Box Constrained Low-rank Matrix Approximation with Missing Values

Manami Tatsukawa[1] and Mirai Tanaka[2]

[1] *Department of Industrial Engineering and Economics, Tokyo Institute of Technology,*
2-12-1 Ookayama, Meguro-ku, Tokyo 152-8550, Japan

[2] *Department of Mathematical Analysis and Statistical Inference, The Institute of Statistical Mathematics,*
10-3 Midori-cho, Tachikawa, Tokyo 190-8562, Japan

Keywords: Low-Rank Matrix Approximation, Missing Data, Matrix Completion with Noise, Principal Component Analysis with Missing Values, Collaborative Filtering, Block Coordinate Descent Method.

Abstract: In this paper, we propose a new low-rank matrix approximation model for completing a matrix with missing values. Our proposed model contains a box constraint that arises from the context of collaborative filtering. Although it is unfortunately NP-hard to solve our model with high accuracy, we can construct a practical algorithm to obtain a stationary point. Our proposed algorithm is based on alternating minimization and converges to a stationary point under a mild assumption.

1 INTRODUCTION

1.1 Background

Low-rank matrix approximation is commonly used for feature extraction. This technique embeds high-dimensional data into a lower dimensional space, because relevant data in a high-dimensional space often lie in a lower dimensional space. Feature extraction enables us to identify potential features that may help to increase prediction accuracy when data are incomplete or contain some noise. We provide two examples of techniques for approximating a data matrix with missing values.

1.1.1 PCA with Missing Value

Principal component analysis (PCA) is a classical technique used for extracting features. This technique embeds high-dimensional data into lower dimensional space. The components embedded into lower dimensional space are called principal components. To handle incompleteness of input data we often use nonlinear models; however, these models cause problems such as overfitting and bad locally optimal solutions. Tipping and Bishop (1999) introduced a probabilistic formulation of PCA. The probabilistic PCA is known to provide a good foundation for handling missing values (Ilin and Raiko, 2010). Probabilistic PCA is often solved by an expectation-maximization (EM) algorithm.

1.1.2 Collaborative Filtering

Low-rank matrix approximation is utilized in recommendation systems such as those found on iTunes, Amazon, and Netflix. In these services, music, book, and movie recommendations are provided to users. Users rate items they have listened to, bought, or watched. Based on the ratings, items are recommended based on the user's preferences or the items' novelty to the user.

Let us consider the following situation: There are m users and n items. Every user rates some of items on a scale of one to five. One is the lowest score and it means that a user does not prefer the item. We represent this with an $m \times n$ matrix. Each row of the matrix represents each user and each column represents each item. We show an example of 3×5 matrix below:

$$\boldsymbol{V} = \begin{array}{c} \\ \text{A} \\ \text{B} \\ \text{C} \end{array} \!\!\begin{array}{c} \begin{array}{ccccc} \text{a} & \text{b} & \text{c} & \text{d} & \text{e} \end{array} \\ \begin{pmatrix} 2 & 5 & ? & 4 & 1 \\ 1 & ? & 1 & 3 & 2 \\ ? & 1 & 4 & ? & 5 \end{pmatrix} \end{array}. \tag{1}$$

Matrix $\boldsymbol{V}$ represent that, for instance, user A rates item b five and does not rate item c, using the symbol "?". In what follows, we use the symbol "?" for representing missing values. In the example shown in Equation (1), users A and B seems to have similar trends. This means that an item that user A rates high may also be preferred by user B. For example, item b, which user A rates five, may also be rated high by

Tatsukawa, M. and Tanaka, M.
Box Constrained Low-rank Matrix Approximation with Missing Values.
DOI: 10.5220/0006612100780084
In *Proceedings of the 7th International Conference on Operations Research and Enterprise Systems (ICORES 2018)*, pages 78-84
ISBN: 978-989-758-285-1

user B. If we complete the missing value of the matrix, we can predict how users prefer items that they have not evaluate yet. This technique of predicting is can be used for recommendation systems.

A technique used in recommendation systems is collaborative filtering (CF). CF algorithms have three main categories: memory-based, model-based, and hybrid (Su and Khoshgoftaar, 2009). Memory-based CF algorithms calculate similarities between users or items to predict users' preferences. Model-based CF algorithms learn a model in order to make predictions. Hybrid CF algorithms combine several CF techniques.

In memory-based CF algorithms, similarities between users or items are used to make predictions. As in measures of similarity, the vector cosine correlation and the Pearson correlation are often used. However, when many values are missing, it is difficult to compute similarities between users. In fact, the number of items might be greater than the number of users and each user may evaluate only a small number of items. For this reason, many items are evaluated by only a few users, while other users do not submit any evaluations.

