Journal of Computational Mathematics
Vol.xx, No.x, 2025, 1–20.

http://www.global-sci.org/jcm
doi:10.4208/jcm.2508-m2024-0288

STOCHASTIC VARIANCE REDUCED GRADIENT FOR TENSOR RECOVERY*

Li Li
Shenzhen Key Laboratory of Advanced Machine Learning and Applications, School of Mathematical Sciences, Shenzhen University, Shenzhen 518060, China
Email: 2250201001@email.szu.edu.cn
Chen Xu and Jian Lu[1)]
Shenzhen Key Laboratory of Advanced Machine Learning and Applications, School of Mathematical Sciences, Shenzhen University, Shenzhen 518060, China
National Center for Applied Mathematics Shenzhen, Shenzhen 518055, China
Emails: chenxuszu@sina.com, jianlu@szu.edu.cn
Ningning Han
School of Mathematical Sciences, Tiangong University, Tianjin 300387, China
Email: ningninghan@tiangong.edu.cn
Lixin Shen
Department of Mathematics, Syracuse University, Syracuse 13244, USA
Email: lshen03@syr.edu

Abstract

Low-rank tensor recovery is pivotal in numerous applications, including image and video processing, machine learning, and data analysis. A common approach to this problem involves convex relaxation, where the tensor rank function is minimized by using the tensor nuclear norm. However, this method can be significantly suboptimal. In addition, the stochastic variance reduced gradient (SVRG) method, a variant of stochastic gradient descent, has been applied to matrix recovery problems.

In this paper, we extend the SVRG method to the tensor framework, introducing the tensor stochastic variance reduced gradient (TSVRG) algorithm for tensor recovery with CP or Tucker rank constraints. TSVRG is designed to achieve higher precision solutions by escaping local minima and identifying superior global optima. Moreover, TSVRG offers reduced computational complexity compared to traditional gradient descent methods. We establish a convergence theorem for TSVRG under the tensor restricted isometry condition when the measurements are linear. Finally, we present numerical results using both synthetic and real data, demonstrating the competitive performance of TSVRG compared to other advanced algorithms.

Mathematics subject classification: 15A69, 68U10, 68W20, 90C25.
Key words: Tensor recovery, Stochastic variance reduced gradient, CP rank, Tucker rank.

1. Introduction

Tensors, as generalizations of vectors and matrices, provide natural representations for massive multimode data sets encountered in many applications, including image and video processing [13, 18, 19, 24], recommendation system [26], signal processing [9], etc. These successful

* Received December 20, 2024 / Revised version received July 2, 2025 / Accepted August 5, 2025 / Published online December 5, 2025 /
1) Corresponding authors

applications are based on the rich structures carried by the tensors. The low-rankness of the tensor is one of the structures. The rank of a tensor can be defined in several ways, two popular ones are as follows:

- CANDECOMP/PARAFAC (CP) rank: The smallest number of rank-1 tensors that sum to form the given tensor. A rank-1 tensor is an outer product of vectors.
- Tucker rank: Generalization involving the rank of the core tensor in a Tucker decomposition.

These ranks derived from the decomposition of CP and Tucker [11] allow the extraction of latent factors from multidimensional data. This is useful in signal processing, data compression, and feature extraction in machine learning.

Tensor completion is one of the most actively studied problems in tensor-related research, yet it is diffusely presented in many different research domains. This dispersion is due to various factors, such as the inherent nature of multidimensional data sets that are often raw and incomplete due to unpredictable or unavoidable reasons such as maloperations, limited permissions, and missing data at random. Tensor completion aims to impute missing or unobserved entries of a partially observed tensor [15].

A low-rank hypothesis is often necessary to restrict the degrees of freedom of missing entries in a tensor. Given a low-rank tensor $\mathcal{T}$ with missing entries, the goal of completing it can be formulated as the following optimization problem:

$$\begin{aligned} &\min_{\mathcal{X}} \ \operatorname{rank}(\mathcal{X}) \\ &\text{s.t.} \quad \mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{T}), \end{aligned} \tag{1.1}$$

where $\mathcal{X}$ represents the completed low-rank tensor of $\mathcal{T}, \Omega$ is an index set denoting the indices of observations, and $\mathcal{P}_\Omega$ is the projection operator onto the set Ω.

Since calculating the tensor rank is an NP-hard problem, directly addressing the problem (1.1) is impractical. To circumvent this, some researchers consider the rank of the target tensor to be fixed and relax the problem (1.1) to the one that minimizes the difference between the observations and their predictions, as follows:

$$\begin{aligned} &\min_{\mathcal{X}} \ D\big(P_\Omega(\mathcal{X}), P_\Omega(\mathcal{T})\big) \\ &\text{s.t.} \quad \operatorname{rank}(\mathcal{X}) \le r, \end{aligned} \tag{1.2}$$

where D is an error measure between $P_\Omega(\mathcal{X})$ and $P_\Omega(\mathcal{T})$, which is often defined as the square of the Frobenius norm of their difference [14, 17]. As the projection operator P_Ω can be viewed as a special linear transform, a generalized tensor recovery model [16] is considered as follows:

$$\begin{aligned} &\min_{\mathcal{X}\in\mathbb{R}^{n_1\times n_2\times\cdots\times n_d}} \ \frac{1}{m}\|\boldsymbol{y} - \mathcal{A}(\mathcal{X})\|_F^2 \\ &\text{s.t.} \qquad\qquad\qquad\quad \operatorname{rank}(\mathcal{X}) \le r, \end{aligned} \tag{1.3}$$

where $\boldsymbol{y} \in \mathbb{R}^m$ denotes the observed data and $\mathcal{A}$ is a linear transform from $\mathbb{R}^{n_1\times n_2\times\cdots\times n_d}$ to $\mathbb{R}^m$.

