Convex Matrix Completion: A Trace-Ball Optimization Perspective

Guangxiang Zeng* Ping Luo† Enhong Chen‡ Hui Xiong§ Hengshu Zhu¶ Qi Liu‖

Abstract

The problem of Matrix Completion (MC) refers to the process of adding entries for unknown or missing values in a matrix. In this paper, we study the convex matrix completion problem in the form of trace norm bounding. Specifically, we propose a robust solution for this problem based on *trace-ball optimization*, which can creatively change the original trace norm constraint into the problem of low-rank matrix factorization. Therefore, by searching in a ball space defined by the new trace constraint, the rank of new matrix can be self-determined such that the local minimum for matrix factorization is the global minimum for the original matrix completion task. Meanwhile, we define a free parameter γ to control the model complexity of our approach in terms of how well it fits the training data. Particularly, we identify a value of γ_b, which is the minimal value of the trace norm, in a way such that the model can exactly fit the known entries in the matrix. Furthermore, we also empirically reveal an important property of our approach: that is, a variable η^* generated by γ is always stable with the increase of the amount of training data. This can help to speed up the tuning of optimal parameters for large matrices. Finally, extensive experiments on several real-world datasets clearly validate the effectiveness of the proposed approach.

1 Introduction

Matrix Completion (MC) [15] is the process of adding entries for unknown and missing values in a matrix. While MC has been widely used in a number of data mining methods such as collaborative filtering [21, 17, 10, 23], compressed sensing [6] and nonnegative matrix factorization (NMF) [24], the most commonly-used scenario of MC is to find a low rank approximation for the partially observed real-valued matrix [5, 4, 18, 8]. Since minimizing the matrix rank directly is intractable, the trace norm (or nuclear norm) is widely used as a convex relaxation of the matrix rank [6].

In this study, we consider the convex MC problem in the form of *bounding trace norm* [14], which aims at minimizing the error term with the constraint that the trace norm is not bigger than a threshold. Indeed, this is a *Semi-Definite Programming* (SDP) problem, which has been explored by some researchers [14] in the past few years. However, there are still several open questions along this line. First, as we know, traditional SDP methods (e.g., interior point method) are usually second-order methods. The memory complexity is very expensive, Therefore, would it be possible for us to design a first-order memory-efficient approach for solving such SDP-like MC problem? Second, what is the impact of threshold parameters on the learning process? Finally, does the best tuned parameter on small datasets still have good performance on larger datasets?

To this end, in this paper we propose a robust solution for the convex MC problem based on *trace-ball optimization*, which creatively change the original Semi-Definite constraint into the problem of low-rank matrix factorization. Therefore, by searching in a ball space defined by the new trace constraint, the rank of new matrix can be self-determined such that the local minimum solution for matrix factorization can be used for constructing global minimum solution for the original matrix completion task. Meanwhile, we define a free parameter γ to control the model complexity of our approach in terms of how well it fits the training data. Particularly, we identify a value of γ_b, which is the minimal value of the trace norm, such that the model can exactly fit the known entries in the matrix. Furthermore, we also empirically reveal an important property of our approach, i.e., a variable η^* generated by γ is always stable with the increase of training data, which can help to speed up the tuning of optimal parameters for large matrices. Specifically, the contributions of this paper can be summarized as follows.

- First, we propose a novel first-order low-rank approach for solving the convex MC problem in the form of trace norm bounding, which is based on the trace-ball optimization. Meanwhile, we also give both the theoretical and empirical analysis on the correctness of the proposed approach.
- Second, we discuss how the model parameter controls how well the model fits the training data, and also propose an empirical method which helps to save time in parameter tuning for large-scale MC problems.

*University of Science and Technology of China, zgx@mail.ustc.edu.cn
†Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, luop@ict.ac.cn
‡University of Science and Technology of China, corresponding author, cheneh@ustc.edu.cn
§Rutgers University, hxiong@rutgers.edu
¶Baidu Research-Big Data Lab, zhuhengshu@baidu.com
‖University of Science and Technology of China, qiliuql@ustc.edu.cn

- Last, we conduct extensive experiments on several real-world datasets. The experimental results clearly validate the effectiveness of our proposed method compared with traditional baselines.

2 Problem Statement

In this study, we consider the MC problem in the form of *bounding trace norm* [14]:

$$\begin{aligned} \min \quad & F(X) = \sum_{(i,j)\in\Omega} (\widehat{R}_{ij} - R_{ij})^2 \\ s.t. \quad & X = \begin{bmatrix} W_1 & \widehat{R} \\ \widehat{R}' & W_2 \end{bmatrix} \succeq 0 \\ & tr(X) \leq \gamma, \end{aligned} \tag{I}$$

where $tr(X)$ is the trace of X, and $X \succeq 0$ means that X is a positive semi-definite matrix, γ is the only free parameter. Obviously, it is a SDP problem. Solving this problem by traditional SDP method (e.g., interior point method) is memory inefficient, since 1) we have to hold the total matrix X in the memory; 2) most traditional SDP methods are based on interior point method, which is a second order-method. For MC problems, it means that the traditional SDP methods need $O(N^4)$ memory storage, where N is the number of rows (or columns) of X.

Motivated by the recent development in low-rank method for solving SDP problems [16], we propose to address the problem by low-rank factorization which allows the sparse representation of the original incomplete matrix R. Specifically, we cancel the positive semi-define constraint $X \succeq 0$ by letting $X = YY'$, where $Y \in \mathbb{R}^{N\times p}$. Then, we turn the original SDP problem into the following *non-convex* problem:

$$\begin{aligned} \min \quad & F(YY') \\ s.t. \quad & tr(YY') \leq \gamma. \end{aligned} \tag{II}$$

To this end, the high memory complexity problems of traditional SDP methods for MC problems are perfectly addressed. However, there are still two questions left:

1. Can we solve Problem (II) to get the solution of (I)? What is the relationship between these two problems? How is the rank of Y in Problem (II) set?
2. How does γ controls the model complexity in terms of how well it fits the training data? Can we speed up the tuning of the optimal parameter γ^* for large matrices?

For the first question, we will demonstrate the positive answer with both detailed theoretical and empirically analysis in Section 3. For the second question, we will study the properties of the γ parameter empirically and have some interesting findings in Section 4. Interested readers can also directly jump to Section 4 for the detailed discussion on γ as there is no difficulty without the understanding of Section 3.

3 The Solution to Problem (I)

In this section, we first analyze the relationship between the local minimum of Problem (II) and the global minimum of Problem (I). Specifically, we propose the theoretical analysis on the condition when the local minimum of Problem (II) can be used to construct the global minimum of Problem (I). Then, we proposed the *Trace Ball* optimization (TBall for short) method to Problem (II). Finally, we give the example to intuitively show the correctness of our solution. Before detailing the analysis, we give the notations first, some of which are adopted from [16].

