Exact and Stable Recovery of Pairwise Interaction Tensors

Shouyuan Chen **Michael R. Lyu** **Irwin King**
The Chinese University of Hong Kong
{sychen,lyu,king}@cse.cuhk.edu.hk

Zenglin Xu
Purdue University
xu218@purdue.edu

Abstract

Tensor completion from incomplete observations is a problem of significant practical interest. However, it is unlikely that there exists an efficient algorithm with provable guarantee to recover a general tensor from a limited number of observations. In this paper, we study the recovery algorithm for pairwise interaction tensors, which has recently gained considerable attention for modeling multiple attribute data due to its simplicity and effectiveness. Specifically, in the absence of noise, we show that one can exactly recover a pairwise interaction tensor by solving a constrained convex program which minimizes the weighted sum of nuclear norms of matrices from $O(nr\log^2(n))$ observations. For the noisy cases, we also prove error bounds for a constrained convex program for recovering the tensors. Our experiments on the synthetic dataset demonstrate that the recovery performance of our algorithm agrees well with the theory. In addition, we apply our algorithm on a temporal collaborative filtering task and obtain state-of-the-art results.

1 Introduction

Many tasks of recommender systems can be formulated as recovering an unknown tensor (multi-way array) from a few observations of its entries [17, 26, 25, 21]. Recently, convex optimization algorithms for recovering a matrix, which is a special case of tensor, have been extensively studied [7, 22, 6]. Moreover, there are several theoretical developments that guarantee *exact recovery* of most low-rank matrices from partial observations using nuclear norm minimization [8, 5]. These results seem to suggest a promising direction to solve the general problem of tensor recovery.

However, there are inevitable obstacles to generalize the techniques for matrix completion to tensor recovery, since a number of fundamental computational problems of matrix is NP-hard in their tensorial analogues [10]. For instance, Håstad showed that it is NP-hard to compute the rank of a given tensor [9]; Hillar and Lim proved the NP-hardness to decompose a given tensor into sum of rank-one tensors even if a tensor is fully observed [10]. The existing evidence suggests that it is very unlikely that there exists an efficient exact recovery algorithm for general tensors with missing entries. Therefore, it is natural to ask whether it is possible to identify a useful class of tensors for which we can devise an exact recovery algorithm.

In this paper, we focus on *pairwise interaction tensors*, which have recently demonstrated strong performance in several recommendation applications, e.g. tag recommendation [19] and sequential data analysis [18]. Pairwise interaction tensors are a special class of general tensors, which directly model the pairwise interactions between different attributes. Take movie recommendation as an example, to model a user's ratings for movies varying over time, a pairwise interaction tensor assumes that each rating is determined by three factors: the user's inherent preference on the movie, the movie's trending popularity and the user's varying mood over time. Formally, pairwise interaction tensor assumes that each entry T_{ijk} of a tensor $\mathcal{T}$ of size $n_1 \times n_2 \times n_3$ is given by following

$$T_{ijk} = \left\langle \mathbf{u}_i^{(a)}, \mathbf{v}_j^{(a)} \right\rangle + \left\langle \mathbf{u}_j^{(b)}, \mathbf{v}_k^{(b)} \right\rangle + \left\langle \mathbf{u}_k^{(c)}, \mathbf{v}_i^{(c)} \right\rangle, \quad \text{for all } (i,j,k) \in [n_1] \times [n_2] \times [n_3], \quad (1)$$

where $\{\mathbf{u}_i^{(a)}\}_{i\in[n_1]}, \{\mathbf{v}_i^{(a)}\}_{j\in[n_2]}$ are r_1 dimensional vectors, $\{\mathbf{u}_j^{(b)}\}_{j\in[n_2]}, \{\mathbf{v}_k^{(b)}\}_{k\in[n_3]}$ are r_2 dimensional vectors and $\{\mathbf{u}_k^{(c)}\}_{k\in[n_3]}, \{\mathbf{v}_i^{(c)}\}_{i\in[n_1]}$ are r_3 dimensional vectors, respectively. [1]

The existing recovery algorithms for pairwise interaction tensor use local optimization methods, which do not guarantee the recovery performance [18, 19]. In this paper, we design efficient recovery algorithms for pairwise interaction tensors with rigorous guarantee. More specifically, in the absence of noise, we show that one can exactly recover a pairwise interaction tensor by solving a constrained convex program which minimizes the weighted sum of nuclear norms of matrices from $O(nr\log^2(n))$ observations, where $n = \max\{n_1, n_2, n_3\}$ and $r = \max\{r_1, r_2, r_3\}$. For noisy cases, we also prove error bounds for a constrained convex program for recovering the tensors.

In the proof of our main results, we reformulated the recovery problem as a constrained matrix completion problem with a special observation operator. Previously, Gross et al. [8] have showed that the nuclear norm heuristic can exactly recover low rank matrix from a sufficient number of observations of an orthogonal observation operator. We note that the orthogonality is critical to their argument. However, the observation operator, in our case, turns out to be non-orthogonal, which becomes a major challenge in our proof. In order to deal with the non-orthogonal operator, we have substantially extended their technique in our proof. We believe that our technique can be generalized to handle other matrix completion problem with non-orthogonal observation operators.

Moreover, we extend existing singular value thresholding method to develop a simple and scalable algorithm for solving the recovery problem in both exact and noisy cases. Our experiments on the synthetic dataset demonstrate that the recovery performance of our algorithm agrees well with the theory. Finally, we apply our algorithm on a temporal collaborative filtering task and obtain state-of-the-art results.

2 Recovering pairwise interaction tensors

In this section, we first introduce the matrix formulation of pairwise interaction tensors and specify the recovery problem. Then we discuss the sufficient conditions on pairwise interaction tensors for which an exact recovery would be possible. After that we formulate the convex program for solving the recovery problem and present our theoretical results on the sample bounds for achieving an exact recovery. In addition, we also show a quadratically constrained convex program is stable for the recovery from noisy observations.