The (stochastic) tensor iterative hard thresholding (StoTIHT/TIHT) methods [6, 7], based on (stochastic) gradient descent (SGD/GD), are commonly used for solving (1.3). However, the GD method is very time-consuming as it requires calculating the gradient over the entire sample set. In contrast, the computational complexity of SGD is significantly lower compared

to GD. Nevertheless, the stochastic gradient introduces bias and variance issues, which can lead to slow convergence and the potential for the algorithm to get stuck in local minima.

Combining the benefits of both SGD and traditional GD while mitigating their respective drawbacks is essential. The stochastic variance reduced gradient (SVRG) method, developed in [10], aims to achieve this balance. SVRG maintains the computational efficiency of SGD while retaining the accuracy and stability of GD. It does this by periodically recalculating a reference full gradient and using it to adjust the stochastic gradients. This adjustment involves a correction term derived from the difference between the current stochastic gradient and the gradient evaluated at the reference point. This correction significantly decreases the variance of the gradient estimates, leading to more stable updates and better convergence rates. The low computational cost per iteration, characteristic of SGD, combined with the reduced variance in gradient estimates, approaching the stability of GD, allows SVRG to achieve faster convergence compared to both SGD and GD. This makes SVRG particularly suitable for large-scale machine learning problems where both computational efficiency and convergence speed are crucial.

SVRG has been successfully applied to various structured prediction problems and neural network learning [10]. Additionally, several variants and enhanced versions of SVRG have been proposed: In [10], a generic framework based on a new SVRG descent method was proposed to accelerate the recovery of nonconvex low-rank matrices; An efficient stochastic variance reduction-gradient support pursuit algorithm and its accelerated version were presented in [22]; A variant of SVRG employing a trust-region-like scheme for selecting step sizes, which was proved to be linearly convergent in expectation for smooth strongly convex functions, and to enjoy a faster convergence rate than traditional SVRG methods, was developed in [27]; An efficient SVRG algorithm to solve the affine rank minimization problem, consisting of finding a matrix of minimum rank from linear measurements, was developed in [8]; and recently, a novel Riemannian extension of the Euclidean SVRG algorithm to a manifold search space was presented in [21]. These advancements highlight the flexibility and effectiveness of SVRG in various optimization contexts.

In this paper, we propose the tensor stochastic variance reduced gradient algorithm to recover tensor data with the CP or Tucker rank constraint. It promises to find a higher precision solution and reduce computational complexity under the guarantee of theory. Specifically, compared to SGD, SVRG achieves faster and more stable convergence by reducing the variance of gradient estimates. Furthermore, its combination of variance reduction and periodic full-gradient corrections enables SVRG to escape local minima more effectively than the noisy and unstable updates of standard SGD. Meanwhile, SVRG has smaller computational complexity compared with GD. We have also demonstrated a linearly convergent guarantee for our TSVRG algorithm under a tensor restricted isometry condition when measurements are linear. Furthermore, we focus on two types of tensor data in order to confirm our theoretical results. Specifically, we will examine tensors derived from synthetic data and video data. Finally, we give results of numerical experiments to demonstrate that TSVRG can indeed be effectively utilized for tensor recovery problem compared to other advanced methods.

The remainder of this paper are organized in the following order. In Section 2, we provide a concise overview of essential concepts and terminologies used in the tensor structure and introduce the TSVRG method. In addition, we demonstrate a linearly convergent guarantee for the proposed TSVRG method in Section 3. Section 4 presents all computational results with the CP or Tucker rank constraint that demonstrate the effectiveness of the algorithm using both synthetic and real data. Finally, we summarize this work and discuss future work in Section 5.

2. Tensor SVRG (TSVRG) Algorithm

In this section, we will present a tensor stochastic variance reduced gradient algorithm to solve the model (1.3). To set the stage, we first provide some necessary concepts and definitions related to tensors.

In this paper, we denote scalars by lowercase letters (e.g. $x \in \mathbb{R}$), vectors by bold lowercase letters (e.g. $\boldsymbol{x} \in \mathbb{R}^{n_1}$), matrices by bold capital letters (e.g. $\mathsf{X} \in \mathbb{R}^{n_1 \times n_2}$), and tensors by calligraphic letters (e.g. $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$). The order of a tensor refers to the number of dimensions it has. Mathematically, if a tensor $\mathcal{X}$ has order d, it can be represented as $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$, where $n_1, n_2, \ldots, n_d$ are the sizes of the tensor along each dimension. We denote the i_1-th entry of a vector $\boldsymbol{x}$ by $\boldsymbol{x}(i_1)$, the (i_1, i_2)-th entry of a matrix X by $\mathsf{X}(i_1, i_2)$, and the $(i_1, i_2, \ldots, i_d)$-th entry of a d-order tensor $\mathcal{X}$ by $\mathcal{X}(i_1, i_2, \ldots, i_d)$. The expression of $[n]$ refers to the set of $\{1, 2, \ldots, n\}$.

We use the term "mode" to describe operations on a specific dimension of a tensor (e.g. mode-n product). Tensor matricization refers to the unfolding of a tensor in a matrix format with a predefined ordering of its modes. The most commonly used tensor matricization is mode-i matricization (also known as mode-i unfolding), which unfolds a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ along its i-th mode into a matrix. This matrix is denoted as $\mathcal{X}^{(i)}$ and has a size of $n_i \times \prod_{j \neq i} n_j$. This process is crucial for various tensor operations and algorithms, as it allows us to manipulate tensors in a more manageable two-dimensional matrix form.