To overcome the weakness in memory-based CF algorithms, model-based CF algorithms have been investigated. Model-based CF approaches use data mining or machine learning algorithms. One of the techniques of model-based CF is dimensionality reduction, such as PCA or singular value decomposition (SVD). As we mentioned above, high-dimensional data is thought to be expressed by lower dimensional data because related data lie in a lower dimensional space.

1.2 Related Work

Low-rank matrix completion and approximation have been studied. In this section, we briefly introduce some papers about these techniques.

Let us consider completing a matrix $\boldsymbol{V} \in (\mathbb{R} \cup \{?\})^{m\times n}$ with missing values, where the symbol ? indicates that the corresponding value is missing. That is, $V_{ij} = ?$ indicates V_{ij} is missing. Candés and Recht (2009) proposed rank minimization to complete matrix $\boldsymbol{V}$. They considered the following problem:

$$\begin{array}{ll} \text{minimize} & \operatorname{rank}(\boldsymbol{X}) \\ \text{subject to} & X_{ij} = V_{ij} \quad ((i,j) \in \Omega), \end{array} \tag{2}$$

where Ω is the set of indices of observed entries, *i.e.*, $\Omega = \{(i,j) : V_{ij} \in \mathbb{R}\}$. Problem (2) is NP-hard because it contains the l_0-norm minimization problem. This difficulty essentially arises from the nonconvexity and discontinuity of the rank function. Hence, they introduced nuclear norm minimization as a convex relaxation of Problem (2). Nuclear norm minimization can be recast as a semidefinite optimization problem (SDP). There are many efficient algorithms and high-quality software packages available for solving SDP, including the interior-point method. However, the computation time for solving SDP is very sensitive to instance size and is unsuitable for solving large instances

Olsson and Oskarsson (2009); Gillis and Glineur (2011) studied the following problem to complete $\boldsymbol{V}$:

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n} W_{ij}(X_{ij} - V_{ij})^2 \\ \text{subject to} & \operatorname{rank}(\boldsymbol{X}) \le r, \end{array} \tag{3}$$

where decision variable $\boldsymbol{X} \in \mathbb{R}^{m\times n}$ is a completed matrix of $\boldsymbol{V}$ and $\boldsymbol{W} \in \{0,1\}^{m\times n}$ is a given weight matrix corresponding to an observation, *i.e.*, $W_{ij} = 1$ for $V_{ij} \in \mathbb{R}$ and otherwise $W_{ij} = 0$. Using Ω, we obtain the following equivalent formulation of Problem (3):

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{(i,j)\in\Omega} (X_{ij} - V_{ij})^2 \\ \text{subject to} & \operatorname{rank}(\boldsymbol{X}) \le r. \end{array} \tag{4}$$

Our formulation is similar to this one and this model is helpful to understand our model. Olsson and Oskarsson (2009) proposed a heuristic based on an approximated continuous (but nonconvex) formulation of Problems (3). Gillis and Glineur (2011) proved the NP-hardness of Problem (3), and equivalently, Problem (4).

On the other hand, the low-rank matrix approximation problem is easily solved when no values are missing. In fact, when Ω is an entire set of indices, Problem (4) is equivalent to the following problem:

$$\begin{array}{ll} \text{minimize} & \|\boldsymbol{X} - \boldsymbol{V}\|_{\mathrm{F}}^2 \\ \text{subject to} & \operatorname{rank}(\boldsymbol{X}) \le r, \end{array}$$

where $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm defined by

$$\|\boldsymbol{A}\|_{\mathrm{F}} = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}^2}$$

for $\boldsymbol{A} \in \mathbb{R}^{m\times n}$. This problem is nonconvex; however, a global optimal solution is obtained by the truncated SVD of $\boldsymbol{V}$. Specifically, it is well known that an optimal solution of this problem can be written as $\sum_{l=1}^{r} \sigma_l \boldsymbol{p}_l \boldsymbol{q}_l^\top$ (Trefethen and Bau, 1997, Theorem 5.9), where σ_l, $\boldsymbol{p}_l$, and $\boldsymbol{q}_l$ respectively represent the l-th largest singular value and corresponding singular vectors of $\boldsymbol{V}$. The computation of all singular values and the vectors of $\boldsymbol{V}$ is expensive. More specifically, it requires a computation time in $O(\min\{m^2n, mn^2\})$, which can be too heavy for a large instance. Instead, the r largest singular values and corresponding singular vectors can be quickly computed by an iterative method if r is small.