A matrix formulation of pairwise interaction tensors. The original formulation of pairwise interaction tensors by Rendle et al. [19] is given by Eq. (1), in which each entry of a tensor is the sum of inner products of feature vectors. We can reformulate Eq. (1) more concisely using matrix notations. In particular, we can rewrite Eq. (1) as follows

$$T_{ijk} = A_{ij} + B_{jk} + C_{ki}, \quad \text{for all } (i,j,k) \in [n_1]\times[n_2]\times[n_3], \tag{2}$$

where we set $A_{ij} = \left\langle \mathbf{u}_i^{(a)}, \mathbf{v}_j^{(a)} \right\rangle$, $B_{jk} = \left\langle \mathbf{u}_j^{(b)}, \mathbf{v}_k^{(b)} \right\rangle$, and $C_{ki} = \left\langle \mathbf{u}_k^{(c)}, \mathbf{v}_i^{(c)} \right\rangle$ for all (i,j,k). Clearly, matrices $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ are rank r_1, r_2 and r_3 matrices, respectively.

We call tensor $\mathcal{T} \in \mathbb{R}^{n_1\times n_2\times n_3}$ a *pairwise interaction tensor*, which is denoted as $\mathcal{T} = \mathrm{Pair}(\mathbf{A}, \mathbf{B}, \mathbf{C})$, if $\mathcal{T}$ obeys Eq. (2). We note that this concise definition is equivalent to the original one. In the rest of this paper, we will exclusively use the matrix formulation of pairwise interaction tensors.

Recovery problem. Suppose we have partial observations of a pairwise interaction tensor $\mathcal{T} = \mathrm{Pair}(\mathbf{A}, \mathbf{B}, \mathbf{C})$. We write $\Omega \subseteq [n_1]\times[n_2]\times[n_3]$ to be the set of indices of m observed entries. In this work, we shall assume Ω is sampled uniformly from the collection of all sets of size m. Our goal is to recover matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and therefore the entire tensor $\mathcal{T}$ from exact or noisy observations of $\{T_{ijk}\}_{(ijk)\in\Omega}$.

Before we proceed to the recovery algorithm, we first discuss when the recovery is possible.

Recoverability: uniqueness. The original recovery problem for pairwise interaction tensors is ill-posed due to a uniqueness issue. In fact, for any pairwise interaction tensor $\mathcal{T} = \mathrm{Pair}(\mathbf{A}, \mathbf{B}, \mathbf{C})$,

[1] For simplicity, we only consider three-way tensors in this paper.

we can construct infinitely manly different sets of matrices $\mathbf{A}', \mathbf{B}', \mathbf{C}'$ such that $\mathrm{Pair}(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \mathrm{Pair}(\mathbf{A}', \mathbf{B}', \mathbf{C}')$. For example, we have $T_{ijk} = A_{ij} + B_{jk} + C_{ki} = (A_{ij} + \delta a_i) + B_{jk} + (C_{ki} + (1-\delta)a_i)$, where $\delta \neq 0$ can be any non-zero constant and $\mathbf{a}$ is an arbitrary non-zero vector of size n_1. Now, we can construct $\mathbf{A}', \mathbf{B}'$ and $\mathbf{C}'$ by setting $A'_{ij} = A_{ij} + \delta a_i, B'_{jk} = B_{jk}$ and $C'_{ki} = C_{ki} + (1-\delta)a_i$. It is clear that $\mathcal{T} = \mathrm{Pair}(\mathbf{A}', \mathbf{B}', \mathbf{C}')$.

This ambiguity prevents us to recover $\mathbf{A}, \mathbf{B}, \mathbf{C}$ even if $\mathcal{T}$ is fully observed, since it is entirely possible to recover $\mathbf{A}', \mathbf{B}', \mathbf{C}'$ instead of $\mathbf{A}, \mathbf{B}, \mathbf{C}$ based on the observations. In order to avoid this obstacle, we construct a set of constraints such that, given any pairwise interaction tensor $\mathrm{Pair}(\mathbf{A}, \mathbf{B}, \mathbf{C})$, there exists unique matrices $\mathbf{A}', \mathbf{B}', \mathbf{C}'$ satisfying the constraints and obeys $\mathrm{Pair}(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \mathrm{Pair}(\mathbf{A}', \mathbf{B}', \mathbf{C}')$. Formally, we prove the following proposition.

Proposition 1. *For any pairwise interaction tensor $\mathcal{T} = \mathrm{Pair}(\mathbf{A}, \mathbf{B}, \mathbf{C})$, there exists unique $\mathbf{A}' \in S_A, \mathbf{B}' \in S_B, \mathbf{C}' \in S_C$ such that $\mathrm{Pair}(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \mathrm{Pair}(\mathbf{A}', \mathbf{B}', \mathbf{C}')$ where we define $S_B = \{\mathbf{M} \in \mathbb{R}^{n_2 \times n_3} : \mathbf{1}^T \mathbf{M} = \mathbf{0}^T\}$,$S_C = \{\mathbf{M} \in \mathbb{R}^{n_3 \times n_1} : \mathbf{1}^T \mathbf{M} = \mathbf{0}^T\}$ and $S_A = \{\mathbf{M} \in \mathbb{R}^{n_1 \times n_2} : \mathbf{1}^T \mathbf{M} = \left(\frac{1}{n_2} \mathbf{1}^T \mathbf{M} \mathbf{1}\right) \mathbf{1}^T\}$.*

We point out that there is a natural connection between the uniqueness issue and the "bias" components, which is a quantity of much attention in the field of recommender system [13]. Due to lack of space, we defer the detailed discussion on this connection and the proof of Proposition 1 to the supplementary material.

Recoverability: incoherence. It is easy to see that recovering a pairwise tensor $\mathcal{T} = \mathrm{Pair}(\mathbf{A}, \mathbf{0}, \mathbf{0})$ is equivalent to recover the matrix $\mathbf{A}$ from a subset of its entries. Therefore, the recovery problem of pairwise interaction tensors subsumes matrix completion problem as a special case. Previous studies have confirmed that the *incoherence condition* is an essential requirement on the matrix in order to guarantee a successful recovery of matrices. This condition can be stated as follows.