For d column vectors $\boldsymbol{a}_1 \in \mathbb{R}^{n_1}, \ldots, \boldsymbol{a}_d \in \mathbb{R}^{n_d}$, the outer product among them is defined as

$$\mathcal{A} = \boldsymbol{a}_1 \circ \cdots \circ \boldsymbol{a}_d,$$

where $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ and $\mathcal{A}(i_1, i_2, \ldots, i_d) = \boldsymbol{a}_1(i_1) \cdots \boldsymbol{a}_d(i_d)$. Using the concept of the outer product, the canonical polyadic decomposition (CPD) of a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ expresses it as a sum of weighted outer products

$$\mathcal{X} = \sum_{r=1}^{R} \lambda_r \boldsymbol{u}_r^{(1)} \circ \boldsymbol{u}_r^{(2)} \circ \cdots \circ \boldsymbol{u}_r^{(d)},$$

where $\lambda_r > 0, \boldsymbol{u}_r^{(k)}$ is a unit vector in $\mathbb{R}^{n_k}$ and R is a positive integer denoting the rank of the decomposition. The smallest number R for which this representation holds is called the CP rank of the tensor $\mathcal{X}$.

The mode-k product of a tensor with a matrix is an essential operation in tensor algebra, often used in tensor decomposition like Tucker decomposition. The mode-k product of a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ with a matrix $\mathsf{A} \in \mathbb{R}^{J \times n_k}$ is denoted as $\mathcal{X} \times_k \mathsf{A}$ and results in a tensor $\mathcal{Y} \in \mathbb{R}^{n_1 \times \cdots \times n_{k-1} \times J \times n_{k+1} \times \cdots \times n_d}$. Mathematically, the element of $\mathcal{Y}$ is computed as

$$\mathcal{Y}(i_1, \ldots, i_{k-1}, j, i_{k+1}, \ldots, i_d) = \sum_{i_k=1}^{n_k} \mathcal{X}(i_1, \ldots, i_{k-1}, i_k, i_{k+1}, \ldots, i_d) \mathsf{A}(j, i_k).$$

In terms of mode-k unfolding notation, $\mathcal{Y}^{(k)} = \mathsf{A}\mathcal{X}^{(k)}$. The operation of mode-$k$ product tensor $\mathcal{X}$ with a matrix A can be seen as reshaping the tensor $\mathcal{X}$ into a matrix, multiplying it by A, and then reshaping it back into a tensor. The Tucker rank of a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ is a tuple $(r_1, r_2, \ldots, r_d)$, where r_k is the rank of the mode-k unfolding of the tensor [11]. By higher-order singular value decomposition (HOSVD) algorithm, the Tucker decomposition of $\mathcal{X}$ expresses this tensor as a core tensor multiplied by a matrix along each mode

$$\mathcal{X} = \mathcal{G} \times_1 \mathsf{U}_1 \cdots \times_d \mathsf{U}_d,$$

where $\mathsf{U}_k \in \mathbb{R}^{n_k \times r_k}$ are factor matrices and $\mathcal{G}$ is called the core tensor of $\mathcal{X}$.

The inner product of $\mathcal{X}$ and $\mathcal{Y}$ in $\mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ is defined as

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \cdots \sum_{i_d=1}^{n_d} \mathcal{X}(i_1, i_2, \ldots, i_d)\mathcal{Y}(i_1, i_2, \ldots, i_d).$$

The Frobenius norm of $\mathcal{X}$ is then defined as $\|\mathcal{X}\|_F := \langle \mathcal{X}, \mathcal{X} \rangle^{1/2}$. These definitions extend the familiar concepts of inner product and norm from vectors and matrices to tensors, allowing for a consistent framework for analyzing and manipulating higher-dimensional data structures.

P_Γ represents the corresponding sampling operator that retrieves solely the entries indexed by Γ

$$\big(\mathcal{P}_\Gamma(\mathcal{X})\big) = \begin{cases} \mathcal{X}(i_1, i_2, \cdots, i_d), & \text{if} \quad (i_1, i_2, \cdots, i_d) \in \Gamma, \\ 0, & \text{if} \quad (i_1, i_2, \cdots, i_d) \in \Gamma^c. \end{cases}$$

Here, Γ denotes a binary indicator tensor of the same shape as $\mathcal{X}$, where each entry is 1 if observed and 0 otherwise. The complement of Γ is denoted by Γ^c. Moreover, the projection operators satisfy $\mathcal{P}_\Gamma(\mathcal{X}) + \mathcal{P}_{\Gamma^c}(\mathcal{X}) = \mathcal{X}$.

In the remainder of this section, we will present the SVRG algorithm for the model (1.3). Let $\mathcal{A}$ be a linear transform from $\mathbb{R}^{n_1 \times \cdots \times n_d}$ to $\mathbb{R}^m$. There exist $\mathcal{A}_i \in \mathbb{R}^{n_1 \times \cdots \times n_d}, i \in \{1, \ldots, m\}$ such that $\boldsymbol{y} = \mathcal{A}(\mathcal{X})$ for $\mathcal{X} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ implies

$$\boldsymbol{y}(i) = \mathcal{A}_i(\mathcal{X}) = \langle \mathcal{A}_i, \mathcal{X} \rangle.$$

Given an integer M, let $S_i, i = 1, \ldots, M$ be a partition of the set $[m]$, that is, $\cup_{i=1}^M S_i = [m]$ and $S_i \cap S_j = \emptyset$ for $i \neq j$. Based on this partition and a given linear transform $\mathcal{A}$ from $\mathbb{R}^{n_1 \times \cdots \times n_d}$ to $\mathbb{R}^m$, we define linear transforms $\mathcal{A}_{S_i}$ from $\mathbb{R}^{n_1 \times \cdots \times n_d}$ to $\mathbb{R}^{|S_i|}$ and function f_i from $\mathbb{R}^{n_1 \times \cdots \times n_d}$ to $\mathbb{R}$, respectively, as

$$\mathcal{A}_{S_i}(\mathcal{X}) := \big(\mathcal{A}_j(\mathcal{X}) : j \in S_i\big), \tag{2.1}$$

$$f_i(\mathcal{X}) := \|\boldsymbol{y}_{S_i} - \mathcal{A}_{S_i}(\mathcal{X})\|_2^2 = \sum_{j \in S_i} \big(\boldsymbol{y}(j) - \langle \mathcal{A}_j, \mathcal{X} \rangle\big)^2. \tag{2.2}$$