When we use the truncated SVD for a matrix containing missing values, we need to complete the matrix. One method for doing this involves completing the input matrix with the average of the non-missing entries in the same rows or columns (Sarwar et al., 2000). Specifically, their method comprises the following steps:

1. Temporarily fill in the missing values of incomplete input matrix $\boldsymbol{V}$ with the average of the non-missing entries in the same columns. We call the completed matrix $\hat{\boldsymbol{V}}$. That is, $\hat{V}_{ij} = (1/|\{i' : (i', j) \in \Omega\}|)\sum_{i':(i',j)\in\Omega} V_{i'j}$ for $(i, j) \notin \Omega$.
2. Compute row average vector $\boldsymbol{\mu}$. The i-th element μ_i of $\boldsymbol{\mu}$ is the average of the i-th row of $\boldsymbol{V}$. That is, $\mu_i = (1/|\{j : (i, j) \in \Omega\}|)\sum_{j:(i,j)\in\Omega} V_{ij}$.
3. Compute the best rank r approximation matrix $\boldsymbol{V}_r$ of $\hat{\boldsymbol{V}} - \boldsymbol{\mu}\boldsymbol{1}^\top$ by using the truncated SVD, where $\boldsymbol{1}$ is a vector of all ones.
4. Return $\boldsymbol{V}_r + \boldsymbol{\mu}\boldsymbol{1}^\top$ as a low-rank approximation of input matrix $\boldsymbol{V}$.

1.3 Our Contribution and Structure of This Paper

The remainder of this paper is organized as follows: Section 2 proposes a new model for low-rank matrix approximation contains a box constraint that arises from the context of CF. The rank minimization models was previously proposed by Candés and Recht (2009) and Olsson and Oskarsson (2009). Their model can be recast as an SDP. There are many software packages available for solving SDP, but SDP is unsuitable for solving large instances. Our model includes Problem (4), which is proved NP-hard by Gillis and Glineur (2011). However, we can solve our model by truncated SVD. Moreover, our model is more suitable for CF than previous models owing to the box constraint. We need not our model recast as an SDP and suitable for large instances. Section 3 proves the NP-hardness of our proposed model. Section 4 proposes an algorithm for solving our proposed model and proves the convergence of the algorithm. Section 5 reports preliminary numerical results to assess our proposed model and algorithm. Section 6 presents concluding remarks.

2 MODEL

In this section, we propose a new model for low-rank matrix approximation that is easier to calculate. Here, matrix $\boldsymbol{X} \in \mathbb{R}^{m\times n}$ denotes a completed matrix of matrix $\boldsymbol{V} \in (\mathbb{R}\cup\{?\})^{m\times n}$ with missing values. We consider minimizing the difference between $\boldsymbol{V}$ and $\boldsymbol{X}$. This idea is described in previous studies.

We consider imposing a box constraint $\boldsymbol{L} \leq \boldsymbol{X} \leq \boldsymbol{U}$, where $\boldsymbol{L} \in (\mathbb{R}\cup\{-\infty\})^{m\times n}$ and $\boldsymbol{U} \in (\mathbb{R}\cup\{+\infty\})^{m\times n}$ satisfy $\boldsymbol{L} \leq \boldsymbol{U}$ and the inequalities are entrywise. For example, $\boldsymbol{L} \leq \boldsymbol{X}$ means $L_{ij} \leq X_{ij}$ for all i, j. The box constraint is important when applying low-rank matrix approximation with missing values to CF, because users evaluate items in some range. For example, an Amazon user evaluates an item by assigning it one to five stars. We model such an evaluation value range as the box constraint. Matrices $\boldsymbol{L}, \boldsymbol{U}$ represent the lower and upper bounds of an evaluation value range, respectively. This constraint is helpful in making exact or approximate predictions and in eliminating outliers.

To prevent a case in which both the rank constraint and the box constraint are not simultaneously fulfilled, we consider variables that fulfil each constraint separately. For this reason, we introduce another variable, $\boldsymbol{Y} \in \mathbb{R}^{m\times n}$, and impose the box constraint on $\boldsymbol{Y}$. We then add the squared Frobenius norm $\|\boldsymbol{X}-\boldsymbol{Y}\|_{\mathrm{F}}^2$ of the difference between $\boldsymbol{X}$ and $\boldsymbol{Y}$ to the objective function as a penalty.

Summarizing the previous argument, we formulate low-rank matrix approximation with missing values as the following optimization problem:

$$\begin{array}{ll} \text{minimize} & \|\boldsymbol{X}-\boldsymbol{Y}\|_{\mathrm{F}}^2 + \lambda \sum\limits_{(i,j)\in\Omega} (Y_{ij} - V_{ij})^2 \\ \text{subject to} & \operatorname{rank}(\boldsymbol{X}) \leq r, \\ & \boldsymbol{L} \leq \boldsymbol{Y} \leq \boldsymbol{U}, \end{array} \tag{5}$$

where λ is a parameter to determine the weight of the two objectives. When we use large λ, the second part in the objective function is emphasized, so that we expect to obtain $\boldsymbol{Y}$ close to $\boldsymbol{V}$ permitting small violation of the low-rank constraint. When we use small λ, the first part in the objective function is emphasized, so that we expect to obtain $\boldsymbol{Y}$ satisfying the two constraints simultaneously permitting small difference from $\boldsymbol{V}$.