Let $\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^T$ be the singular value decomposition of a rank r matrix $\mathbf{M}$. We call matrix $\mathbf{M}$ is (μ_0, μ_1)-incoherent if $\mathbf{M}$ satisfies:

A0. For all $i \in [n_1]$ and $j \in [n_2]$, we have $\frac{n_1}{r} \sum_{k \in [r]} U_{ik}^2 \leq \mu_0$ and $\frac{n_2}{r} \sum_{k \in [r]} V_{jk}^2 \leq \mu_0$.

A1. The maximum entry of $\mathbf{U}\mathbf{V}^T$ is bounded by $\mu_1 \sqrt{r/(n_1 n_2)}$ in absolute value.

It is well known the recovery is possible only if the matrix is (μ_0, μ_1)-incoherent for bounded μ_0, μ_1 (i.e, μ_0, μ_1 is poly-logarithmic with respect to n). Since the matrix completion problem is reducible to the recovery problem for pairwise interaction tensors, our theoretical result will inherit the incoherence assumptions on matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$.

Exact recovery in the absence of noise. We first consider the scenario where the observations are exact. Specifically, suppose we are given m observations $\{T_{ijk}\}_{(ijk) \in \Omega}$, where Ω is sampled from uniformly at random from $[n_1] \times [n_2] \times [n_3]$. We propose to recover matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and therefore tensor $\mathcal{T} = \mathrm{Pair}(\mathbf{A}, \mathbf{B}, \mathbf{C})$ using the following convex program,

$$\begin{aligned} \underset{\mathbf{X} \in S_A, \mathbf{Y} \in S_B, \mathbf{Z} \in S_C}{\text{minimize}} \quad & \sqrt{n_3} \left\|\mathbf{X}\right\|_* + \sqrt{n_1} \left\|\mathbf{Y}\right\|_* + \sqrt{n_2} \left\|\mathbf{Z}\right\|_* \\ \text{subject to} \quad & X_{ij} + Y_{jk} + Z_{ki} = T_{ijk}, \quad (i,j,k) \in \Omega, \end{aligned} \tag{3}$$

where $\left\|\mathbf{M}\right\|_*$ denotes the nuclear norm of matrix $\mathbf{M}$, which is the sum of singular values of $\mathbf{M}$, and S_A, S_B, S_C is defined in Proposition 1.

We show that, under the incoherence conditions, the above nuclear norm minimization method successful recovers a pairwise interaction tensor $\mathcal{T}$ when the number of observations m is $O(nr \log^2 n)$ with high probability.

Theorem 1. *Let $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ be a pairwise interaction tensor $\mathcal{T} = \mathrm{Pair}(\mathbf{A}, \mathbf{B}, \mathbf{C})$ and $\mathbf{A} \in S_A, \mathbf{B} \in S_B, \mathbf{C} \in S_C$ as defined in Proposition 1. Without loss of generality assume that $9 \leq n_1 \leq n_2 \leq n_3$. Suppose we observed m entries of $\mathcal{T}$ with the locations sampled uniformly at random from $[n_1] \times [n_2] \times [n_3]$ and also suppose that each of $\mathbf{A}, \mathbf{B}, \mathbf{C}$ is (μ_0, μ_1)-incoherent. Then, there exists a universal constant C, such that if*

$$m > C \max\{\mu_1^2, \mu_0\} n_3 r \beta \log^2(6n_3),$$

where $r = \max\{\mathrm{rank}(\mathbf{A}), \mathrm{rank}(\mathbf{B}), \mathrm{rank}(\mathbf{C})\}$ and $\beta > 2$ is a parameter, the minimizing solution $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ for program Eq. (3) is unique and satisfies $\mathbf{X} = \mathbf{A}, \mathbf{Y} = \mathbf{B}, \mathbf{Z} = \mathbf{C}$ with probability at least $1 - \log(6n_3) 6n_3^{2-\beta} - 3n_3^{2-\beta}$.

Stable recovery in the presence of noise. Now, we move to the case where the observations are perturbed by noise with bounded energy. In particular, our noisy model assumes that we observe

$$\hat{T}_{ijk} = T_{ijk} + \sigma_{ijk}, \quad \text{for all } (i,j,k) \in \Omega, \tag{4}$$

where σ_{ijk} is a noise term, which maybe deterministic or stochastic. We assume σ has bounded energy on Ω and specifically that $\|\mathcal{P}_\Omega(\sigma)\|_F \leq \epsilon_1$ for some $\epsilon_1 > 0$, where $\mathcal{P}_\Omega(\cdot)$ denotes the restriction on Ω. Under this assumption on the observations, we derive the error bound of the following quadratically-constrained convex program, which recover $\mathcal{T}$ from the noisy observations.

$$\begin{aligned} \underset{\mathbf{X}\in S_A, \mathbf{Y}\in S_B, \mathbf{Z}\in S_C}{\text{minimize}} \quad & \sqrt{n_3}\,\|\mathbf{X}\|_* + \sqrt{n_1}\,\|\mathbf{Y}\|_* + \sqrt{n_2}\,\|\mathbf{Z}\|_* \\ \text{subject to} \quad & \left\|\mathcal{P}_\Omega(\text{Pair}(\mathbf{X},\mathbf{Y},\mathbf{Z})) - \mathcal{P}_\Omega(\hat{\mathcal{T}})\right\|_F \leq \epsilon_2. \end{aligned} \tag{5}$$