Then, the cost function of the model (1.3) can be rewritten as

$$F(\mathcal{X}) := \frac{1}{m}\|\boldsymbol{y} - \mathcal{A}(\mathcal{X})\|_F^2 = \frac{1}{m}\sum_{i=1}^M f_i(\mathcal{X}). \tag{2.3}$$

Accordingly, the model (1.3) becomes

$$\begin{aligned} \min_{\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}} \quad & F(\mathcal{X}) \\ \text{s.t.} \quad & \operatorname{rank}(\mathcal{X}) \leq r. \end{aligned} \tag{2.4}$$

We propose the tensor stochastic variance reduced gradient algorithm, detailed in Algorithm 2.1. This algorithm operates on K cycles. In each cycle k, an iterate $\widetilde{\mathcal{X}}_k$ is used to compute the gradient $\nabla F(\widetilde{\mathcal{X}}_k)$. After initialization $\mathcal{X}_0 \leftarrow \widetilde{\mathcal{X}}_k$, a set of n inner iterations, indexed by t, with an update $\widetilde{\mathcal{X}}_{k+1} \leftarrow \mathcal{X}_n$ is performed, where

$$\mathcal{X}_{t+1} = \mathcal{H}_r\big(\mathcal{X}_t - \eta\big(\nabla f_{l_t}(\mathcal{X}_t) - \nabla f_{l_t}(\widetilde{\mathcal{X}}_k) + \nabla F(\widetilde{\mathcal{X}}_k)\big)\big), \tag{2.5}$$

Algorithm 2.1: Tensor Stochastic Variance Reduced Gradient for Model (1.3).

Input: The given tensor CP or Tucker rank r, the maximum number of outer loops K, the maximum number of inner loops n, the stepsize η.
Initialization: $\widetilde{\mathcal{X}}_0$.
for $k = 0, 1, 2, \ldots, K-1$ **do**
  Compute the gradient $\nabla F(\widetilde{\mathcal{X}}_k)$.
  $\mathcal{X}_0 \leftarrow \widetilde{\mathcal{X}}_k$.
  for $t = 0, 1, 2, \ldots, n-1$ **do**
    Randomly select $l_t \in \{1, \ldots, M\}$.
    $\mathcal{X}_{t+1} = \mathcal{H}_r(\mathcal{X}_t - \eta(\nabla f_{l_t}(\mathcal{X}_t) - \nabla f_{l_t}(\widetilde{\mathcal{X}}_k) + \nabla F(\widetilde{\mathcal{X}}_k)))$.
  end
  $\widetilde{\mathcal{X}}_{k+1} = \mathcal{X}_n$.
  Exit if a stopping criterion is satisfied.
end
Output: $\breve{\mathcal{X}} = \widetilde{\mathcal{X}}_k$.

and $l_t \in \{1, \ldots, M\}$ is chosen randomly. Since the expected value of $\nabla f_{l_t}(\widetilde{\mathcal{X}}_k)$ over all possible $l_t \in \{1, \ldots, M\}$ is equal to $\nabla F(\widetilde{\mathcal{X}}_k), \nabla f_{l_t}(\widetilde{\mathcal{X}}_k) - \nabla F(\widetilde{\mathcal{X}}_k)$ can be seen as the bias in the gradient estimate $\nabla f_{l_t}(\widetilde{\mathcal{X}}_k)$. Thus, in every inner iteration, the algorithm randomly selects a stochastic gradient $\nabla f_{l_t}(\mathcal{X}_t)$ evaluated at $\mathcal{X}_t$ and corrects it on a perceived bias. In general, $\nabla f_{l_t}(\mathcal{X}_t) - \nabla f_{l_t}(\widetilde{\mathcal{X}}_k) + \nabla F(\widetilde{\mathcal{X}}_k)$ represents an unbiased estimator of $\nabla F(\mathcal{X}_t)$, but with a variance expected to be smaller than if one were simply using $\nabla f_{l_t}(\mathcal{X}_t)$. This variance reduction is why the method is referred to as the SVRG method [10].

In addition, note that the t-th iteration does not update directly to the $(t+1)$-th iteration. Similarly to TIHT [7] with CP rank and StoTIHT [6] with Tucker rank, a more detailed process from the t-th iteration to the $(t+1)$-th iteration involves projecting onto the constraint space $\mathcal{M}_r$ employing a hard thresholding operator $\mathcal{H}_r(\cdot)$, where

$$\mathcal{M}_r = \{\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d} \,:\, \mathrm{rank}(\mathcal{X}) \leq r\}.$$

The operator $\mathcal{H}_r(\mathcal{X})$ denotes the hard thresholding operator which calculates a best rank-r approximation $\mathcal{X}^{best}$ of a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$.

- If r is given the Tucker rank $(r_1, r_2 \cdots, r_d)$, we use HOSVD to get $\mathcal{X}^{best}$ [5], that is,

$$\mathcal{H}_r(\mathcal{X}) = \mathcal{H}_r(\mathcal{G}) \times_1 \mathsf{U}_1 \cdots \times_d \mathsf{U}_d,$$

  where $\mathcal{H}_r(\mathcal{G})(i_1, i_2, \cdots, i_d) = \mathcal{G}(i_1, i_2, \cdots, i_d), i_1 \leq r_1, i_2 \leq r_2, \cdots, i_d \leq r_d$. And $\mathcal{H}_r(\mathcal{G})(i_1, i_2, \cdots, i_d) = 0$, otherwise, and $\mathcal{G}(i_1, i_2, \cdots, i_d)$ denotes the $(i_1, i_2, \cdots, i_d)$-th entry element of the core tensor $\mathcal{G}$.