Notations. We use $\mathbb{R}$, $\mathbb{R}_+$, $\mathbb{R}^p$, and $\mathbb{R}^{p\times q}$ to denote the space of real numbers, nonnegative real numbers, real p-dimensional column vectors, and real $p \times q$ matrices, respectively. For $A \in \mathbb{R}^{p\times q}$, $A' \in \mathbb{R}^{q\times p}$ means the transpose matrix of A. We use $\|\cdot\|$ to denote the Euclidean norm for vectors. By $\mathbb{S}^p$ we denote the space of real $p \times p$ symmetric matrices, and we define $\mathbb{S}^p_+$ and $\mathbb{S}^p_{++}$ to be the subsets of $\mathbb{S}^p$ consisting of the positive semi-definite and positive definite matrices, respectively. For a matrix $X \in \mathbb{S}^p$, we write $X \succeq 0$ and $X \succ 0$ to indicate that $X \in \mathbb{S}^p_+$ and $X \in \mathbb{S}^p_{++}$, respectively. We let $tr(X)$ denote the trace of a matrix $X \in \mathbb{R}^{n\times n}$, i.e. the sum of the diagonal elements of X. For $A, B \in \mathbb{R}^{p\times q}$, we define $A \circ B \equiv tr(A'B)$. It is obvious that $tr(X) = I \circ X$, where I is the identity matrix.

For a differentiable function $F(X)$, the notation $\nabla_X F(X_0)$ refers to the gradient of $F(X)$ at X_0 with respect to the variable X. Finally, $||R||_*$ denotes the nuclear norm of R, which is also called the trace norm of R.

3.1 From local optimum to global optimum.

Motivated by the results in [3, 1, 16], we have the following properties that embody the relationship of Problem (I) and Problem (II).

Next, we will propose Lemmas 3.1 and 3.2 and Theorems 3.1 and 3.2. These properties guarantee that we can get the global minimum of Problem (I) by achieving the local minimum of Problem (II).

LEMMA 3.1. *Provided that $F(X)$ is a convex function of X. X is a optimal solution of Problem (I) if and only if there exists a $\sigma \in \mathbb{R}_+$ and a symmetric matrix $S \in \mathbb{S}^N$ such that the following conclusions hold:*

$$\begin{aligned} & 1) && tr(X) && \leq && \gamma, \\ & 2) && X && \succeq && 0, \\ & 3) && \sigma(tr(X) - \gamma) && = && 0, \\ & 4) && S && \succeq && 0, \\ & 5) && SX && = && 0, \end{aligned} \tag{3.1}$$

where $S = \nabla_X F(X) + \sigma I$.

The proof of the above lemma is a little complicated, we provide the proof in the Appendix as it does not related to the main idea of this work, which just verifies the KKT condition to Problem (I).

LEMMA 3.2. *If Y is a local optimum of Problem (II), then there exists an $\alpha \in \mathbb{R}_+$ such that*

$$\begin{array}{lrcl} 1) & tr(YY') & \leq & \gamma, \\ 2) & \alpha(tr(YY') - \gamma) & = & 0, \\ 3) & S_Y Y & = & 0. \end{array}$$

where $S_Y = \nabla_X F(YY') + \alpha I$

Proof. These are the first-order KKT conditions of Problem (II) (see [2]).

With the above two lemmas, we also have the following two theorems, which are similar to those in [3, 1, 16]. The proof of these two theorems is omitted due to the space limitation. Interested readers could refer to the three citation papers.

THEOREM 3.1. *Provided that $F(X)$ is a convex function of X. An Y which satisfies the conclusions of Lemma 3.2 provides a global minimum point YY' of Problem (I) if the matrix*

$$S_Y = \nabla_X F(YY') + \alpha I \tag{3.2}$$

is positive semi-definite, where $\alpha = -\frac{\nabla_X F(YY') \circ YY'}{tr(YY')}$.

THEOREM 3.2. *In the case $p = N$, any local minimum $Y \in \mathbb{R}^{N \times N}$ of Problem (II) provides the global minimum $X = YY'$ of Problem (I).*

Now, the above two theorems actually provide us a method to solve Problem (I) via solving Problem (II). Algorithm 1 summarizes this process. First, we solve Problem (II) with the setting that the rank of Y_p is set to p ($p = 1$ in the first round). Then, by Theorem 3.1, if $S_{Y_p} \succeq 0$ then we get a global minimum $Y_p Y_p'$ of Problem (I). If not, updating p to $(p + 1)$ we repeat the same process again. To verify if a matrix is positive semi-definite, we can only check whether all its eigenvalues are non-negative.

Specifically, in the new round the initial value of Y_{p+1} is set to $[Y_p|0]$ (adding a column of zero vector to Y_p)), which is a saddle point of the feasible set in the space $\mathbb{R}^{N \times (p+1)}$ of Problem (II). At this time, the eigenvector v_{min} of the minimal eigenvalue ρ_{min} of S_{Y_p} actually provides a decent direction $Z_{p+1} = [0|v_{min}]$ ($[0|v_{min}]$ means adding the column vector v_{min} to a $N \times p$ zero matrix) to search for a better solution. Finally, Theorem 3.2 guarantees that we can always find the global minimum if the p is increased one by one until large enough.

In Algorithm 1, Line 6 checks whether S_{Y_p} is positive semi-definite. If $\rho_{min} \geq -\varepsilon$, S_{Y_p} can be viewed as positive semi-definite. Here, ρ_{min} is the minimal eigenvalue of S_{Y_p}, ε is a small positive value to control the accuracy of the algorithm which is set to 10^{-5} in this study.

Algorithm 1 Solution to Problem (I).

Input:
Accuracy level, ε;
Incomplete matrix, R;
Trace constraint, γ;
Output: X;

1: $p = 1$, $stop = 0$;
2: random initial $Y_p \in \mathbb{R}^{N \times p}$ s.t. $tr(Y_p Y_p') \leq \gamma$;
3: **while** $stop \neq 1$ **do**
4: Find a solution Y_p of Problem (II) by exploiting a decent direction Z_p if available, which satisfies the conclusions of Lemma 3.2;
5: Find the smallest eigenvalue ρ_{min} and the corresponding eigenvector v_{min} of matrix S_{Y_p};
6: **if** $\rho_{min} \geq -\varepsilon$ **then**
7: $stop = 1$;
8: **else**
9: $p = p + 1$;
10: $Y_p = [Y_{p-1}|0]$;
11: A decent direction from the saddle point Y_p is given by $Z_p = [0|v_{min}]$.
12: **end if**
13: **end while**
14: Return $X = Y_p Y_p'$;

3.2 Trace Ball Optimization. Next, we propose the solution to Problem (II), whose output Y satisfies the conclusions of Lemma 3.2. The only constraint in Problem (II) is $tr(YY') \leq \gamma$, which makes the feasible set of this problem a ball sphere and its inside with the radius $\sqrt{\gamma}$ in space $\mathbb{R}^{N \times p}$. This is why we call our method to Problem (II) *Trace Ball* Optimization. Let $g(Y) = F(YY')$, the method is summarized in Algorithm 2.