Theorem 2. *Let $\mathcal{T} = \text{Pair}(\mathbf{A},\mathbf{B},\mathbf{C})$ and $\mathbf{A} \in S_A, \mathbf{B} \in S_B, \mathbf{C} \in S_C$. Let Ω be the set of observations as described in Theorem 1. Suppose we observe $\hat{T}_{ijk}$ for $(i,j,k) \in \Omega$ as defined in Eq.* (4) *and also assume that $\|\mathcal{P}_\Omega(\sigma)\|_F \leq \epsilon_1$ holds. Denote the reconstruction error of the optimal solution $\mathbf{X},\mathbf{Y},\mathbf{Z}$ of convex program Eq.* (5) *as $\mathbf{E} = \text{Pair}(\mathbf{X},\mathbf{Y},\mathbf{Z}) - \mathcal{T}$. Also assume that $\epsilon_1 \leq \epsilon_2$. Then, we have*

$$\|\mathbf{E}\|_* \leq 5\sqrt{\frac{2rn_1n_2^2}{8\beta\log(n_1)}}(\epsilon_1+\epsilon_2),$$

with probability at least $1 - \log(6n_3)6n_3^{2-\beta} - 3n_3^{2-\beta}$.

The proof of Theorem 1 and Theorem 2 is available in the supplementary material.

Related work. Rendle et al. [19] proposed pairwise interaction tensors as a model used for tag recommendation. In a subsequent work, Rendle et al. [18] applied pairwise interaction tensors in the sequential analysis of purchase data. In both applications, their methods using pairwise interaction tensor demonstrated excellent performance. However, their algorithms are prone to local optimal issues and the recovered tensor might be very different from its true value. In contrast, our main results, Theorem 1 and Theorem 2, guarantee that a convex program can exactly or accurately recover the pairwise interaction tensors from $O(nr\log^2(n))$ observations. In this sense, our work can be considered as a more effective way to recover pairwise interaction tensors from partial observations.

In practice, various tensor factorization methods are used for estimating missing entries of tensors [12, 20, 1, 26, 16]. In addition, inspired by the success of nuclear norm minimization heuristics in matrix completion, several work used a generalized nuclear norm for tensor recovery [23, 24, 15]. However, these work do not guarantee exact recovery of tensors from partial observations.

3 Scalable optimization algorithm

There are several possible methods to solving the optimization problems Eq. (3) and Eq. (5). For small problem sizes, one may reformulate the optimization problems as semi-definite programs and solve them using interior point method. The state-of-the-art interior point solvers offer excellent accuracy for finding the optimal solution. However, these solvers become prohibitively slow for pairwise interaction tensors larger than $100 \times 100 \times 100$. In order to apply the recover algorithms on large scale pairwise interaction tensors, we use singular value thresholding (SVT) algorithm proposed recently by Cai et al. [3], which is a first-order method with promising performance for solving nuclear norm minimization problems.

We first discuss the SVT algorithm for solving the exact completion problem Eq. (3). For convenience, we reformulate the original optimization objective Eq. (3) as follows,

$$\begin{aligned} \underset{\mathbf{X}\in S_A, \mathbf{Y}\in S_B, \mathbf{Z}\in S_C}{\text{minimize}} \quad & \|\mathbf{X}\|_* + \|\mathbf{Y}\|_* + \|\mathbf{Z}\|_* \\ \text{subject to} \quad & \frac{X_{ij}}{\sqrt{n_3}} + \frac{Y_{jk}}{\sqrt{n_1}} + \frac{Z_{ki}}{\sqrt{n_2}} = T_{ijk}, \quad (i,j,k) \in \Omega, \end{aligned} \tag{6}$$

where we have incorporated coefficients on the nuclear norm terms into the constraints. It is easy to see that the recovered tensor is given by $\text{Pair}(n_3^{-1/2}\mathbf{X}, n_1^{-1/2}\mathbf{Y}, n_2^{-1/2}\mathbf{Z})$, where $\mathbf{X},\mathbf{Y},\mathbf{Z}$ is the

optimal solution of Eq. (6). Our algorithm solves a slightly relaxed version of the reformulated objective Eq. (6),

$$\begin{aligned} \underset{\mathbf{X}\in S_A,\mathbf{Y}\in S_B,\mathbf{Z}\in S_C}{\text{minimize}} \; & \tau\left(\|\mathbf{X}\|_* + \|\mathbf{Y}\|_* + \|\mathbf{Z}\|_*\right) + \frac{1}{2}\left(\|\mathbf{X}\|_F^2 + \|\mathbf{Y}\|_F^2 + \|\mathbf{Z}\|_F^2\right) \\ \text{subject to} \quad & \frac{X_{ij}}{\sqrt{n_3}} + \frac{Y_{jk}}{\sqrt{n_1}} + \frac{Z_{ki}}{\sqrt{n_2}} = T_{ijk}, \quad (i,j,k) \in \Omega. \end{aligned} \tag{7}$$

It is easy to see that Eq. (7) is closely related to Eq. (6) and the original problem Eq. (3), as the relaxed problem converges to the original one as $\tau \to \infty$. Therefore by selecting a large value the parameter τ, a minimizing solution to Eq. (7) nearly minimizes Eq. (3).

Our algorithm iteratively minimizes Eq. (7) and produces a sequence of matrices $\{\mathbf{X}^k, \mathbf{Y}^k, \mathbf{Z}^k\}$ converging to the optimal solution $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ that minimizes Eq. (7). We begin with several definitions. For observations $\Omega = \{a_i, b_i, c_i | i \in [m]\}$, let operators $\mathcal{P}_{\Omega_A} : \mathbb{R}^{n_1\times n_2} \to \mathbb{R}^m$, $\mathcal{P}_{\Omega_B} : \mathbb{R}^{n_2\times n_3} \to \mathbb{R}^m$ and $\mathcal{P}_{\Omega_C} : \mathbb{R}^{n_3\times n_1} \to \mathbb{R}^m$ represents the influence of $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ on the m observations. In particular,

$$\mathcal{P}_{\Omega_A}(\mathbf{X}) = \frac{1}{\sqrt{n_3}}\sum_{i=1}^m X_{a_i b_i}\delta_i, \quad \mathcal{P}_{\Omega_B}(\mathbf{Y}) = \frac{1}{\sqrt{n_1}}\sum_{i=1}^m Y_{b_i c_i}\delta_i, \text{ and } \mathcal{P}_{\Omega_C}(\mathbf{Z}) = \frac{1}{\sqrt{n_2}}\sum_{i=1}^m Z_{c_i a_i}\delta_i.$$