- If r is given the CP rank $R, \mathcal{H}_r(\mathcal{X})$ uses CPD to get $\mathcal{X}^{best}$ [7], i.e.

$$\mathcal{H}_r(\mathcal{X}) = \sum_{i=1}^{R} \mathcal{H}_r(\lambda_i) \boldsymbol{u}_i^{(1)} \circ \boldsymbol{u}_i^{(2)} \circ \cdots \circ \boldsymbol{u}_i^{(d)},$$

  where $\mathcal{H}_r(\lambda_i) = \lambda_i, i \leq R, \mathcal{H}_r(\lambda_i) = 0, i > R$.

We also assume that

$$\|\mathcal{H}_r(\widehat{\mathcal{X}}_t) - \widehat{\mathcal{X}}_t\|_F \leq \theta_0 \big\|\widehat{\mathcal{X}}_t^{best} - \widehat{\mathcal{X}}_t\big\|_F \tag{2.6}$$

holds for all $t = 1, 2, \ldots, n$ with some $\theta_0 \in [1, \infty)$ as same as [20]. Here

$$\widehat{\mathcal{X}}_t = \mathcal{X}_t - \eta\big(\nabla f_{l_t}(\mathcal{X}_t) - \nabla f_{l_t}(\widetilde{\mathcal{X}}_k) + \nabla F(\widetilde{\mathcal{X}}_k)\big).$$

Denote

$$\widehat{\mathcal{X}}_t^{best} = \mathrm{argmin}_{\mathrm{rank}(\mathcal{X})\leq r}\|\widehat{\mathcal{X}}_t - \mathcal{X}\|_F$$

as the best rank-r approximation of $\widehat{\mathcal{X}}_t$ corresponding with CP or Tucker decomposition of tensor (given by the CPD or HOSVD). The different tensor ranks correspond to different parameters θ_0 [5]. In this work, we will also presume that such an approximation $\widehat{\mathcal{X}}_t^{best}$ exists. The TSVRG algorithm for solving problem (2.4) is given in Algorithm 2.1.

3. Linear Convergence Analysis for TSVRG

In this section, we provide a theoretical analysis on the linear convergence of the TSVRG algorithm. We begin with the concept of the tensor restricted isometry property (TRIP) introduced in [4, 6]. In the following, "rank" denotes CP or Tucker rank.

Definition 3.1. *Let $\mathcal{A} : \mathbb{R}^{n_1\times\cdots\times n_d} \to \mathbb{R}^m$ and $\mathcal{A}_{S_i} : \mathbb{R}^{n_1\times\cdots\times n_d} \to \mathbb{R}^{|S_i|}$ be given by (2.1) for an associated partition S_i, $i \in [M]$, $\cup_{i=1}^M S_i = [m]$ and $S_i \cap S_j = \emptyset$ for $i \neq j$. We say $\mathcal{A}$ and $\mathcal{A}_{S_i}$ meet the TRIP if there exists a tensor restricted isometry constant $\delta_{\boldsymbol{r}} \in (0, 1)$ such that*

$$(1 - \delta_{\boldsymbol{r}})\|\mathcal{X}\|_F^2 \leq \frac{1}{m}\|\mathcal{A}(\mathcal{X})\|_2^2, \quad \|\mathcal{A}_{S_i}(\mathcal{X})\|_2^2 \leq (1 + \delta_{\boldsymbol{r}})\|\mathcal{X}\|_F^2 \tag{3.1}$$

hold for all tensors $\mathcal{X} \in \mathbb{R}^{n_1\times\cdots\times n_d}$ with their rank at most $\boldsymbol{r}$.

We remark that the TRIP automatically implies that

$$\frac{1}{m}\|\mathcal{A}(\mathcal{X})\|_2^2 \leq (1 + \delta_{\boldsymbol{r}})\|\mathcal{X}\|_F^2. \tag{3.2}$$

This inequality holds because

$$\frac{1}{m}\|\mathcal{A}(\mathcal{X})\|_2^2 = \frac{1}{m}\sum_{i=1}^M \|\mathcal{A}_{S_i}(\mathcal{X})\|_2^2 \leq \frac{1}{m}\sum_{i=1}^M (1 + \delta_r)\|\mathcal{X}\|_F^2.$$

In the following discussion, we always assume that the linear mappings $\mathcal{A}$ and $\mathcal{A}_{S_i}$ satisfy TRIP with constant $\delta_{3\boldsymbol{r}}$ and the functions F and f_i defined in (2.3) and (2.2), respectively. By a direct computation, the gradients of F and f_i defined in (2.3) and (2.2) are

$$\begin{aligned} \nabla F(\mathcal{X}) &= \frac{2}{m}\sum_{i=1}^m \mathcal{A}_i\big(\langle\mathcal{A}_i, \mathcal{X}\rangle - \boldsymbol{y}(i)\big), \\ \nabla f_i(\mathcal{X}) &= 2\sum_{j\in S_i} \mathcal{A}_j\big(\langle\mathcal{A}_j, \mathcal{X}\rangle - \boldsymbol{y}(j)\big), \end{aligned} \tag{3.3}$$

respectively. Obviously,

$$\langle\mathcal{X} - \mathcal{Y}, \nabla F(\mathcal{X}) - \nabla F(\mathcal{Y})\rangle = \frac{2}{m}\left\langle \mathcal{X} - \mathcal{Y}, \sum_{i=1}^m \mathcal{A}_i\langle\mathcal{A}_i, \mathcal{X} - \mathcal{Y}\rangle \right\rangle = \frac{2}{m}\|\mathcal{A}(\mathcal{X} - \mathcal{Y})\|_2^2, \tag{3.4}$$

and similarly

$$\langle\mathcal{X} - \mathcal{Y}, \nabla f_i(\mathcal{X}) - \nabla f_i(\mathcal{Y})\rangle = 2\|\mathcal{A}_i(\mathcal{X} - \mathcal{Y})\|_2^2. \tag{3.5}$$