When $\nabla g(Y_t) = 0$, the conclusions of Lemma 3.2 hold with $\alpha = 0$. If $\nabla g(Y_t) \neq 0$, we search the solution in the feasible set of $tr(YY') \leq \gamma$. Generally, we use the gradient descent method, in each round of which we find a proper direction and a step $\tau > 0$ to decrease the objective function. As shown in Figure 1 we have the following 4 cases:

- Case 1: $tr(Y_t Y_t') < \gamma$, Y_t is inside the feasible set. Then, the problem in Equation (3.3) has the solution $\tau > 0$ as shown in Figure 1(a).

$$\begin{array}{rl} \tau = \arg\min\limits_{\tau} & g(Y_t - \tau \cdot \nabla g(Y_t)) \\ s.t. & ||Y_t - \tau \cdot \nabla g(Y_t)||^2 \leq \gamma \end{array} \tag{3.3}$$

- Case 2: $tr(Y_t Y_t') = \gamma$ and $tr(Y_t \nabla g(Y_t)') > 0$. It means Y_t is on the sphere and the opposite of $\nabla g(Y_t)$ points into the feasible set. Then, the problem in Equation (3.3) has the solution $\tau > 0$ as shown in Figure 1(b).
- Case 3: $tr(Y_t Y_t') = \gamma$ and $\exists \alpha > 0$ s.t. $\nabla g(Y_t) = -2\alpha Y_t$. Then, the conclusions of Lemma 3.2 hold as shown in Figure 1(c).

- Case 4: Y_t is on the sphere and the opposite of $\nabla g(Y_t)$ points outside the feasible set. By solving the problem in Equation (3.4) we can still find the decent vector. Here, we normalize the size of Y_{t+1} such that it is on the sphere and also decreases the objective function. As shown in Figure 1(d) the red line shows the decent vector we get.

$$\tau = \arg\min_{\tau} g(\sqrt{\gamma} \cdot \frac{Y_t - \tau \cdot \nabla g(Y_t)}{||Y_t - \tau \cdot \nabla g(Y_t)||}) \quad (3.4)$$

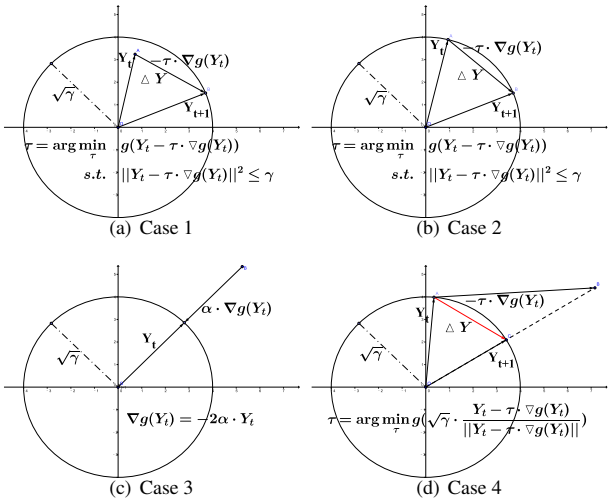

Figure 1: Sphere decent vector searching.

Algorithm 2 Solution to Problem (II).

Input:
Initial $Y_0 \in \mathbb{R}^{N \times p}$ s.t. $tr(Y_0 Y_0') \leq \gamma$;
Incomplete matrix, R;
Trace constraint, γ;

Output: Y, which satisfies the conclusions of Lemma 3.2;

```
t = 0;
while ∇g(Y_t) ≠ 0 do
  if tr(Y_t Y_t') < γ or tr(Y_t ∇g(Y_t)') > 0 then
    Search τ by the problem in Equation (3.3);
    Y_{t+1} = Y_t − τ · ∇g(Y_t);
  else if ∃α > 0 s.t. ∇g(Y_t) = −2αY_t then
    Break;
  else
    Search τ by the problem in Equation (3.4):
    Y_{t+1} = √γ · (Y_t − τ·∇g(Y_t)) / ||Y_t − τ·∇g(Y_t)||;
  end if
  t = t + 1;
end while
Return Y = Y_t;
```

3.3 The running example. First we sample our example dataset from the widely-used *MovieLens* [19] dataset, we randomly sampled 1,000 ratings, which involve with 35 users and 43 movies. Among these 1,000 ratings, we use 800 for training and the remaining 200 for testing. Figure 2 shows the training process by solving Problem (I) with $\gamma = 25$. It shows that as the increase of iterations the loss function $F(YY')$ continuously decreases. Each bar indicates that a local minimum of Problem (II) was found there, and the length of the bar stands for $-\rho_{min}$ to the corresponding S_{Y_p} (ρ_{min} is the minimal eigenvalue of the matrix S_{Y_p}). As we can see, $-\rho_{min}$ decreases up to 0 as the iteration continues. Finally, it will reach or cross 0, which means that S_{Y_p} becomes positive semi-definite. This example empirically shows the correctness of the proposed method.

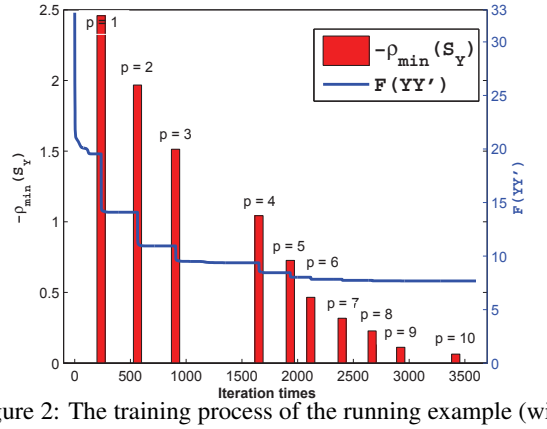

Figure 2: The training process of the running example (with $\gamma = 25.0$). Each bar indicates that a local minimum of Problem (II) was found there, and the length of the bar stands for $-\rho_{min}$ for the corresponding S_{Y_p}.

4 Discussion on the Parameter γ

In this section, we discuss the properties on the parameter γ in Problem (I). Here, γ is used as the upper bound of the trace of the resultant matrix. Intuitively, γ controls the model complexity and is closely related to how well the resultant matrix fits the training data.

Before we detail the effect of γ, we first look at the following optimization problem:

$$\begin{aligned} \min \quad & tr(X) \\ s.t. \quad & \widehat{R}_{ij} = R_{ij}, (i,j) \in \Omega, \\ & X = \begin{bmatrix} W_1 & \widehat{R} \\ \widehat{R}' & W_2 \end{bmatrix} \succeq 0. \end{aligned} \quad (III)$$

This problem actually aims to achieve the minimal value of $tr(X)$ such that the training error equals to 0. In this study this minimal trace value for the target matrix R is denoted by $\gamma_b(R)$ (or γ_b if R is clearly given by the context). For the description convenience, we set $\gamma = \eta \cdot \gamma_b$ ($\eta > 0$) in the following. Next, we will show that the parameter η controls how well the resultant $\widehat{R}$ fits the training data.