It is easy to verify that $\mathcal{P}_{\Omega_A}(\mathbf{X}) + \mathcal{P}_{\Omega_B}(\mathbf{Y}) + \mathcal{P}_{\Omega_C}(\mathbf{Z}) = \mathcal{P}_{\Omega}(\text{Pair}(n_3^{-1/2}\mathbf{X}, n_1^{-1/2}\mathbf{Y}, n_2^{-1/2}\mathbf{Z}))$. We also denote $\mathcal{P}^*_{\Omega_A}$ be the adjoint operator of $\mathcal{P}_{\Omega_A}$ and similarly define $\mathcal{P}^*_{\Omega_B}$ and $\mathcal{P}^*_{\Omega_C}$. Finally, for a matrix $\mathbf{X}$ for size $n_1 \times n_2$, we define center$(\mathbf{X}) = \mathbf{X} - \frac{1}{n_1}\mathbf{1}\mathbf{1}^T\mathbf{X}$ as the column centering operator that removes the mean of each n_2 columns, i.e., $\mathbf{1}^T\text{center}(\mathbf{X}) = \mathbf{0}^T$.

Starting with $\mathbf{y}^0 = 0$ and $k = 1$, our algorithm iteratively computes

$$\begin{aligned} \text{Step (1).} \quad \mathbf{X}^k &= \text{shrink}_A(\mathcal{P}^*_{\Omega_A}(\mathbf{y}^{k-1}), \tau), \\ \mathbf{Y}^k &= \text{shrink}_B(\mathcal{P}^*_{\Omega_B}(\mathbf{y}^{k-1}), \tau), \\ \mathbf{Z}^k &= \text{shrink}_C(\mathcal{P}^*_{\Omega_C}(\mathbf{y}^{k-1}), \tau), \\ \text{Step (2e).} \quad \mathbf{e}^k &= \mathcal{P}_{\Omega}(\mathcal{T}) - \mathcal{P}_{\Omega}(\text{Pair}(n_3^{-1/2}\mathbf{X}, n_1^{-1/2}\mathbf{Y}, n_2^{-1/2}\mathbf{Z})) \\ \mathbf{y}^k &= \mathbf{y}^{k-1} + \delta\mathbf{e}^k. \end{aligned}$$

Here shrink$_A$ is a shrinkage operator defined as follows

$$\text{shrink}_A(\mathbf{M}, \tau) \triangleq \underset{\tilde{\mathbf{M}}\in S_A}{\arg\min} \frac{1}{2}\left\|\tilde{\mathbf{M}} - \mathbf{M}\right\|_F^2 + \tau\left\|\tilde{\mathbf{M}}\right\|_*. \tag{8}$$

Shrinkage operators shrink$_B$ and shrink$_C$ are defined similarly except they require $\tilde{\mathbf{M}}$ belongs S_B and S_C, respectively. We note that our definition of the shrinkage operators shrink$_A$, shrink$_B$ and shrink$_C$ are slightly different from that of the original SVT [3] algorithm, where $\tilde{\mathbf{M}}$ is unconstrained. We can show that our constrained version of shrinkage operators can also be calculated using singular value decompositions of column centered matrices.

Let the SVD of the column centered matrix center$(\mathbf{M})$ be center$(\mathbf{M}) = \mathbf{U}\Sigma\mathbf{V}^T$, $\Sigma = \text{diag}(\{\sigma_i\})$. We can prove that the shrinkage operator shrink$_B$ is given by

$$\text{shrink}_B(\mathbf{M}, \tau) = \mathbf{U}\,\text{diag}(\{\sigma_i - \tau\}_+)\mathbf{V}^T, \tag{9}$$

where s_+ is the positive part of s, that is, $s_+ = \max\{0, s\}$. Since subspace S_C is structurally identical to S_B, it is easy to see that the calculation of shrink$_C$ is identical to that of shrink$_B$. The computation of shrink$_A$ is a little more complicated. We have

$$\text{shrink}_A(\mathbf{M}, \tau) = \mathbf{U}\,\text{diag}(\{\sigma_i - \tau\}_+)\mathbf{V}^T + \frac{1}{\sqrt{n_1 n_2}}\left(\{\delta - \tau\}_+ + \{\delta + \tau\}_-\right)\mathbf{1}\mathbf{1}^T, \tag{10}$$

where $\mathbf{U}\Sigma\mathbf{V}^T$ is still the SVD of center$(\mathbf{M})$, $\delta = \frac{1}{\sqrt{n_1 n_2}}\mathbf{1}^T\mathbf{M}\mathbf{1}$ is a constant and $s_- = \min\{0, s\}$ is the negative part of s. The algorithm iterates between Step (1) and Step (2e) and produces a series of $(\mathbf{X}^k, \mathbf{Y}^k, \mathbf{Z}^k)$ converging to the optimal solution of Eq. (7). The iterative procedure terminates

when the training error is small enough, namely, $\left\|\mathbf{e}^k\right\|_F \leq \epsilon$. We refer interested readers to [3] for a convergence proof of the SVT algorithm.