3 HARDNESS

In this section, we show the NP-hardness of Problem (5). Specifically, we prove the following result:

Theorem 1. *When $\boldsymbol{V} \in ([0,1] \cup \{?\})^{m\times n}$ and $r = 1$, it is NP-hard to find an approximate solution to Problem* (5) *with an objective function accuracy of less than* $2^{-12}(mn)^{-7}$.

To prove this theorem, we employ the following theorem:

Theorem 2 (Gillis and Glineur (2011, Theorem 1.2))**.** *When $\boldsymbol{V} \in ([0,1] \cup \{?\})^{m\times n}$ and $r = 1$, it is NP-hard to find an approximate solution to Problem* (4) *with an objective function accuracy of less than* $2^{-12}(mn)^{-7}$.

Proof of Theorem 1. In Problem (5), we set $L_{ij} = U_{ij} = V_{ij}$ for $(i,j) \in \Omega$; otherwise, $L_{ij} = -\infty$ and $U_{ij} = +\infty$. Then, for $(i,j) \in \Omega$, Y_{ij} is fixed to V_{ij} and the second part of the objective function is removed. Thus, the resulting objective function can be written as $\sum_{(i,j)\in\Omega}(X_{ij} - V_{ij})^2 + \sum_{(i,j)\notin\Omega}(X_{ij} - Y_{ij})^2$. The latter part of this expression is also removed because Y_{ij} for $(i,j) \notin \Omega$ is unconstrained and is able to coincide with X_{ij}. As a result, we obtain Problem (4). In this procedure, we have reduced Problem (4) to Problem (5). This reduction is clearly in polynomial time. Hence, the hardness for Problem (4) also holds for Problem (5). □

Remark 1. Theorem 1 is easily generalized to any r because Theorem 2 is generalized to any r (Gillis and Glineur, 2011, Remark 3).

4 ALGORITHM

We propose an alternating minimization algorithm for solving Problem (5). In this section, we use the following extended-real-valued functions:

$$\begin{aligned} f_0(\boldsymbol{X},\boldsymbol{Y}) &= \|\boldsymbol{X}-\boldsymbol{Y}\|_{\mathrm{F}}^2 + \lambda \sum_{(i,j)\in\Omega}(Y_{ij} - V_{ij})^2, \\ f_1(\boldsymbol{X}) &= \iota(\mathrm{rank}(\boldsymbol{X}) \le r), \\ f_2(\boldsymbol{Y}) &= \iota(\boldsymbol{L} \le \boldsymbol{Y} \le \boldsymbol{U}), \\ f(\boldsymbol{X},\boldsymbol{Y}) &= f_0(\boldsymbol{X},\boldsymbol{Y}) + f_1(\boldsymbol{X}) + f_2(\boldsymbol{Y}), \end{aligned}$$

where ι is the indicator function. That is, $f_1(\boldsymbol{X}) = 0$ if $\mathrm{rank}(\boldsymbol{X}) \le r$; otherwise $f_1(\boldsymbol{X}) = +\infty$; and $f_2(\boldsymbol{Y}) = 0$ if $\boldsymbol{L} \le \boldsymbol{Y} \le \boldsymbol{U}$; otherwise $f_2(\boldsymbol{Y}) = +\infty$. Note that the minimization of $f(\boldsymbol{X},\boldsymbol{Y})$ is equivalent to Problem (5). We consider the alternating minimization of $f(\boldsymbol{X},\boldsymbol{Y})$ as shown in Algorithm 1. Each iteration of this algorithm is easily computed as we see below.

In the update of $\boldsymbol{X}$, we set $\boldsymbol{X}^{(k+1)}$ to the best rank-r approximation of $\boldsymbol{Y}^{(k)}$. That is, $\boldsymbol{X}^{(k+1)}$ is an optimal solution of the following subproblem:

$$\begin{array}{ll} \text{minimize} & \|\boldsymbol{X} - \boldsymbol{Y}^{(k)}\|_{\mathrm{F}}^2 \\ \text{subject to} & \mathrm{rank}(\boldsymbol{X}) \le r. \end{array}$$

Algorithm 1: Alternating minimization algorithm for solving Problem (5).

```
Take initial guess $(\boldsymbol{X}^{(0)}, \boldsymbol{Y}^{(0)}) \in \mathrm{dom}\, f_1 \times \mathrm{dom}\, f_2$.
for $k = 0, 1, 2, \ldots$ until convergence:
    Update $\boldsymbol{X}^{(k+1)} = \mathrm{argmin}_{\boldsymbol{X}}\, f(\boldsymbol{X}, \boldsymbol{Y}^{(k)})$.
    Update $\boldsymbol{Y}^{(k+1)} = \mathrm{argmin}_{\boldsymbol{Y}}\, f(\boldsymbol{X}^{(k+1)}, \boldsymbol{Y})$.
```

Although this subproblem is a nonconvex optimization problem, the optimal solution is easily computed by the truncated SVD of $\boldsymbol{Y}^{(k)}$.