4.1 The effect of η. We consider three situations, namely $\eta = 1$, $\eta < 1$, and $\eta > 1$.

4.1.1 When $\eta = 1$. First, when $\eta = 1$ (i.e. $\gamma = \gamma_b$) we have the property:

THEOREM 4.1. *Problem (I) with $\gamma = \gamma_b$ is equivalent to Problem (III). It means the solution of Problem (I) with $\eta = 1$ is also the solution of Problem (III), and vice versa.*

Proof. To prove Theorem 4.1, we assume that X_b^1 is the solution of Problem (III) and X_b^2 is the solution of Problem (I) with $\gamma = \gamma_b$. Thus, we have

$$X_b^1 = \begin{bmatrix} W_1^1 & R_b^1 \\ R_b^{1'} & W_2^1 \end{bmatrix} \succeq 0, X_b^2 = \begin{bmatrix} W_1^2 & R_b^2 \\ R_b^{2'} & W_2^2 \end{bmatrix} \succeq 0 \quad (4.5)$$

Then we have $\gamma_b = tr(X_b^1)$, $tr(X_b^2) \leq \gamma_b$ and $R_b^1(i,j) = R_{ij}$, $(i,j) \in \Omega$, immediately. Here $R_b^1(i,j)$ means the i-th row the j-th column entity of matrix R_b^1.

Obviously, X_b^1 is a solution of Problem (I), because $X_b^1 \succeq 0, tr(X_b) = \gamma_b \leq \gamma_b, F(X) \geq 0$ and $F(X_b^1) = 0$.

Now we prove X_b^2 is also a solution of Problem (III) by contradiction. Firstly, we show that $R_b^2(i,j) = R_{ij}$, $(i,j) \in \Omega$. If not, we have $F(X_b^2) > 0$. From above we know that X_b^1 is also a solution of Problem (I) and we have $F(X_b^1) = 0$. So we have $F(X_b^2) > F(X_b^1)$, which contradicts the fact that X_b^2 is the solution of Problem (I). Secondly, we show that $tr(X_b^2) = \gamma_b$. If not, because $tr(X_b^2) \leq \gamma_b$, we have $tr(X_b^2) < \gamma_b = tr(X_b^1)$. Together with $X_b^2 \succeq 0$ and $R_b^2(i,j) = R_{ij}$, $(i,j) \in \Omega$, it contradicts the fact that X_b^1 is the solution of Problem (III). So we have $X_b^2 \succeq 0$, $R_b^2(i,j) = R_{ij}$, $(i,j) \in \Omega$ and $tr(X_b^2) = \gamma_b$, which means X_b^2 also a solution of Problem (III).

It indicates that when $\eta = 1$ the objective function $F(X)$ in Problem (I) can be minimized to 0.

4.1.2 When $\eta < 1$. Second, when $\eta < 1$ (i.e. $\gamma < \gamma_b$) it is clear that the minimal value of the loss function $F(X)$ in Problem (I) is strictly bigger than 0. It means that there exists the training errors under this situation. The smaller η is, the more strict constraint imposed on the model, and the less degree that the resultant $\widehat{R}$ fits the training data. Thus, η can be used to avoid training over-fitting.

4.1.3 When $\eta > 1$. Third, when $\eta > 1$ (i.e. $\gamma > \gamma_b$) the objective function $F(X)$ in Problem (I) can also be minimized to 0. However, the performance becomes unstable since $\eta > 1$ greatly relaxes the model complexity.

We use the dataset used in Figure 2 to futher demonstrate the effect of η. For different value of η for Problem (I), we calculate the training error and testing error (in terms of Root Mean Square Error, RMSE for short) of the corresponding solution which optimizes Problem (I). Figure 3 shows that as the increase of η the training error becomes smaller monotonously. When $\eta >= 1$, the training error equals to 0, and can not decrease further. As to the testing error, it first decreases as the increase of η. Then, when $\eta \geq 0.3$ the testing error increases. Especially when $\eta > 1$, the testing error increases rapidly and then becomes unstable. Note that in this example the model performs the best around $\eta = 0.3$. In summary, Figure 3 shows that the parameter η acts to control how well the model fits the training data.

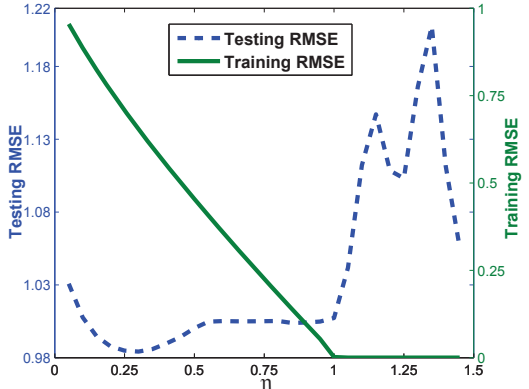

Figure 3: The training error vs. testing error

4.2 The stable η for high performance. Next, we will empirically show that a stable η can always output high performance for different sized incomplete rating matrices datasets from a same data source.

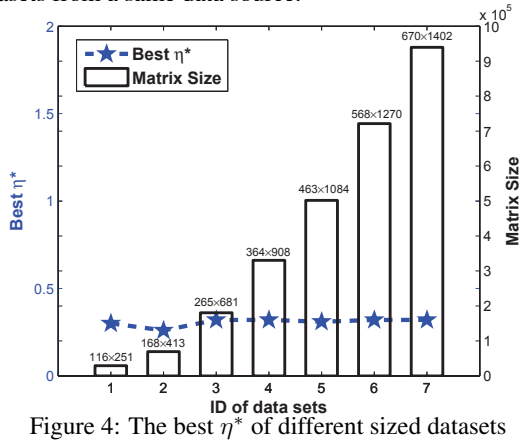

Figure 4: The best η^* of different sized datasets

To show this empirical result, we first randomly sample 7 different-sized datasets, denoted by $R_1, \cdots, R_7$, from the *MovieLens* dataset. For any R_i ($i = 1, \cdots, 7$), we randomly split it into two parts, 80% for training, 20% for test. The sizes of these datasets are shown at the top of the histogram in Figure 4. For each dataset R_i, we tune the optimal $\gamma^*(R_i)$ for Problem (I) to get the best performance on its testing data, and also compute $\gamma_b(R_i)$ in Problem (III). We find that $\gamma^*(R_i)$ changes a lot for different dataset R_i. However, $\eta^* = \frac{\gamma^*(R_i)}{\gamma_b(R_i)}$ remains stable for $i = 1, \cdots, 7$. Figure 4 shows the sizes of the 7 datasets and the corresponding η^*. It is clear that this best η^* is always around 0.3 for all these 7 datasets. The experimental section will further illustrate this observation on different datasets.