The optimization problem for noisy completion Eq. (5) can be solved in a similar manner. We only need to modify Step (2e) to incorporate the quadratical constraint of Eq. (5) as follows

$$\text{Step (2n).} \qquad \mathbf{e}^k = \mathcal{P}_\Omega(\hat{\mathcal{T}}) - \mathcal{P}_\Omega(\text{Pair}(n_3^{-1/2}\mathbf{X}, n_1^{-1/2}\mathbf{Y}, n_2^{-1/2}\mathbf{Z}))$$
$$\begin{bmatrix} \mathbf{y}^k \\ s^k \end{bmatrix} = \mathcal{P}_\mathcal{K}\left(\begin{bmatrix} \mathbf{y}^{k-1} \\ s^{k-1} \end{bmatrix} + \delta \begin{bmatrix} \mathbf{e}^k \\ -\epsilon \end{bmatrix}\right),$$

where $\mathcal{P}_\Omega(\hat{\mathcal{T}})$ is the noisy observations and the cone projection operator $\mathcal{P}_\mathcal{K}$ can be explicitly computed by

$$\mathcal{P}_\mathcal{K} : (x,t) \to \begin{cases} (x,t) & \text{if } \|x\| \leq t, \\ \frac{\|x\|+t}{2\|x\|}(x, \|x\|) & \text{if } -\|x\| \leq t \leq \|x\|, \\ (0,0) & \text{if } t \leq -\|x\|. \end{cases}$$

By iterating between Step (1) and Step (2n) and selecting a sufficiently large τ, the algorithm generates a sequence of $\{\mathbf{X}^k, \mathbf{Y}^k, \mathbf{Z}^k\}$ that converges to a nearly optimal solution to the noisy completion program Eq. (5) [3]. We have also included a detailed description of both algorithms in the supplementary material.

At each iteration, we need to compute one singular value decomposition and perform a few elementary matrix additions. We can see that for each iteration k, $\mathbf{X}^k$ vanishes outside of $\Omega_A = \{a_i b_i\}$ and is sparse. Similarly $\mathbf{Y}^k$,$\mathbf{Z}^k$ are also sparse matrices. Previously, we showed that the computation of shrinkage operators requires a SVD of a column centered matrix center$(\mathbf{M}) - \frac{1}{n_1}\mathbf{1}\mathbf{1}^T\mathbf{X}$, which is the sum of a sparse matrix $\mathbf{M}$ and a rank-one matrix. Clearly the matrix-vector multiplication of the form center$(\mathbf{M})\mathbf{v}$ can be computed with time $O(n+m)$. This enables the use of Lanczos method based SVD implementations for example PROPACK [14] and SVDPACKC [2], which only needs subroutine of calculating matrix-vector products. In our implementation, we develop a customized version of SVDPACKC for computing the shrinkage operators. Further, for an appropriate choice of τ, $\{\mathbf{X}^k, \mathbf{Y}^k, \mathbf{Z}^k\}$ turned out to be low rank matrices, which matches the observations in the original SVT algorithm [3]. Hence, the storage cost $\mathbf{X}^k, \mathbf{Y}^k, \mathbf{Z}^k$ can be kept low and we only need to perform a partial SVD to get the first r singular vectors. The estimated rank r is gradually increased during the iterations using a similar method suggested in [3, Section 5.1.1]. We can see that, in sum, the overall complexity per iteration of the recovery algorithm is $O(r(n+m))$.

4 Experiments

Phase transition in exact recovery. We investigate how the number of measurements affects the success of exact recovery. In this simulation, we fixed $n_1 = 100, n_2 = 150, n_3 = 200$ and $r_1 = r_2 = r_3 = r$. We tested a variety of choices of (r, m) and for each choice of (r, m), we repeat the procedure for 10 times. At each time, we randomly generated $\mathbf{A} \in S_A, \mathbf{B} \in S_B, \mathbf{C} \in S_C$ of rank r. We generated $\mathbf{A} \in S_A$ by sampling two factor matrices $\mathbf{U}_A \in \mathbb{R}^{n_1 \times r}, \mathbf{V}_A \in \mathbb{R}^{n_2 \times r}$ with i.i.d. standard Gaussian entries and setting $\mathbf{A} = \mathcal{P}_{S_A}(\mathbf{U}_A\mathbf{V}_A^T)$, where $\mathcal{P}_{S_A}$ is the orthogonal projection onto subspace S_A. Matrices $\mathbf{B} \in S_B$ and $\mathbf{C} \in S_C$ are sampled in a similar way. We uniformly sampled a subset Ω of m entries and reveal them to the recovery algorithm. We deemed $\mathbf{A}, \mathbf{B}, \mathbf{C}$ successfully recovered if $(\|\mathbf{A}\|_F + \|\mathbf{B}\|_F + \|\mathbf{C}\|_F)^{-1}(\|\mathbf{X} - \mathbf{A}\|_F + \|\mathbf{Y} - \mathbf{B}\|_F + \|\mathbf{Z} - \mathbf{C}\|_F) \leq 10^{-3}$, where $\mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$ are the recovered matrices. Finally, we set the parameters τ, δ of the exact recovery algorithm by $\tau = 10\sqrt{n_1 n_2 n_3}$ and $\delta = 0.9m(n_1 n_2 n_3)^{-1}$.

Figure 1 shows the results of these experiments. The x-axis is the ratio between the number of measurements m and the degree of freedom $d = r(n_1 + n_2 - r) + r(n_2 + n_3 - r) + r(n_3 + n_1 - r)$. Note that a value of x-axis smaller than one corresponds to a case where there is infinite number of solutions satisfying given entries. The y-axis is the rank r of the synthetic matrices. The color of each grid indicates the empirical success rate. White denotes exact recovery in all 10 experiments, and black denotes failure for all experiments. From Figure 1 (Left), we can see that the algorithm succeeded almost certainly when the number of measurements is 2.5 times or larger than the degree of freedom for most parameter settings. We also observe that, near the boundary of $m/d \approx 2.5$, there is a relatively sharp phase transition. To verify this phenomenon, we repeated the experiments,

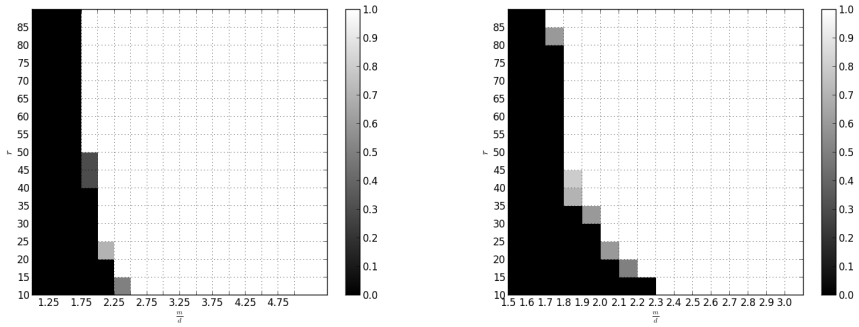


Figure 1: Phase transition with respect to rank and degree of freedom. Left: $m/d \in [1, 5]$. Right: $m/d \in [1.5, 3.0]$.

but only vary m/d between 1.5 and 3.0 with finer steps. The results on Figure 1 (Right) shows that the phase transition continued to be sharp at a higher resolution.