The update of $\boldsymbol{Y}$ is also easily computed. In fact, we solve the following subproblem to update $\boldsymbol{Y}$:

$$\begin{array}{ll} \text{minimize} & \|\boldsymbol{X}^{(k+1)} - \boldsymbol{Y}\|_{\mathrm{F}}^2 + \lambda \sum_{(i,j)\in\Omega}(Y_{ij} - V_{ij})^2 \\ \text{subject to} & \boldsymbol{L} \le \boldsymbol{Y} \le \boldsymbol{U}. \end{array} \tag{6}$$

Note that this subproblem is separable. The separated subproblem for each entry reduces to the minimization of a univariate convex quadratic function over a closed interval. Specifically, we solve

$$\begin{array}{ll} \text{minimize} & (1+\lambda)Y_{ij}^2 - 2(X_{ij}^{(k+1)} + \lambda V_{ij})Y_{ij} \\ \text{subject to} & L_{ij} \le Y_{ij} \le U_{ij} \end{array}$$

for each $(i,j) \in \Omega$ and

$$\begin{array}{ll} \text{minimize} & Y_{ij}^2 - 2X_{ij}^{(k+1)}Y_{ij} \\ \text{subject to} & L_{ij} \le Y_{ij} \le U_{ij} \end{array}$$

for each $(i,j) \notin \Omega$. These have the following closed-form solution:

$$Y_{ij}^{(k+1)} = \begin{cases} L_{ij} & (A_{ij} \le L_{ij}), \\ A_{ij} & (L_{ij} < A_{ij} < U_{ij}), \\ U_{ij} & (U_{ij} \le A_{ij}), \end{cases}$$

where

$$A_{ij} = \begin{cases} \dfrac{X_{ij}^{(k+1)} + \lambda V_{ij}}{1+\lambda} & ((i,j) \in \Omega), \\ X_{ij}^{(k+1)} & ((i,j) \notin \Omega). \end{cases}$$

Hence, we can solve Subproblem (6) by thresholding, which requires a computation time in $O(mn)$.

A sequence generated by Algorithm 1 converges to a stationary point of f under a mild assumption. Here, we say point $(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}})$ is a stationary point of f in the sense of Tseng (2001) if $(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}}) \in \mathrm{dom}\, f = \{(\boldsymbol{X},\boldsymbol{Y}) : f(\boldsymbol{X},\boldsymbol{Y}) < +\infty\}$ and

$$f'(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}}; \Delta\boldsymbol{X}, \Delta\boldsymbol{Y}) \ge 0 \ (\forall(\Delta\boldsymbol{X}, \Delta\boldsymbol{Y})),$$

where $f'(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}}; \Delta\boldsymbol{X}, \Delta\boldsymbol{Y})$ is the lower directional derivative of f at $(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}})$ in the direction $(\Delta\boldsymbol{X}, \Delta\boldsymbol{Y})$, *i.e.*,

$$f'(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}}; \Delta\boldsymbol{X}, \Delta\boldsymbol{Y})$$

$$= \liminf_{\varepsilon \downarrow 0} \frac{f(\bar{\boldsymbol{X}} + \varepsilon \Delta \boldsymbol{X}, \bar{\boldsymbol{Y}} + \varepsilon \Delta \boldsymbol{Y}) - f(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}})}{\varepsilon}.$$

Note that this definition works even if f is nonsmooth. Specifically, the following theorem holds.

Theorem 3. *Let $\{(\boldsymbol{X}^{(k)}, \boldsymbol{Y}^{(k)})\}$ be a sequence generated by Algorithm 1 and assume that level set $L = \{(\boldsymbol{X}, \boldsymbol{Y}) : f(\boldsymbol{X}, \boldsymbol{Y}) \le f(\boldsymbol{X}^{(0)}, \boldsymbol{Y}^{(0)})\}$ is bounded. Then, $\{(\boldsymbol{X}^{(k)}, \boldsymbol{Y}^{(k)})\}$ has at least one cluster point. In addition, every cluster point is a stationary point of f.*

To prove this theorem, the following technical lemma is required.