In other words, we observe that the same η^* may always output high performance when the size of the training

datasets become bigger and bigger. It indicates that the effects of the same η on the model complexity are similar even for different datasets, in spite that the values of γ_b may change for different datasets.

The identification γ^* for different sized datasets. With this observation we can easily identify γ^* in Problem (I) for the best performance on the different sized datasets from a same data source. Assume that we originally have a small matrix R_1 and a big one R_2. For R_1 we can tune the best $\gamma^*(R_1)$ for Problem (I) and compute $\gamma_b(R_1)$ in Problem (III). When R_1 changes to R_2 with bigger size, we can also compute $\gamma_b(R_2)$. Then, since $\frac{\gamma^*(R_2)}{\gamma_b(R_2)} \approx \frac{\gamma^*(R_1)}{\gamma_b(R_1)}$, we can set $\gamma^*(R_2) = \gamma_b(R_2) \cdot \frac{\gamma^*(R_1)}{\gamma_b(R_1)}$, which may still output high performance on R_2.

Therefore, the proposed method is extremely helpful for parameter tuning for large scale matrix. The value of η^*, which works well on the sampled small dataset, may still perform well even when the size of the dataset become larger. It greatly helps to save much time for parameter tuning for large scale matrix completion.

Solving Problem (III). Clearly, in order to identify η^* we have to get γ_b first by solving Problem (III). For this problem, we use the penalty method [9] which turns the optimization problem with constraint to the lagrange function form without constraint and then use the low-rank method in [1] to solve it. The penalty method ensure that we can get an approximate solution of the original problem within a tolerance (the difference between $\widehat{R}$ and R). The tolerance level is set to 10^{-5} in this study.

5 Experiment

In this section, we evaluate our method on different data collections. We will show that compared with previous trace norm regularization methods [18, 5], the best model parameter of our method is much more stable as the datasets evolve from smaller ones to bigger ones.

5.1 The baseline methods for comparison. We choose the *Trace Regularization* (TReg for short) method [18, 5] in the form of Problem (IV) as the baseline method for comparison.

$$\min \sum_{(i,j)\in\Omega} (\widehat{R}_{ij} - R_{i,j})^2 + \lambda||\widehat{R}||_*. \qquad (IV)$$

Generally speaking, since the left term and right term in Problem (IV) do not change in the same magnitude, the parameter λ may vary a lot for the best performances on different datasets. To alleviate this issue we can also normalize the left term by $|\Omega|$ as follows:

$$\min \frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} (\widehat{R}_{ij} - R_{i,j})^2 + \mu||\widehat{R}||_*. \qquad (V)$$

However, we still cannot totally address it since the rank of $\widehat{R}$ cannot be estimated. It is also clear that Problem (V) is equivalent to Problem (IV) with $\mu = \lambda/|\Omega|$.

For our method in Problem (I), we set $\gamma = \eta^* \cdot \gamma_b$. In this way we actually use γ_b (from Problem (III)) to normalize the trace norm, which helps to eliminate the inconsistency from different-sized data sets (from the same source). Thus, the same η^* may always work well when the size of dataset changes. Our experiments will further validate this.

Table 1: Basic statistics of data sets.

	id	m	n	$\|\Omega\|$	$\|\Omega\|/(m \times n)$
MovieLens	M_1	116	251	7,699	0.2644
	M_2	168	413	14,485	0.2088
	M_3	265	681	28,496	0.1579
	M_4	364	908	43,389	0.1313
	M_5	463	1,084	58,550	0.1167
	M_6	568	1,270	75,813	0.1051
	M_7	670	1,402	91,727	0.0977
Jester	J_1	223	120	16,984	0.6347
	J_2	438	140	33,798	0.5512
	J_3	867	140	66,693	0.5495
	J_4	1,283	140	98,205	0.5467
	J_5	1,566	140	116,735	0.5325
	J_6	1,804	140	133,390	0.5282

5.2 Data preparation and experiment summary. We randomly sampled two groups of different sized datasets from two data sources, namely *MovieLens* [19] and *Jester* [11]. We randomly splitted each dataset into two parts, 80% for training, 20% for test. The basic statistics of the two group of datasets are shown in Table 1. In the following we will empirically show that:

- For the different sized datasets, the best parameter for our method is much more stable than those for the compared methods.
- If we use the best parameter tuned at dataset whose size is the smallest, the performance of our method is significantly better than others when the size of dataset becomes larger.

5.3 Tuning the best parameter. For each data set, we train the model on the training data, and find the best model parameter based on the performance on the test data. For our model, the model parameter η changes from 0.0 to 2.0 by the step of 0.01. For the TReg method we find that its performance becomes worse when $\lambda > 10.0$. Thus, we let λ change from 1.0 to 10.0 by the step of 0.1. Tables 2 and 3 show the best model parameter and the corresponding performance in terms of RMSE on the evolving datasets from the two data sources, respectively.

Table 2 shows that as the datasets changed from M_1 to M_7 the best parameter η^* of our TBall method is always around 3.1. Its relative change rate (in terms of the ratio between the standard deviation and the average) is 0.0668. However, the best parameter λ^* changes from 2.9 to 5.4 and the relative change rate is 0.1768. Thus, the best model parameter of the TBall method is much more stable than that of the TReg method. From Table 2 we can also see that the

best performance of the TBall method in terms of RMSE is comparable with that of the TReg method. The average of TReg is slightly better than that of TBall. Table 3 shows similar results on the data collection of *Jester*.

Table 2: Tuning the best parameter on *MovieLens*

	TBall		TReg		
data set	η^*	RMSE	λ^*	$\lambda^*/\|\Omega\|$	RMSE
M_1	0.30	0.9641	2.9	4.6782E-04	0.9601
M_2	0.26	0.9267	3.8	3.2624E-04	0.9229
M_3	0.32	0.9293	4.2	1.8325E-04	0.9255
M_4	0.32	0.9289	4.6	1.3200E-04	0.9209
M_5	0.31	0.9102	4.8	1.0235E-04	0.9000
M_6	0.32	0.9089	5.0	8.2319E-05	0.9008
M_7	0.32	0.9120	5.4	7.3394E-05	0.9060
ave	0.31	0.9257	4.4	1.9534E-04	0.9194
std	0.0205	0.0182	0.7754	1.3723E-04	0.0193
std/ave	**0.0668**	0.0197	0.1768	0.7025	0.0210