Stability of recovering from noisy data. In this simulation, we show the recovery performance with respect to noisy data. Again, we fixed $n_1 = 100, n_2 = 150, n_3 = 200$ and $r_1 = r_2 = r_3 = r$ and tested against different choices of (r, m). For each choice of (r, m), we sampled the ground truth $\mathbf{A}, \mathbf{B}, \mathbf{C}$ using the same method as in the previous simulation. We generated Ω uniformly at random. For each entry $(i, j, k) \in \Omega$, we simulated the noisy observation $\hat{T}_{ijk} = T_{ijk} + \epsilon_{ijk}$, where ϵ_{ijk} is a zero-mean Gaussian random variable with variance σ_n^2. Then, we revealed $\{\hat{T}_{ijk}\}_{(ijk)\in\Omega}$ to the noisy recovery algorithm and collect the recovered matrix $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$. The error of recovery result is measured by $(\|\mathbf{X} - \mathbf{A}\|_F + \|\mathbf{Y} - \mathbf{B}\|_F + \|\mathbf{Z} - \mathbf{C}\|_F)/(\|\mathbf{A}\|_F + \|\mathbf{B}\|_F + \|\mathbf{C}\|_F)$. We tested the algorithm with a range of noise levels and for each different configuration of (r, m, σ_n^2), we repeated the experiments for 10 times and recorded the mean and standard deviation of the relative error.

noise level	relative error
0.1	0.1020 ± 0.0005
0.2	0.1972 ± 0.0007
0.3	0.2877 ± 0.0011
0.4	0.3720 ± 0.0015
0.5	0.4524 ± 0.0015

(a) Fix $r = 20$, $m = 5d$ and noise level varies.

observations m	relative error
$m = 3d$	0.1445 ± 0.0008
$m = 4d$	0.1153 ± 0.0006
$m = 5d$	0.1015 ± 0.0004
$m = 6d$	0.0940 ± 0.0007
$m = 7d$	0.0920 ± 0.0011

(b) Fix $r = 20$, 0.1 noise level and m varies.

rank r	relative error
10	0.1134 ± 0.0006
20	0.1018 ± 0.0007
30	0.0973 ± 0.0037
40	0.1032 ± 0.0212
50	0.1520 ± 0.0344

(c) Fix $m = 5d$, 0.1 noise level and r varies.

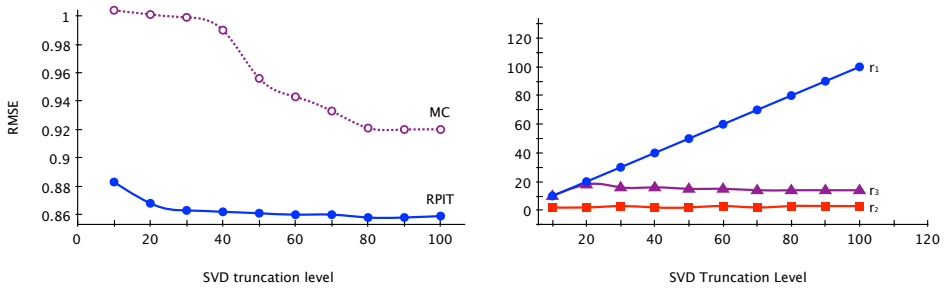
Table 1: Simulation results of noisy data.

We present the result of the experiments in Table 1. From the results in Table 1(a), we can see that the error in the solution is proportional to the noise level. Table 1(b) indicates that the recovery is not reliable when we have too few observations, while the performance of the algorithm is much more stable for a sufficient number of observations around four times of the degree of freedom. Table 1(c) shows that the recovery error is not affected much by the rank, as the number of observations scales with the degree of freedom in our setting.

Temporal collaborative filtering. In order to demonstrate the performance of pairwise interaction tensor on real world applications, we conducted experiments on the Movielens dataset. The MovieLens dataset contains 1,000,209 ratings from 6,040 users and 3,706 movies from April, 2000 and February, 2003. Each rating from Movielens dataset is accompanied with time information provided in seconds. We transformed each timestamp into its corresponding calendar month. We randomly select 10% ratings as test set and use the rest of the ratings as training set. In the end, we obtained a tensor $\mathcal{T}$ of size $6040 \times 3706 \times 36$, in which the axes corresponded to user, movie and timestamp respectively, with 0.104% observed entries as the training set. We applied the noisy recovery algorithm on the training set. Following previous studies which applies SVT algorithm on movie recommendation datasets [11], we used a pre-specified truncation level r for computing SVD in each iteration, i.e., we only kept top r singular vectors. Therefore, the rank of recovered matrices are at most r.

We evaluated the prediction performance in terms of root mean squared error (RMSE). We compared our algorithm with noisy matrix completion method using standard SVT optimization algorithm [3, 4] to the same dataset while ignore the time information. Here we can regard the noisy matrix completion algorithm as a special case of the recover a pairwise interaction tensor of size $6040 \times 3706 \times 1$, i.e., the time information is ignored. We also noted that the training tensor had more than one million observed entries and 80 millions total entries. This scale made a number of tensor recovery algorithms, for example Tucker decomposition and PARAFAC [12], impractical to apply on the dataset. In contrast, our recovery algorithm took 2430 seconds to finish on a standard workstation for truncation level $r = 100$.