Lemma 3.1. *For any tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ with their rank at most $\boldsymbol{r}$, the following inequalities hold:*

$$\langle \mathcal{X} - \mathcal{Y}, \nabla F(\mathcal{X}) - \nabla F(\mathcal{Y}) \rangle \geq \delta^-_{3\boldsymbol{r}} \|\mathcal{X} - \mathcal{Y}\|_F^2, \tag{3.6}$$

$$\langle \nabla f_i(\mathcal{X}) - \nabla f_i(\mathcal{Y}), \mathcal{X} - \mathcal{Y} \rangle \leq \delta^+_{3\boldsymbol{r}} \|\mathcal{X} - \mathcal{Y}\|_F^2, \tag{3.7}$$

and

$$f_i(\mathcal{Y}) \leq f_i(\mathcal{X}) + \langle \nabla f_i(\mathcal{X}), \mathcal{Y} - \mathcal{X} \rangle + \frac{\delta^+_{3r}}{2} \|\mathcal{X} - \mathcal{Y}\|^2, \tag{3.8}$$

where $\delta^-_{3\boldsymbol{r}} = 2(1 - \delta_{3\boldsymbol{r}})$ and $\delta^+_{3\boldsymbol{r}} = 2(1 + \delta_{3\boldsymbol{r}})$.

Proof. Eq. (3.6) is a direct consequence of Eq. (3.4) and the TRIP condition, while Eq. (3.7) is a direct consequence of Eq. (3.5) and the TRIP condition. To prove that (3.8) works. By (2.2),

$$\begin{aligned} f_i(\mathcal{Y}) &= \sum_{j \in S_i} \big(\boldsymbol{y}(j) - \langle \mathcal{A}_j, \mathcal{X} \rangle - \langle \mathcal{A}_j, \mathcal{Y} - \mathcal{X} \rangle\big)^2 \\ &= \sum_{j \in S_i} \Big(\big(\boldsymbol{y}(j) - \langle \mathcal{A}_j, \mathcal{X} \rangle\big)^2 - 2\big(\boldsymbol{y}(j) - \langle \mathcal{A}_j, \mathcal{X} \rangle\big)\langle \mathcal{A}_j, \mathcal{Y} - \mathcal{X} \rangle + \langle \mathcal{A}_j, \mathcal{Y} - \mathcal{X} \rangle^2 \Big) \\ &= f_i(\mathcal{X}) + \langle \nabla f_i(\mathcal{X}), \mathcal{Y} - \mathcal{X} \rangle + \|\mathcal{A}_{S_i}(\mathcal{Y} - \mathcal{X})\|_2^2. \end{aligned} \tag{3.9}$$

In the last equation above, the first term is from the definition of f_i, the second term is from Eq. (3.3), and the last is simply the ℓ_2 norm. Clearly, Eq. (3.8) is the direct consequence of the above equation and the TRIP condition. □

Lemma 3.2. *For given $\mathcal{X}$ and $\mathcal{Y}$ in $\mathbb{R}^{n_1 \times \cdots \times n_d}$ with ranks less than $\boldsymbol{r}$, let Γ be a space that contains the* $\operatorname{span}(\mathcal{X},\ \mathcal{Y})$ *spanned by tensors $\mathcal{X}$, $\mathcal{Y}$. Let $\mathcal{P}_\Gamma : \mathbb{R}^{n_1 \times \cdots \times n_d} \to \Gamma$ be the orthogonal projection onto Γ. Then, for all $i \in [M]$,*

$$\big\|\mathcal{P}_\Gamma\big(\nabla f_i(\mathcal{X}) - \nabla f_i(\mathcal{Y})\big)\big\|_F^2 \leq \delta^+_{3r} \langle \mathcal{X} - \mathcal{Y}, \nabla f_i(\mathcal{X}) - \nabla f_i(\mathcal{Y}) \rangle, \tag{3.10}$$

$$\big\|\mathcal{P}_\Gamma\big(\nabla F(\mathcal{X}) - \nabla F(\mathcal{Y})\big)\big\|_F^2 \leq \delta^+_{3r} \langle \mathcal{X} - \mathcal{Y}, \nabla F(\mathcal{X}) - \nabla F(\mathcal{Y}) \rangle \tag{3.11}$$

hold with $\delta^+_{3r} = 2(1 + \delta_{3r})$.

Proof. Define $h_i : \mathbb{R}^{n_1 \times \cdots \times n_d} \to \mathbb{R}$ as follows:

$$h_i(\mathcal{Z}) := f_i(\mathcal{Z}) - f_i(\mathcal{X}) - \langle \nabla f_i(\mathcal{X}), \mathcal{Z} - \mathcal{X} \rangle.$$

By definition, $h_i(\mathcal{X}) = 0$ and $h_i(\mathcal{Z}) = \|\mathcal{A}_{S_i}(\mathcal{Z} - \mathcal{X})\|_2^2 \geq 0$.