Table 3: Tuning the best parameter on *Jester*

	TBall		TReg		
data set	η^*	RMSE	λ^*	$\lambda^*/\|\Omega\|$	RMSE
J_1	0.25	4.3273	4.5	3.3040E-04	4.2939
J_2	0.24	4.4100	5.5	2.0405E-04	4.3843
J_3	0.26	4.3654	6.8	1.2687E-04	4.3413
J_4	0.24	4.3408	7.4	9.3937E-05	4.3120
J_5	0.26	4.3168	7.7	8.2548E-05	4.2921
J_6	0.26	4.3305	8.6	8.0585E-05	4.3110
ave	0.25	4.3485	6.8	1.5306E-04	4.3225
std	0.0090	0.0314	1.3769	8.9865E-05	0.0320
std/ave	**0.0357**	0.0072	0.2040	0.5871	0.0074

5.4 A fixed model parameter for different sized datasets. In the following we show how the models perform if the parameters are set to the ones which output the best performance on the smallest data set. As shown in Table 2, on M_1 the best parameters for the three models are $\eta = 0.30$, $\lambda = 2.9$ and $\lambda/|\Omega| = 4.6782E-04$. Then, fixing these parameters we test the performances of the three models on the datasets changed from M_2 to M_7 by 5-fold cross validation. Table 4 shows these performances in terms of RMSE. It is clear that under this situation the method of TBall is significantly better than the other two models. It indicates that the model parameter tuned on the initial data set may still work well when the sizes of training datasets become larger. As shown in Table 5 we can see the similar results on the data collection of *Jester*.

Table 4: RMSE of different sized datasets on *MovieLens*

	TBall	TReg	
data set	$\eta = 0.30$	$\lambda = 2.9$	$\lambda/\|\Omega\| = 4.6782E-04$
M_2	**0.9303±0.0053**	0.9499 ± 0.0039	0.9361 ± 0.0047
M_3	**0.9279±0.0025**	0.9591 ± 0.0038	0.9778 ± 0.0028
M_4	**0.9222±0.0035**	0.9632 ± 0.0043	1.0141 ± 0.0029
M_5	**0.9144±0.0028**	0.9605 ± 0.0050	1.0401 ± 0.0025
M_6	**0.9118±0.0033**	0.9658 ± 0.0038	1.0635 ± 0.0029
M_7	**0.9095±0.0034**	0.9679 ± 0.0052	1.0815 ± 0.0028
avg	**0.9193±0.0074**	0.9611 ± 0.0058	1.0189 ± 0.0499

Table 5: RMSE of different sized datasets on *Jester*

	TBall	TReg	
data set	$\eta = 0.25$	$\lambda = 4.5$	$\lambda/\|\Omega\| = 3.3040E-04$
J_2	**4.3939±0.0171**	4.4137 ± 0.0206	4.4917 ± 0.0180
J_3	**4.3515±0.0070**	4.5022 ± 0.0122	4.6101 ± 0.0099
J_4	**4.3434±0.0091**	4.5803 ± 0.0190	4.7092 ± 0.0033
J_5	**4.3285±0.0180**	4.5929 ± 0.0137	4.7484 ± 0.0138
J_6	**4.3124±0.0136**	4.6082 ± 0.0173	4.7979 ± 0.0139
avg	**4.3459±0.0274**	4.5394 ± 0.0727	4.6715 ± 0.1090

6 Related Work and Discussion

Matrix completion technique has been widely used in a number of data mining methods such as collaborative filtering [21, 17], compressed sensing [6] and nonnegative matrix factorization (NMF) [24]. Approximating a target matrix $R \in \mathbb{R}^{m \times n}$ with missing values by the convex method has been studied widely. Candes and Plan [5] give out a usual formulation of this problem:

$$\begin{array}{ll} \min & ||X||_* \\ s.t. & \sum\limits_{(i,j)\in\Omega} (X_{ij} - R_{ij})^2 \leq \delta \end{array} \qquad (VI)$$

where $\delta \geq 0$. If $\delta = 0$, this problem is equal to Problem (III). Candès and Tao [7] proved that when an incomplete matrix R has certain properties (e.g., *strong incoherence property* and $|\Omega| \geq C(m+n)^{6/5} r \log(m+n)$, where r is the rank of R), solving Problem (III) can recover R with high probability. Similar results are also provided in [6]. In [21] the authors proposed maximum-margin matrix factorization. It minimizes the trade-off between the trace norm of X and its hinge-loss relative to the known value. In [4] the authors proposed the singular value thresholding (SVT) algorithm for Problem (III). Jaggi et al. [14] solved Problem (II) by turning the bounded trace constraint into equality trace constraint, and then directly applied the Hazan's method [12]. Wang et al. [22] extended the orthogonal matching pursuit procedure [20] from the vector case to the matrix case and apply it to the matrix completion problem.

All of the above previous works focus on improving MC algorithms on the perspectives of effectiveness and efficiency, few of them pay attention to the problem that the best tuned parameter on the sampled small dataset may not perform well on the larger datasets from the same data source. In this paper, we proposed an empirical studied on this problem, and found that the ratio of the best tuned γ^* (for Problem (II)) and γ_b (the solution of Problem (III)) is almost invariant when the size of the training datasets (from the same source) varies.

The solution to Problem (I) is mainly inspired by [16]. It provides the method for solving optimization problems defined on the cone of positive semi-definite matrices with the equality constraints. Their problem is different to ours in the sense that they use the equality constraints while we has the inequality constraints. It is not a trivial task to extend

the framework to handle the inequality constraints, and we also give the theoretical proof and empirical study on the correctness of the proposed method.

7 Conclusion

In this paper, we proposed a first-order efficient solution for convex matrix completion problem based on trace-ball optimization. Specifically, we first proposed to change the original trace norm constraint into the problem of low-rank matrix factorization. Therefore, by searching in a ball space defined by the new trace constraint, the rank of new matrix can be self-determined such that the local minimum for matrix factorization is the global minimum for the original matrix completion task. Meanwhile, we explored the free parameter γ to control the model complexity of our approach in terms of how well it fits the training data. Moreover, we also empirically revealed that a variable η^* generated by γ is always stable with the increase of training data, which can help to speed up the tuning of optimal parameters for large matrices. Finally, we conducted extensive experiments on two real-world datasets. Experimental results clearly validate the effectiveness of the proposed approach.

Acknowledgements

This research was partially supported by grants from the National Science Foundation for Distinguished Young Scholars of China (Grant No. 61325010), the Natural Science Foundation of China (Grant No. 61403358), the Fundamental Research Funds for the Central Universities of China (Grant No. WK0110000042) and the Anhui Provincial Natural Science Foundation (Grant No. 1408085QF110). Ping Luo is supported by the National Natural Science Foundation of China (No. 61473274) and National 863 Program (No.2014AA015105).

References

[1] FRANCIS BACH, JULIEN MAIRAL, AND JEAN PONCE, *Convex sparse matrix factorizations*, arXiv preprint arXiv:0812.1869, (2008).