Lemma 1. *Effective domain* $\operatorname{dom} f$ *of f is closed.*

Proof. Clearly, $\operatorname{dom} f = \operatorname{dom} f_1 \times \operatorname{dom} f_2$ and $\operatorname{dom} f_2$ is closed. Thus, we only need to show the closedness of $\operatorname{dom} f_1$. Take $\boldsymbol{X} \notin \operatorname{dom} f_1$ arbitrarily. Then, the r-th largest singular value $\sigma_r(\boldsymbol{X})$ of $\boldsymbol{X}$ is positive. We arbitrarily take $\boldsymbol{E}$ such that $\|\boldsymbol{E}\|_2 < \sigma_r(\boldsymbol{X})$. From Golub and van Loan (2013, Corollary 8.6.2), we can easily prove that $\sigma_l(\boldsymbol{X} + \boldsymbol{E}) \ge \sigma_l(\boldsymbol{X}) - \|\boldsymbol{E}\|_2 > 0$ for $l = 1, \dots, r$, so that $\boldsymbol{X} + \boldsymbol{E} \notin \operatorname{dom} f_1$. This indicates the closedness of $\operatorname{dom} f_1$. □

Here, we provide a proof of Theorem 3. The following proof is essentially based on the discussion in Tseng (2001, Sections 3 and 4).

Proof of Theorem 3. From the optimality in each update, sequence $\{(\boldsymbol{X}^{(k)}, \boldsymbol{Y}^{(k)})\}$ is contained in $L \subset \operatorname{dom} f$, so that $\{(\boldsymbol{X}^{(k)}, \boldsymbol{Y}^{(k)})\}$ is bounded. Hence, $\{(\boldsymbol{X}^{(k)}, \boldsymbol{Y}^{(k)})\}$ has at least one cluster point on $\operatorname{dom} f$ because $\operatorname{dom} f$ is closed from Lemma 1.

Let $\{(\boldsymbol{X}^{(k_j)}, \boldsymbol{Y}^{(k_j)})\}$ be a subsequence of $\{(\boldsymbol{X}^{(k)}, \boldsymbol{Y}^{(k)})\}$ that converges to a cluster point $(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}})$. To show that $(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}})$ is a stationary point, we only have to prove $f'(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}}; \Delta \boldsymbol{X}, \Delta \boldsymbol{Y}) \ge 0$ for any $(\Delta \boldsymbol{X}, \Delta \boldsymbol{Y})$ because $(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}}) \in \operatorname{dom} f$. From the optimality in each update, we obtain

$$\begin{aligned} f(\boldsymbol{X}^{(k_{j+1})}, \boldsymbol{Y}^{(k_{j+1})}) &\le f(\boldsymbol{X}^{(k_j+1)}, \boldsymbol{Y}^{(k_j)}) \\ &\le f(\boldsymbol{X}, \boldsymbol{Y}^{(k_j)}) \ (\forall \boldsymbol{X}). \end{aligned}$$

Taking the limit as j tends to infinity, we obtain

$$f(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}}) \le f(\boldsymbol{X}, \bar{\boldsymbol{Y}}) \ (\forall \boldsymbol{X}) \tag{7}$$

because f is continuous on $\operatorname{dom} f$. In addition,

$$f(\boldsymbol{X}^{(k_j)}, \boldsymbol{Y}^{(k_j)}) \le f(\boldsymbol{X}^{(k_j)}, \boldsymbol{Y}) \ (\forall \boldsymbol{Y})$$

holds. Taking the limit as j tends to infinity, we obtain

$$f(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}}) \le f(\bar{\boldsymbol{X}}, \boldsymbol{Y}) \ (\forall \boldsymbol{Y}). \tag{8}$$

Because f_0 is differentiable, the following relationship holds for any $(\Delta \boldsymbol{X}, \Delta \boldsymbol{Y})$:

$$\begin{aligned} & f'(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}}; \Delta \boldsymbol{X}, \Delta \boldsymbol{Y}) \\ &= \langle \nabla f_0(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}}), (\Delta \boldsymbol{X}, \Delta \boldsymbol{Y}) \rangle \\ &\quad + \liminf_{\varepsilon \downarrow 0} \left(\frac{f_1(\bar{\boldsymbol{X}} + \varepsilon \Delta \boldsymbol{X}) - f_1(\bar{\boldsymbol{X}})}{\varepsilon} \right. \\ &\qquad \left. + \frac{f_2(\bar{\boldsymbol{Y}} + \varepsilon \Delta \boldsymbol{Y}) - f_2(\bar{\boldsymbol{Y}})}{\varepsilon} \right) \\ &\ge \langle \nabla_{\boldsymbol{X}} f_0(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}}), \Delta \boldsymbol{X} \rangle + \langle \nabla_{\boldsymbol{Y}} f_0(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}}), \Delta \boldsymbol{Y} \rangle \\ &\quad + \liminf_{\varepsilon \downarrow 0} \frac{f_1(\bar{\boldsymbol{X}} + \varepsilon \Delta \boldsymbol{X}) - f_1(\bar{\boldsymbol{X}})}{\varepsilon} \\ &\quad + \liminf_{\varepsilon \downarrow 0} \frac{f_2(\bar{\boldsymbol{Y}} + \varepsilon \Delta \boldsymbol{Y}) - f_2(\bar{\boldsymbol{Y}})}{\varepsilon} \\ &= \langle \nabla_{\boldsymbol{X}} f_0(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}}), \Delta \boldsymbol{X} \rangle + \langle \nabla_{\boldsymbol{Y}} f_0(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}}), \Delta \boldsymbol{Y} \rangle \\ &\quad + f_1'(\bar{\boldsymbol{X}}; \Delta \boldsymbol{X}) + f_2'(\bar{\boldsymbol{Y}}; \Delta \boldsymbol{Y}) \\ &= \liminf_{\varepsilon \downarrow 0} \frac{f(\bar{\boldsymbol{X}} + \varepsilon \Delta \boldsymbol{X}, \bar{\boldsymbol{Y}}) - f(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}})}{\varepsilon} \\ &\quad + \liminf_{\varepsilon \downarrow 0} \frac{f(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}} + \varepsilon \Delta \boldsymbol{Y}) - f(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}})}{\varepsilon} \\ &\ge 0, \end{aligned}$$

where the last inequality holds because of inequalities (7) and (8). □

Remark 2. The boundedness of level set $L = \{(\boldsymbol{X}, \boldsymbol{Y}) : f(\boldsymbol{X}, \boldsymbol{Y}) \le f(\boldsymbol{X}^{(0)}, \boldsymbol{Y}^{(0)})\}$ holds if, for example, $L_{ij} > -\infty$ and $U_{ij} < +\infty$ for all i, j.

Remark 3. We can extend our proposed algorithm to a closed convex constraint on $\boldsymbol{Y}$, instead of the box constraint. If we can efficiently solve the subproblem to update $\boldsymbol{Y}$, the algorithm works well. In practice, the subproblem can be solved faster than the subproblem to update $\boldsymbol{X}$ with the truncated SVD on $\boldsymbol{Y}$.

We proved the convergence to a stationary point above instead of an optimal solution. A set of stationary points contains every local optimal solution and of course the global optimal solution. Thus, in practice, we run our algorithm from multiple initial points and select the best stationary point provided by our algorithm.

5 PRELIMINARY EXPERIMENTS

We show preliminary numerical results using synthetic dataset to investigate a convergence rate of our algorithm and an effect given by differences of initial points. We executed all experiments on a macOS Sierra 10.12.6 with an Intel Core m3, a 1.1 GHz clock speed, and 8 GB of physical memory. We implemented our algorithms in MATLAB (R2017b).

We generated matrix $\boldsymbol{A} \in \mathbb{R}^{20 \times 100}$ such that $\operatorname{rank}(\boldsymbol{A}) = 10$ and $0.5 \le A_{ij} \le 5.5$ for all (i, j) by multiplying two matrices $\boldsymbol{B} \in \mathbb{R}^{20 \times 9}$ and $\boldsymbol{C} \in \mathbb{R}^{9 \times 100}$

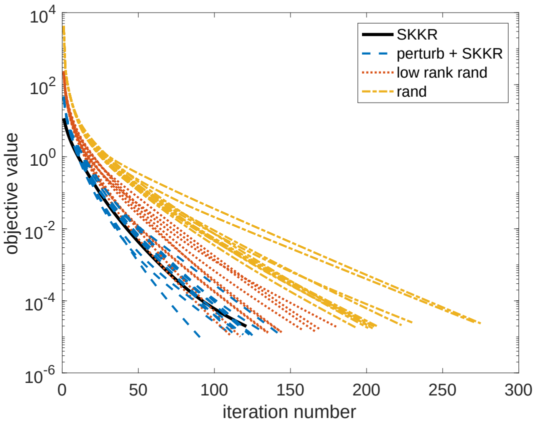

Figure 1: Comparison of iteration numbers.

and some scaling, where all entries of $\boldsymbol{B}$ and $\boldsymbol{C}$ are identically sampled from the standard normal distribution. Then, we round all entries of $\boldsymbol{A}$, *i.e.*, $\bar{A}_{ij} \in \{1,2,\dots,5\}$, and randomly missed 80% of them. We used the incomplete matrix as $\boldsymbol{V}$. In our algorithm, we set parameters to $r = 10$ for computing truncated SVD and $\lambda = 1$ for the weight parameter in the objective function.

