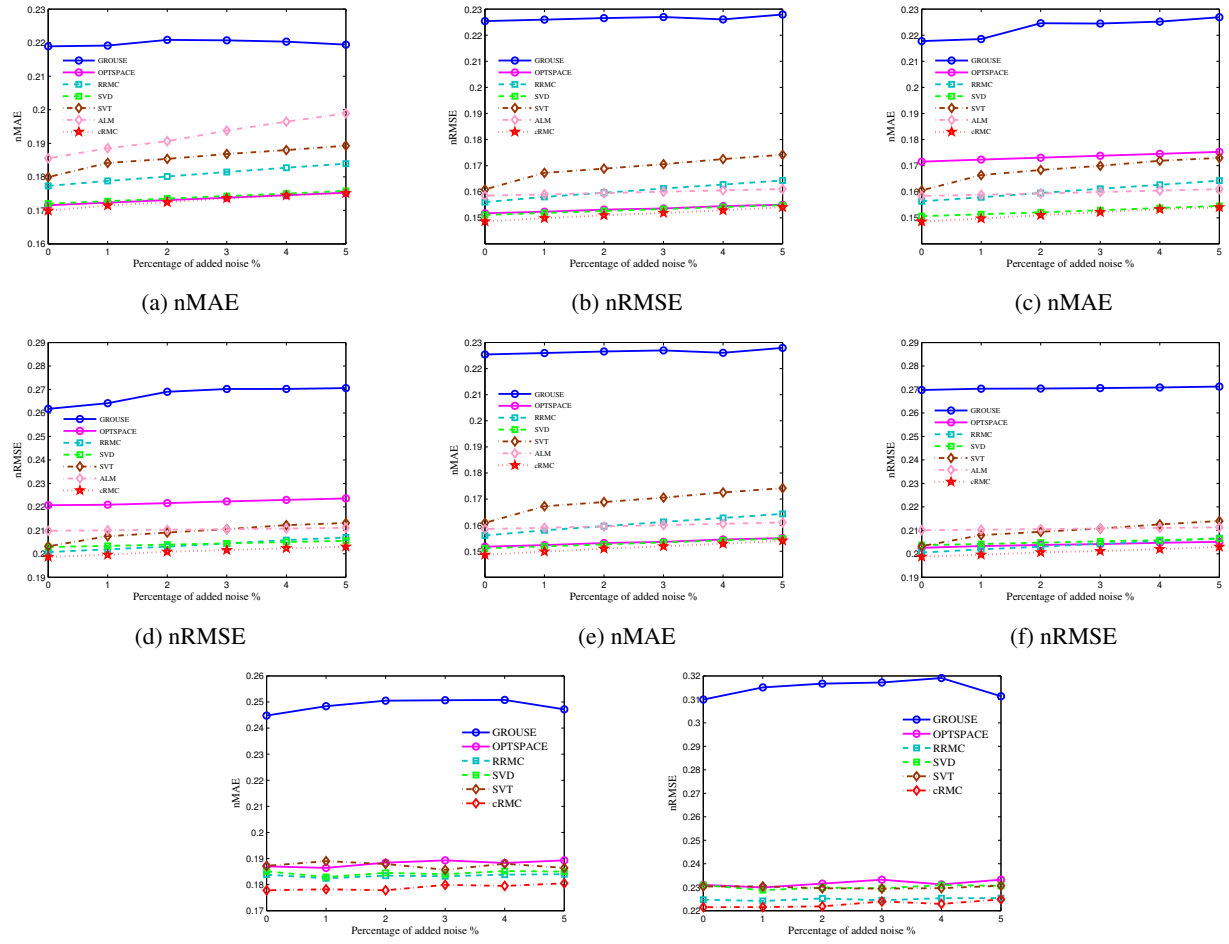
Sweetrs Rating Prediction

In this section, we present experiment results on Sweetrs data set. In this data set, the size of the matrix is 390×47, and we let 10% ratings as training data. Table 2 shows the results of our experiment on Sweetrs data. In the experiment, ALM breaks down during training, hence we ignore this method. In this table, we can find out that the performance of our method is better than the other ones in this case.



(a) ε_1 vs nMAE (b) ε_2 vs nMAE (c) γ vs nMAE

Figure 1: Fig.1a shows the relation between nMAE and ε_1. Fig. 1b represents the effect of ε_2 on nMAE. Fig. 1c displays the change of nMAE with respect to γ.



(a) nMAE (b) nRMSE (c) nMAE

(d) nRMSE (e) nMAE (f) nRMSE

(g) nMAE (h) nRMSE

Figure 2: Rating prediction performance on noisy data sets. Jokes Dataset1: (a) and (b), Jokes Dataset2: (c) and (d), Jokes Dataset3: (e) and (f), Sweetrs: (g) and (h).

4.2 Robustness Evaluation on Synthetic Noise Data

In real life, it usually happens that some users click the wrong ratings by accident or there are even some users who make vicious ratings on purpose. If a method can't deal with these noisy ratings properly, a terrible model will be generated and it may lead to unexpected loss.