[2] STEPHEN POYTHRESS BOYD AND LIEVEN VANDENBERGHE, *Convex optimization*, Cambridge university press, 2004.

[3] SAMUEL BURER AND RENATO DC MONTEIRO, *Local minima and convergence in low-rank semidefinite programming*, Mathematical Programming, 103 (2005), pp. 427–444.

[4] JIAN-FENG CAI, EMMANUEL J CANDÈS, AND ZUOWEI SHEN, *A singular value thresholding algorithm for matrix completion*, SIAM Journal on Optimization, 20 (2010), pp. 1956–1982.

[5] EMMANUEL J CANDES AND YANIV PLAN, *Matrix completion with noise*, Proceedings of the IEEE, 98 (2010), pp. 925–936.

[6] EMMANUEL J CANDÈS AND BENJAMIN RECHT, *Exact matrix completion via convex optimization*, Foundations of Computational mathematics, 9 (2009), pp. 717–772.

[7] EMMANUEL J CANDÈS AND TERENCE TAO, *The power of convex relaxation: Near-optimal matrix completion*, IEEE Transactions on Information Theory, 56 (2010), pp. 2053–2080.

[8] MIRO DUDIK, ZAID HARCHAOUI, JÉRÔME MALICK, ET AL., *Lifted coordinate descent for learning with trace-norm regularization*, in AISTATS-Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics-2012, vol. 22, 2012, pp. 327–336.

[9] ROBERT M FREUND, *Penalty and barrier methods for constrained optimization*, Massachusetts, Institute of Technology, (2004).

[10] YONG GE, QI LIU, HUI XIONG, ALEXANDER TUZHILIN, AND JIAN CHEN, *Cost-aware travel tour recommendation*, in Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2011, pp. 983–991.

[11] KEN GOLDBERG, THERESA ROEDER, DHRUV GUPTA, AND CHRIS PERKINS, *Eigentaste: A constant time collaborative filtering algorithm*, Information Retrieval, 4 (2001), pp. 133–151.

[12] ELAD HAZAN, *Sparse approximate solutions to semidefinite programs*, in LATIN 2008: Theoretical Informatics, Springer, 2008, pp. 306–316.

[13] ROGER A HORN AND CHARLES R JOHNSON, *Matrix analysis*, Cambridge university press, 2012.

[14] MARTIN JAGGI, MAREK SULOVSK, ET AL., *A simple algorithm for nuclear norm regularized problems*, in Proceedings of the 27th International Conference on Machine Learning (ICML-10), 2010, pp. 471–478.

[15] CHARLES R JOHNSON, *Matrix completion problems: a survey*, in Proceedings of Symposia in Applied Mathematics, vol. 40, 1990, pp. 171–198.

[16] MICHEL JOURNÉE, F BACH, P-A ABSIL, AND RODOLPHE SEPULCHRE, *Low-rank optimization on the cone of positive semidefinite matrices*, SIAM Journal on Optimization, 20 (2010), pp. 2327–2351.

[17] YEHUDA KOREN, ROBERT BELL, AND CHRIS VOLINSKY, *Matrix factorization techniques for recommender systems*, Computer, 42 (2009), pp. 30–37.

[18] SHIQIAN MA, DONALD GOLDFARB, AND LIFENG CHEN, *Fixed point and bregman iterative methods for matrix rank minimization*, Mathematical Programming, 128 (2011), pp. 321–353.

[19] BRADLEY N MILLER, ISTVAN ALBERT, SHYONG K LAM, JOSEPH A KONSTAN, AND JOHN RIEDL, *Movielens unplugged: experiences with an occasionally connected recommender system*, in Proceedings of the 8th International Conference on Intelligent User Interfaces, ACM, 2003, pp. 263–266.

[20] YAGYENSH CHANDRA PATI, RAMIN REZAIIFAR, AND PS KRISHNAPRASAD, *Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition*, in Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on, IEEE, 1993, pp. 40–44.

[21] NATHAN SREBRO, JASON RENNIE, AND TOMMI S JAAKKOLA, *Maximum-margin matrix factorization*, in Advances in Neural Information Processing Systems, 2004, pp. 1329–1336.

[22] ZHENG WANG, MING JUN LAI, ZHAOSONG LU, WEI FAN, HASAN DAVULCU, AND JIEPING YE, *Rank-one matrix pursuit for matrix completion*, in Proceedings of the 31st International Conference on Machine Learning (ICML-14), JMLR Workshop and Conference Proceedings, 2014, pp. 91–99.

[23] LE WU, ENHONG CHEN, QI LIU, LINLI XU, TENGFEI BAO, AND LEI ZHANG, *Leveraging tagging for neighborhood-aware probabilistic matrix factorization*, in Proceedings of the 21st ACM international conference on Information and knowledge management, ACM, 2012, pp. 1854–1858.

[24] HENGSHU ZHU, ENHONG CHEN, HUI XIONG, KUIFEI YU, HUANHUAN CAO, AND JILEI TIAN, *Mining mobile user preferences for personalized context-aware recommendation*, ACM Trans. Intell. Syst. Technol., 5 (2014), pp. 58:1–58:27.

Appendix

The proof of Lemma 3.1:

Proof. "⇒": Given X^* and $\sigma \in \mathbb{R}_+$ such that the conclusions (3.1) hold, we prove X^* is the optimal solution of Problem (I).

From 1), 2) we know $X^* \in \{X : tr(X) \leq \gamma, X \succeq 0\}$. For any $X \in \{X : tr(X) \leq \gamma, X \succeq 0\}$, from the convex property of function $F(X)$ we have

(7.6) $F(X) - F(X^*) \geq \nabla_X F(X^*) \circ (X - X^*)$.

From 4) we have

$$\begin{aligned} S \circ X &= tr(SX) \\ &= tr(X^{\frac{1}{2}} S X^{\frac{1}{2}}) \geq 0, \end{aligned} \tag{7.7}$$

where $X^{\frac{1}{2}}$ is the square root of X, which is also a positive semi-definite matrix. Note that there is only one square root of a positive semi-definite matrix X [13].

From 5) we have

(7.8) $S \circ X^* = tr(SX^*) = 0$

With the above properties, we consider the following two circumstances, namely $tr(X^*) < \gamma$ or $tr(X^*) = \gamma$.

When $tr(X^*) < \gamma$**:**

From 3) we know $\sigma = 0$, so we have $\nabla_X F(X^*) = S$.

$$\begin{aligned} \nabla_X F(X^*) \circ (X - X^*) &= S \circ (X - X^*) \\ &= S \circ X - S \circ X^* \\ &\geq 0 \end{aligned} \tag{7.9}$$

When $tr(X^*) = \gamma$**:**

$$\begin{aligned} & \nabla_X F(X^*) \circ (X - X^*) \\ = \; & (S - \sigma I) \circ (X - X^*) \\ = \; & S \circ X + \sigma(\gamma - tr(X)) \geq 0 \end{aligned} \tag{7.10}$$

So in both the circumstances we have $F(X) - F(X^*) \geq 0$ (based on (7.6)), which means X^* is the optimal solution of the Problem (I).