For any $\mathcal{U}$ and $\mathcal{V}$ in Γ, we know that the ranks of $\mathcal{U}, \mathcal{V}$, and their linear combinations are less than $2\boldsymbol{r}$. Hence, using (3.8), we have

$$h_i(\mathcal{U}) \leq h_i(\mathcal{V}) + \langle \nabla f_i(\mathcal{V}) - \nabla f_i(\mathcal{X}), \mathcal{U} - \mathcal{V} \rangle + (1 + \delta_{3\boldsymbol{r}}) \|\mathcal{U} - \mathcal{V}\|_F^2.$$

In particular, choosing

$$\mathcal{V} = \mathcal{Y}, \quad \mathcal{U} = \mathcal{Y} - \frac{1}{2(1 + \delta_{3\boldsymbol{r}})} \mathcal{P}_\Gamma\big(\nabla h_i(\mathcal{Y})\big),$$

both in Γ, we have

$$0 \le h_i(\mathcal{Y}) - \frac{1}{2(1+\delta_{3\boldsymbol{r}})}\big\langle \nabla f_i(\mathcal{Y}) - \nabla f_i(\mathcal{X}), \mathcal{P}_\Gamma\big(\nabla f_i(\mathcal{Y}) - \nabla f_i(\mathcal{X})\big)\big\rangle + \frac{1}{4(1+\delta_{3\boldsymbol{r}})}\big\|\mathcal{P}_\Gamma\big(\nabla f_i(\mathcal{X}) - \nabla f_i(\mathcal{Y})\big)\big\|_F^2.$$

Since

$$\big\langle \nabla f_i(\mathcal{Y}) - \nabla f_i(\mathcal{X}), \mathcal{P}_\Gamma\big(\nabla f_i(\mathcal{Y}) - \nabla f_i(\mathcal{X})\big)\big\rangle = \big\|\mathcal{P}_\Gamma\big(\nabla f_i(\mathcal{X}) - \nabla f_i(\mathcal{Y})\big)\big\|_F^2,$$

we therefore have

$$\frac{1}{4(1+\delta_{3\boldsymbol{r}})}\big\|\mathcal{P}_\Gamma\big(\nabla f_i(\mathcal{X}) - \nabla f_i(\mathcal{Y})\big)\big\|_F^2 \le h_i(\mathcal{Y}) = f_i(\mathcal{Y}) - f_i(\mathcal{X}) - \langle \nabla f_i(\mathcal{X}), \mathcal{Y} - \mathcal{X}\rangle.$$

By exchange the roles of $\mathcal{X}$ and $\mathcal{Y}$ in the above inequality, we have

$$\frac{1}{4(1+\delta_{3\boldsymbol{r}})}\big\|\mathcal{P}_\Gamma\big(\nabla f_i(\mathcal{X}) - \nabla f_i(\mathcal{Y})\big)\big\|_F^2 \le f_i(\mathcal{X}) - f_i(\mathcal{Y}) - \langle \nabla f_i(\mathcal{Y}), \mathcal{X} - \mathcal{Y}\rangle.$$

Adding these two inequalities leads to the desired inequality (3.10).

Similarly, since (3.2) holds for the map $\mathcal{A}$, the inequality (3.11) also holds. □

Lemma 3.3. *For given $\mathcal{X}$ and $\mathcal{Y}$ in $\mathbb{R}^{n_1\times\cdots\times n_d}$ with ranks less than $\boldsymbol{r}$, let Γ be a space that contains the* $\operatorname{span}(\mathcal{X},\mathcal{Y})$ *spanned by tensors $\mathcal{X},\mathcal{Y}$. Let $\mathcal{P}_\Gamma : \mathbb{R}^{n_1\times\cdots\times n_d} \to \Gamma$ be the orthogonal projection onto Γ. Then, for all $i \in [M]$,*

$$\big\|\mathcal{X} - \mathcal{Y} - \eta\mathcal{P}_\Gamma\big(\nabla F(\mathcal{X}) - \nabla F(\mathcal{Y})\big)\big\|_F \le \rho_{3r}\|\mathcal{X} - \mathcal{Y}\|_F, \tag{3.12}$$

$$\mathbb{E}_i\big\|\mathcal{X} - \mathcal{Y} - \eta\mathcal{P}_\Gamma\big(\nabla f_i(\mathcal{X}) - \nabla f_i(\mathcal{Y})\big)\big\|_F \le \rho_{3r}\|\mathcal{X} - \mathcal{Y}\|_F \tag{3.13}$$

hold with

$$\rho_{3r} = \sqrt{1 + \eta^2\delta_{3r}^+\delta_{3r}^- - 2\eta\delta_{3r}^-}, \quad \delta_{3r}^- = 2(1-\delta_{3r}), \quad \delta_{3r}^+ = 2(1+\delta_{3r}),$$

and the stepsize $\eta \le 2/\delta_{3\boldsymbol{r}}^+$.

Here, the mathematical expectation is denoted by $\mathbb{E}$ to reflect the mean value of a given random variable in the sense of probability.

Proof. We first prove inequality (3.12).

$$\begin{aligned}&\big\|\mathcal{X} - \mathcal{Y} - \eta\mathcal{P}_\Gamma\big(\nabla F(\mathcal{X}) - \nabla F(\mathcal{Y})\big)\big\|_F^2\\ &= \|\mathcal{X} - \mathcal{Y}\|_F^2 + \eta^2\big\|\mathcal{P}_\Gamma\big(\nabla F(\mathcal{X}) - \nabla F(\mathcal{Y})\big)\big\|_F^2\\ &\quad - 2\eta\big\langle \mathcal{X} - \mathcal{Y}, \mathcal{P}_\Gamma\big(\nabla F(\mathcal{X}) - \nabla F(\mathcal{Y})\big)\big\rangle.\end{aligned}$$

By inequality (3.11), we have

$$\begin{aligned}&\big\|\mathcal{X} - \mathcal{Y} - \eta\mathcal{P}_\Gamma\big(\nabla F(\mathcal{X}) - \nabla F(\mathcal{Y})\big)\big\|_F^2\\ &\le \|\mathcal{X} - \mathcal{Y}\|_F^2 - \big(2\eta - \eta^2\delta_{3r}^+\big)\langle \mathcal{X} - \mathcal{Y}, \nabla F(\mathcal{X}) - \nabla F(\mathcal{Y})\rangle\\ &\le \|\mathcal{X} - \mathcal{Y}\|_F^2 - \delta_{3r}^-\big(2\eta - \eta^2\delta_{3r}^+\big)\|\mathcal{X} - \mathcal{Y}\|_F^2,\end{aligned}$$

where the last inequality follows from (3.6), with $\eta \le 2/\delta_{3\boldsymbol{r}}^+$.