To evaluate the robustness of our method and prove its strong ability to deal with outliers, we perform experiments

	Method	5%		10%		15%		20%	
		nRMSE	nMAE	nRMSE	nMAE	nRMSE	nMAE	nRMSE	nMAE
D1	SVD	0.2354	0.1813	0.2261	0.1750	0.2213	0.1720	0.2186	0.1693
	SVT	0.2328	0.1918	0.2253	0.1848	0.2201	0.1799	0.2163	0.1764
	RRMC	0.2296	0.1889	0.2228	0.1820	0.2180	0.1773	0.2142	0.1736
	ALM	0.2387	0.1945	0.2351	0.1878	0.2360	0.1855	0.2400	0.1861
	GROUSE	0.2643	0.2223	0.2626	0.2213	0.2613	0.2199	0.2614	0.2190
	OPTSPACE	0.2405	0.1861	0.2287	0.1765	0.2207	0.1715	0.2178	0.1683
	cRMC	**0.2293**	**0.1804**	**0.2212**	**0.1737**	**0.2161**	**0.1699**	**0.2126**	**0.1672**
D2	SVD	0.2297	0.1658	0.2059	0.1525	0.2091	0.1506	0.2012	0.1492
	SVT	0.2153	0.1720	0.2080	0.1653	0.2032	0.1605	0.2001	0.1573
	RRMC	0.2106	0.1664	0.2047	0.1604	0.2008	0.1563	0.1979	0.1534
	ALM	0.2137	0.1610	0.2110	0.1592	0.2098	0.1584	0.2091	0.1578
	GROUSE	0.2731	0.2282	0.2719	0.2263	0.2671	0.2211	0.2661	0.2204
	OPTSPACE	0.2126	0.1590	0.2057	0.1535	0.2020	0.1506	0.2000	0.1495
	cRMC	**0.2097**	**0.1584**	**0.2022**	**0.1513**	**0.1987**	**0.1486**	**0.1962**	**0.1466**
D3	SVD	0.2135	0.1573	0.2069	0.1531	0.2037	0.1511	0.2019	0.1497
	SVT	0.2155	0.1730	0.2078	0.1655	0.2033	0.1609	0.2002	0.1577
	RRMC	0.2104	0.1661	0.2042	0.1599	0.2005	0.1560	0.1979	0.1534
	ALM	0.2146	0.1614	0.2113	0.1593	0.2100	0.1585	0.2094	0.1581
	GROUSE	0.2721	0.2278	0.2725	0.2269	0.2698	0.2254	0.2716	0.2232
	OPTSPACE	0.2120	0.1589	0.2057	0.1542	0.2026	0.1517	0.2006	0.1499
	cRMC	**0.2091**	**0.1551**	**0.2021**	**0.1512**	**0.1987**	**0.1485**	**0.1964**	**0.1468**

Table 1: Matrix recovery performance on Jester Jokes rating prediction

Method	Wikipedia		Epinions		Sweetrs	
	nRMSE	nMAE	nRMSE	nMAE	nRMSE	nMAE
SVD	0.2726	0.0834	0.3575	0.1349	0.2309	0.1851
SVT	0.2548	0.0781	0.3519	0.1343	0.2303	0.1872
RRMC	0.2432	0.0873	0.3471	0.1535	0.2247	0.1838
ALM	0.2733	0.0844	0.3698	0.1402	-	-
GROUSE	0.2587	0.0868	0.3523	0.1383	0.3100	0.2448
OPTSPACE	0.2457	0.0753	0.3393	0.1350	0.2310	0.1870
cRMC	**0.2386**	**0.0697**	**0.3336**	**0.1206**	**0.2215**	**0.1778**

Table 2: Matrix recovery performance on Wikipedia, Epinions and Sweetrs.

on Jokes and Sweetrs data sets. We impose noise to the observed data from 0% to 5% each time, and all these noises are set to be the largest or the lowest value randomly. In the experiments, we assume 10% of the ratings are known.

In the experiment, bad ratings are treated as outliers. It is obvious that our objective function explicitly take the unclean training data into consideration, and the value of ε_1 is used to leave out the outliers. In this way, we protect the matrix's original low-rank structure from being distorted.

From Fig. 2, we can see that when the noise increases, the prediction accuracy of all methods tend to degenerate slightly and outliers distort the low-rank structure of the original matrix. It is easy to observe our method outperforms other methods consistently, while the percentage of the noise increases from 0% to 5%. It is well proved from these experiments that our method can deal with outliers properly, and keep the correct low-rank structure of a matrix.