"⇐": Given X^* is the optimal solution of the Problem (I), we prove there exists a $\sigma \in \mathbb{R}_+$ such that the all the conclusions in (3.1) hold.

We know that X^* satisfy both 1) and 2), because they are constraints of Problem (I). We also know $tr(X^*) \geq 0$ for the reason of $X^* \succeq 0$. Thus, we have $0 \leq tr(X^*) \leq \gamma$. Next, we consider the the following three circumstances, namely $tr(X^*) = 0$, $tr(X^*) = \gamma$, or $0 < tr(X^*) < \gamma$.

When $tr(X^*) = 0$:

We set $\sigma = 0$. Then, 3) is satisfied, and $S = \nabla_X F(X^*)$.

(7.11) $tr(X^*) = tr((X^*)^{\frac{1}{2}} \cdot (X^*)^{\frac{1}{2}}) = 0 \Leftrightarrow (X^*)^{\frac{1}{2}} = 0$

Thus, $X^* = (X^*)^{\frac{1}{2}} \cdot (X^*)^{\frac{1}{2}} = 0 \Rightarrow SX^* = 0$, indicating 5) is satisfied.

Next, we prove 4) by contradiction. We assume there exists $y_0 \in \mathbb{R}^N$ that $y_0' \nabla_X F(X^*) y_0 < 0$. Let $y = \frac{\gamma}{||y_0||} y_0$. We consider the following function:

(7.12) $\mathcal{G}(t) = F(X^* + t \cdot yy'), t \in [0, 1]$

Obviously, we have

(7.13) $\{X^* + t \cdot yy' : t \in [0, 1]\} \subseteq \{X : tr(X) \leq \gamma, X \succeq 0\}$

Then, we have

$$\begin{aligned} \frac{\partial \mathcal{G}(t)}{\partial t} \mid_{t=0} &= y' \nabla_X F(X^*) y \\ &= (\frac{\gamma}{||y_0||})^2 y_0' \nabla_X F(X^*) y_0 < 0 \end{aligned} \tag{7.14}$$

(7.12), (7.13) and (7.14) together indicate that we can find another matrix $X^* + t \cdot yy'$ ($t \in [0, 1]$) such that

(7.15) $F(X^* + t \cdot yy') < F(X^*)$

It contradicts with the fact that X^* is the optimal solution of the Problem (I). Thus, for any $y_0 \in \mathbb{R}^N$, we have $y_0' S y_0 = y_0' \nabla_X F(X^*) y_0 \geq 0$, which means $S \succeq 0$.

When $tr(X^*) = \gamma$:

Clearly, 3) holds.

We let $\sigma = -\frac{\nabla_X F(X^*) \circ X^*}{\gamma}$. First, we prove $\sigma \geq 0$ by contradiction. Assume $\sigma < 0$, then $\nabla_X F(X^*) \circ X^* > 0$. Consider the following function:

(7.16) $\mathcal{H}(t) = F((1 - t)X^*), t \in [0, 1]$

Obviously, we have

(7.17) $\{(1 - t)X^* : t \in [0, 1]\} \subseteq \{X : tr(X) \leq \gamma, X \succeq 0\}$

(7.18) $\frac{\partial \mathcal{H}(t)}{\partial t} \mid_{t=0} = -\nabla_X F(X^*) \circ X^* < 0$

Similarly, we can see that the fact that X^* is the optimal solution of Problem (I) contradicts with (7.16), (7.17) and (7.18). Thus, $\sigma \geq 0$.

Next, we prove 4) by contradiction. We assume there exists $y_0 \in \mathbb{R}^N$ such that $y_0' S y_0 < 0$. Let $y = \frac{\gamma}{||y_0||} y_0$. We consider the following function:

(7.19) $\mathcal{G}(t) = F((1 - t)X^* + t \cdot yy'), t \in [0, 1]$

Obviously, we have

(7.20) $\{(1 - t)X^* + t \cdot yy' : t \in [0, 1]\} \subseteq \{X : tr(X) \leq \gamma, X \succeq 0\}$

$$\begin{aligned} \frac{\partial \mathcal{G}(t)}{\partial t} \mid_{t=0} &= S \circ yy' - S \circ X^* \\ &= (\frac{\gamma}{||y_0||})^2 y_0' S y_0 < 0 \end{aligned} \tag{7.21}$$

We can see that the fact that X^* is the optimal solution of the Problem (I) contradicts with (7.19), (7.20) and (7.21). Thus, for any $y_0 \in \mathbb{R}^N$, we have $y_0' S y_0 \geq 0$, which means $S \succeq 0$.

Then, we have

$$\begin{aligned} 0 &= S \circ X^* = tr(SX^*) \\ &= tr((X^*)^{\frac{1}{2}} S (X^*)^{\frac{1}{2}}) \\ &= tr(((X^*)^{\frac{1}{2}} S^{\frac{1}{2}})(S^{\frac{1}{2}} (X^*)^{\frac{1}{2}})) \\ &= tr((S^{\frac{1}{2}} (X^*)^{\frac{1}{2}})'(S^{\frac{1}{2}} (X^*)^{\frac{1}{2}})) \\ \Leftrightarrow \; & S^{\frac{1}{2}} (X^*)^{\frac{1}{2}} = 0 \\ \Rightarrow \; & SX^* = S^{\frac{1}{2}} (S^{\frac{1}{2}} (X^*)^{\frac{1}{2}})(X^*)^{\frac{1}{2}} = 0 \end{aligned} \tag{7.22}$$

Thus, 5) is also proved.

When $0 < tr(X^*) < \gamma$:

We set $\sigma = 0$. Thus, 3) is satisfied. Now, we prove $\nabla_X F(X^*) \succeq 0$ and $\nabla_X F(X^*) X^* = 0$, which corresponds to 4) and 5).

We can prove $\nabla_X F(X^*) \succeq 0$ almost the same as what we do in the situation "$tr(X^*) = 0$", with a slight change that t varies in $[0, 1 - \frac{tr(X)}{\gamma}]$.

Then, we prove $\nabla_X F(X^*) \circ X^* \geq 0$ and $\nabla_X F(X^*) \circ X^* \leq 0$ by contradictions, which we can consider the following two functions respectively.

$$\begin{aligned} & \mathcal{H}_1(t) = F((1 + t)X^*), t \in [0, \tfrac{\gamma}{tr(X)} - 1] \\ & \mathcal{H}_2(t) = F((1 - t)X^*), t \in [0, 1] \end{aligned} \tag{7.23}$$

Now we have $\nabla_X F(X^*) \circ X^* = 0$. Similar to (7.22), we have $\nabla_X F(X^*) X^* = 0$ immediately.

Therefore, the proof is complete.