The experimental result is shown in Figure 2. The empirical result of Figure 2(a) suggests that, by incorporating the temporal information, pairwise interaction tensor recovery algorithm consistently outperformed the matrix completion method. Interestingly, we can see that, for most parameter settings in Figure 2(b), our algorithm recovered a rank 2 matrix $\mathbf{Y}$ representing the change of movie popularity over time and a rank 15 matrix $\mathbf{Z}$ that encodes the change of user interests over time. The reason of the improvement on the prediction performance may be that the recovered matrix $\mathbf{Y}$ and $\mathbf{Z}$ provided meaningful signal. Finally, we note that our algorithm achieves a RMSE of 0.858 when the truncation level is set to 50, which slightly outperforms the RMSE=0.861 (quote from Figure 7 of the paper) result of 30-dimensional Bayesian Probabilistic Tensor Factorization (BPTF) on the same dataset, where the authors predict the ratings by factorizing a $6040 \times 3706 \times 36$ tensor using BPTF method [26]. We may attribute the performance gain to the modeling flexibility of pairwise interaction tensor and the learning guarantees of our algorithm.



(a) (b)

Figure 2: Empirical results on the Movielens dataset. (a) Comparison of RMSE with different truncation levels. MC: Matrix completion algorithm. RPIT: Recovery algorithm for pairwise interaction tensor. (b) Rank of recovered matrix $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$. $r_1 = \mathrm{rank}(\mathbf{X}), r_2 = \mathrm{rank}(\mathbf{Y}), r_3 = \mathrm{rank}(\mathbf{Z})$.

5 Conclusion

In this paper, we proved rigorous guarantees for convex programs for recovery of pairwise interaction tensors with missing entries, both in the absence and in the presence of noise. We designed a scalable optimization algorithm for solving the convex programs. We supplemented our theoretical results with simulation experiments and a real-world application to movie recommendation. In the noiseless case, simulations showed that the exact recovery almost always succeeded if the number of observations is a constant time of the degree of freedom, which agrees asymptotically with the theoretical result. In the noisy case, the simulation results confirmed that the stable recovery algorithm is able to reliably recover pairwise interaction tensor from noisy observations. Our results on the temporal movie recommendation application demonstrated that, by incorporating the temporal information, our algorithm outperforms conventional matrix completion and achieves state-of-the-art results.

Acknowledgments

This work was fully supported by the Basic Research Program of Shenzhen (Project No. JCYJ20120619152419087 and JC201104220300A), and the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK 413212 and CUHK 415212).

References

[1] Evrim Acar, Daniel M Dunlavy, Tamara G Kolda, and Morten Mørup. Scalable tensor factorizations for incomplete data. *Chemometrics and Intelligent Laboratory Systems*, 106(1):41–56, 2011.

[2] M Berry et al. Svdpackc (version 1.0) user's guide, university of tennessee tech. *Report (393-194, 1993 (Revised October 1996).*, 1993.

[3] Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[4] Emmanuel J Candes and Yaniv Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.

[5] Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.

[6] A Evgeniou and Massimiliano Pontil. Multi-task feature learning. 2007.

[7] Maryam Fazel, Haitham Hindi, and Stephen P Boyd. A rank minimization heuristic with application to minimum order system approximation. In *American Control Conference, 2001*, 2001.

[8] David Gross, Yi-Kai Liu, Steven T Flammia, Stephen Becker, and Jens Eisert. Quantum state tomography via compressed sensing. *Physical review letters*, 105(15):150401, 2010.

[9] Johan Håstad. Tensor rank is np-complete. *Journal of Algorithms*, 11(4):644–654, 1990.

[10] Christopher Hillar and Lek-Heng Lim. Most tensor problems are np hard. *JACM*, 2013.

[11] Prateek Jain, Raghu Meka, and Inderjit Dhillon. Guaranteed rank minimization via singular value projection. In *NIPS*, 2010.

[12] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[13] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[14] Rasmus Munk Larsen. Propack-software for large and sparse svd calculations. *Available online.*, 2004.

[15] Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. Tensor completion for estimating missing values in visual data. In *ICCV*, 2009.

[16] Ian Porteous, Evgeniy Bart, and Max Welling. Multi-hdp: A non-parametric bayesian model for tensor factorization. In *AAAI*, 2008.

[17] Steffen Rendle, Leandro Balby Marinho, Alexandros Nanopoulos, and Lars Schmidt-Thieme. Learning optimal ranking with tensor factorization for tag recommendation. In *SIGKDD*, 2009.

[18] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *WWW*, 2010.

[19] Steffen Rendle and Lars Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *ICDM*, 2010.

[20] Amnon Shashua and Tamir Hazan. Non-negative tensor factorization with applications to statistics and computer vision. In *ICML*, 2005.

[21] Yue Shi, Alexandros Karatzoglou, Linas Baltrunas, Martha Larson, Alan Hanjalic, and Nuria Oliver. Tfmap: Optimizing map for top-n context-aware recommendation. In *SIGIR*, 2012.

[22] Nathan Srebro, Jason DM Rennie, and Tommi Jaakkola. Maximum-margin matrix factorization. *NIPS*, 2005.

[23] Ryota Tomioka, Kohei Hayashi, and Hisashi Kashima. Estimation of low-rank tensors via convex optimization. *arXiv preprint arXiv:1010.0789*, 2010.

[24] Ryota Tomioka, Taiji Suzuki, Kohei Hayashi, and Hisashi Kashima. Statistical performance of convex tensor decomposition. *NIPS*, 2011.

[25] Jason Weston, Chong Wang, Ron Weiss, and Adam Berenzweig. Latent collaborative retrieval. ICML, 2012.

[26] Liang Xiong, Xi Chen, Tzu-Kuo Huang, Jeff Schneider, and Jaime G Carbonell. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *SDM*, 2010.