In this experiment, we compared four methods of generating initial points:

SKKR We applied the method proposed by Sarwar et al. (2000) to given $\boldsymbol{V}$ and used the output as an initial point of our algorithm;

perturb + SKKR We applied the method proposed by Sarwar et al. (2000) to perturbed $\boldsymbol{V}$'s and used the outputs as initial points of our algorithm;

low rank rand We generated initial points of our algorithm in the same way of generating $\boldsymbol{A}$;

rand We generated 20×100 matrices whose entries were identically sampled from the standard normal distribution and used it as initial points.

In each method of perturb + SKKR, low rank rand, and rand, we generated 10 initial points and obtained 10 different solutions.

The differences of iteration numbers necessary to converge to stationary points are shown in Figure 1. The horizontal axis is iteration number and the vertical axis is objective value. From Figure 1, we can see that if we run our algorithm from initial points generated by perturb + SKKR and low rank rand, it converged faster than from an initial point provided by SKKR. On the other hand, it took much more time to converge if it started from initial points generated by rand.

The objective values at the resulting stationary points are shown in Figure 2. The vertical axis is objective value. The black horizontal line around 2×10^{-5} depicts the objective value of the stationary point provided by our algorithm starting from an initial point generated by SKKR. From Figure 2, we can see that all solutions provided by our algorithm starting from initial points generated by perturb + SKKR and low rank rand are better than that by SKKR. Some solutions resulting from rand is also smaller than that from SKKR.

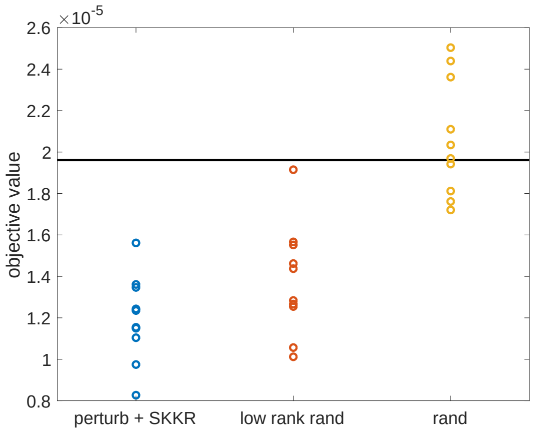

Figure 2: Comparison of objective values.

6 CONCLUSION

In this paper, we proposed a low-rank matrix approximation model for completing a matrix with missing values. Our proposed model utilizes not only a rank constraint but also a box constraint. Owing to the box constraint, this model shows promise for use in recommendation systems. In addition, we proposed an alternating minimization algorithm for solving our proposed model, and proved that a sequence generated by our proposed algorithm converges to a stationary point under a mild assumption. Our numerical results are preliminary, however, we showed that our proposed algorithm converges quickly and provides better solution than the existing method proposed by Sarwar et al. (2000) in a specific case. The performance of our algorithm depends on an initial guess. Hence, we should try multiple initial guesses and take the best one. In our experiments, we fixed parameters r and λ. However, we should employ cross validation to decide appropriate values for such parameters. We left extensive numerical experiments as our future work.

ACKNOWLEDGEMENTS

We thank to Prof. Tomomi Matsui and anonymous referees for providing helpful and useful comments, respectively.

REFERENCES

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

Gillis, N. and Glineur, F. (2011). Low-rank matrix approximation with weights or missing data is NP-hard. *SIAM Journal on Matrix Analysis and Applications*, 32(4):1149–1165.

Golub, G. H. and van Loan, C. F. (2013). *Matrix Computations*. The Johns Hopkins University Press, fourth edition.

Ilin, A. and Raiko, T. (2010). Practical approaches to principal component analysis in the presence of missing values. *Journal of Machine Learning Research*, 11(Jul):1957–2000.

Olsson, C. and Oskarsson, M. (2009). A convex approach to low rank matrix approximation with missing data. In Salberg, A.-B., Hardeberg, J. Y., and Jenssen, R., editors, *Proceedings of the 16th Scandinavian Conference on Image Analysis (SCIA '09)*, pages 301–309.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2000). Application of dimensionality reduction in recommender system—a case study. In Kohavi, R., Masand, B., Spiliopoulou, M., and Srivastava, J., editors, *Proceedings of the ACM WEBKDD 2000 Workshop*.

Su, X. and Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009(421425):1–19.

Tipping, M. E. and Bishop, C. M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society. Series B*, 61(3):611–622.

Trefethen, L. N. and Bau, D. (1997). *Numerical Linear Algebra*. SIAM.

Tseng, P. (2001). Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109(3):475–494.