5 Conclusions

In this paper, we proposed a new robust matrix recovery method using joint capped trace norm and capped ℓ_1-norm. Our capped trace norm can approximate the rank minimization problem tighter than the trace norm to achieve better matrix recovery results. The capped ℓ_1-norm based error function enhances the robustness of the proposed objective. Both capped trace norm and capped ℓ_1-norm are non-smooth non-convex. To solve this difficult optimization problem, we proposed a new optimization algorithm with rigorous convergence analysis. The evaluation experiments were performed on both synthetic and real world applications (collaborative filtering and social network link prediction) data. All empirical results demonstrate the effectiveness of the proposed robust matrix recovery model.

References

[Abernethy *et al.*, 2009] J. Abernethy, F. Bach, T. Evgeniou, and J. P. Vert. A new approach to collaborative filtering: Operator estimation with spectral regularization. *JMLR*, 10:803–826, 2009.

[Balzano *et al.*, 2010] Laura Balzano, Robert Nowak, and Benjamin Recht. Online identification and tracking of subspaces from highly incomplete information. In *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, pages 704–711. IEEE, 2010.

[Cai *et al.*, 2008] Jian-Feng Cai, Emmanuel J. Candes, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM J. on Optimization*, 20(4):1956–1982, 2008.

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

[Candes and Tao, 2009] Emmanuel J. Candes and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Trans. Inform. Theory*, 56(5):2053–2080, 2009.

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

[Huang *et al.*, 2013] Jin Huang, Feiping Nie, Heng Huang, Yu Lei, and Chris Ding. Social trust prediction using rank-k matrix recovery. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 2647–2653. AAAI Press, 2013.

[Ji and Ye, 2009] S Ji and Y Ye. An accelerated gradient method for trace norm minimization. *ICML*, 2009.

[Keshavan and Oh, 2009] Raghunandan H Keshavan and Sewoong Oh. A gradient descent algorithm on the grassman manifold for matrix completion. *arXiv preprint arXiv:0910.5260*, 2009.

[Leskovec *et al.*, 2009] J. Leskovec, K.J. Lang, A. Dasgupta, and M.W. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123, 2009.

[Lin *et al.*, 2010] Zhouchen Lin, Minming Chen, and Yi Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010.

[Liu *et al.*, 2009] Y-J Liu, D Sun, and K-C Toh. An implementable proximal point algorithmic framework for nuclear norm minimization. *Optimization Online*, 2009.

[Ma *et al.*, 2009] S Ma, D Goldfarb, and L Chen. Fixed point and bregman iterative methods for matrix rank minimization. *Mathematical Programming*, 2009.

[Marshall *et al.*, 2009] Albert W. Marshall, Ingram Olkin, and Barry Arnold. *Inequalities Theory of Majorization and Its Applications (Second Edition)*. Springer Series in Statistics, 2009.

[Massa and Avesani, 2006] Paolo Massa and Paolo Avesani. Trust-aware bootstrapping of recommender systems. In *ECAI Workshop on Recommender Systems*, pages 29–33. Citeseer, 2006.

[Mazumder *et al.*, 2009] Rahul Mazumder, Trevor Hastie, and Rob Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *submitted to JMLR*, 2009.

[Nie *et al.*, 2010] Feiping Nie, Heng Huang, Xiao Cai, and Chris Ding. Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization. In *NIPS*, 2010.

[Nie *et al.*, 2012] Feiping Nie, Heng Huang, and Chris H. Q. Ding. Low-rank matrix recovery via efficient schatten p-norm minimization. In *AAAI*, 2012.

[Nie *et al.*, 2014] Feiping Nie, Jianjun Yuan, and Heng Huang. Optimal mean robust principal component analysis. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pages 1062–1070, 2014.

[Nie *et al.*, 2015] Feiping Nie, Hua Wang, Heng Huang, and Chris Ding. Joint schatten p-norm and ℓ_p-norm robust matrix completion for missing value recovery. *Knowledge and Information Systems*, 42(3):525–544, 2015.

[Nie *et al.*, 2017] Feiping Nie, Xiaoqian Wang, and Heng Huang. Multiclass capped ℓ_p-norm SVM for robust classifications. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 2415–2421, 2017.

[Recht *et al.*, 2010] Benjamin Recht, Maryam Fazel, and Pablo A. Parrilo. Guaranteed minimumrank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.

[Rennie and Srebro, 2005] Jason Rennie and Nathan Srebro. Fast maximum margin matrix factorization for collaborative prediction. *ICML*, 2005.

[Srebro *et al.*, 2004] N Srebro, J Rennie, and T. Jaakkola. Maximummargin matrix factorization. *NIPS*, 17:1329–1336, 2004.

[Sun *et al.*, 2013] Qian Sun, Shuo Xiang, and Jieping Ye. Robust principal component analysis via capped norms. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 311–319, 2013.

[Toh and Yun, 2010] K.C. Toh and S. Yun. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pacific Journal of Optimization*, 6:615–640, 2010.

[Zhang, 2008] Tong Zhang. Multi-stage convex relaxation for learning with sparse regularization. *NIPS*, 2008.

[Zhang, 2010] Tong Zhang. Analysis of multi-stage convex relaxation for sparse regularization. *JMLR*, pages 1081–1107, 2010.