Next, we prove the inequality (3.13). From the above discussion, we have

$$\begin{aligned}&\big\|\mathcal{X}-\mathcal{Y}-\eta\mathcal{P}_\Gamma\big(\nabla f_i(\mathcal{X})-\nabla f_i(\mathcal{Y})\big)\big\|_F^2\\ &\le \|\mathcal{X}-\mathcal{Y}\|_F^2-\big(2\eta-\eta^2\delta_{3r}^+\big)\langle\mathcal{X}-\mathcal{Y},\nabla f_i(\mathcal{X})-\nabla f_i(\mathcal{Y})\rangle.\end{aligned}$$

Taking the expectation on the above equation and using the fact of $\mathbb{E}_i\nabla f_i(\mathcal{X})=\nabla F(\mathcal{X})$, we obtain

$$\begin{aligned}&\mathbb{E}_i\big\|\mathcal{X}-\mathcal{Y}-\eta\mathcal{P}_\Gamma\big(\nabla f_i(\mathcal{X})-\nabla f_i(\mathcal{Y})\big)\big\|_F^2\\ &\le \|\mathcal{X}-\mathcal{Y}\|_F^2-\big(2\eta-\eta^2\delta_{3r}^+\big)\langle\mathcal{X}-\mathcal{Y},\nabla F(\mathcal{X})-\nabla F(\mathcal{Y})\rangle.\end{aligned}$$

Using (3.6) with $\eta\le 2/\delta_{3\boldsymbol{r}}^+$, we further deduce

$$\mathbb{E}_i\big\|\mathcal{X}-\mathcal{Y}-\eta\mathcal{P}_\Gamma\big(\nabla f_i(\mathcal{X})-\nabla f_i(\mathcal{Y})\big)\big\|_F^2\le\rho_{3r}^2\|\mathcal{X}-\mathcal{Y}\|_F^2,\quad \rho_{3r}^2=1+\eta^2\delta_{3r}^+\delta_{3r}^--2\eta\delta_{3r}^-.$$

Finally, the desired result for (3.13) follows from Jensen inequality, $(\mathbb{E}Z)^2\le\mathbb{E}(Z)^2$. □

Theorem 3.1. *Assume that the operators*

$$\mathcal{A}:\mathbb{R}^{n_1\times n_2\times\cdots\times n_d}\ \to\ \mathbb{R}^m,\quad \mathcal{A}_{l_i}:\mathbb{R}^{n_1\times n_2\times\cdots\times n_d}\ \to\ \mathbb{R}^l,$$

used in the generation of linear measurements, satisfy the TRIP defined in Definition 3.1 *with parameter* $\delta_{3r}\le 1/71$. *Furthermore, assume that the rank-r approximation operator* $\mathcal{H}_r(\cdot)$ *satisfies* (2.6) *for all* $t=0,1,\ldots,n-1$ *with some* $\theta\in[1,\infty)$. *Let* $\mathcal{X}^*$ *denote the optimal solution of* (2.4) *and let* $\mathcal{X}_0$ *denote the initial tensor. Then at the* t*-th iteration in Algorithm* 2.1, *TSVRG converges linearly in expectation, as follows:*

$$\mathbb{E}_{l_t}\|\widetilde{\mathcal{X}}_k-\mathcal{X}^*\|_F\le\kappa^k\|\widetilde{\mathcal{X}}_0-\mathcal{X}^*\|_F,$$

where the step size η *satisfies*

$$\frac{3\delta_{3r}^--\omega}{3\delta_{3r}^+\delta_{3r}^-}<\eta<\frac{3\delta_{3r}^-+\omega}{3\delta_{3r}^+\delta_{3r}^-}$$

with

$$\begin{aligned}&\kappa=\frac{-3\theta^{n+1}+\theta^n+2\theta}{1-\theta}<1,\quad \omega=\sqrt{71\delta_{3r}^2-72\delta_{3r}+1},\\ &\theta=2\rho_{3r}=2\sqrt{1+\eta^2\delta_{3r}^+\delta_{3r}^--2\eta\delta_{3r}^-},\quad \delta_{3r}^-=2(1-\delta_{3r}),\quad \delta_{3r}^+=2(1+\delta_{3r}).\end{aligned}$$

Proof. For a given k and the initial tensor $\mathcal{X}_0$, denote

$$\tilde{\mathcal{X}}_{t,k}=\mathcal{X}_t-\eta\big(\nabla f_{l_t}(\mathcal{X}_t)-\nabla f_{l_t}(\widetilde{\mathcal{X}}_k)+\nabla F(\widetilde{\mathcal{X}}_k)\big).$$

We then generate a sequence $\{\mathcal{X}_t\}_{t=1}^n$ via

$$\mathcal{X}_{t+1}=\mathcal{H}_r(\tilde{\mathcal{X}}_{t,k}).$$

By the Eckart-Young theorem, which ensures that $\mathcal{X}_{t+1}$ is the nearest rank-r approximation to $\tilde{\mathcal{X}}_{t,k}$ in the Frobenius norm, we have

$$\|\mathcal{X}_{t+1}-\tilde{\mathcal{X}}_{t,k}\|_F^2\le\|\mathcal{X}^*-\tilde{\mathcal{X}}_{t,k}\|_F^2.$$

Together with the identity

$$\|\mathcal{X}_{t+1}-\mathcal{X}^*\|_F^2 = \|\mathcal{X}_{t+1}-\tilde{\mathcal{X}}_{t,k}\|_F^2 - \|\mathcal{X}^*-\tilde{\mathcal{X}}_{t,k}\|_F^2 - 2\langle \mathcal{X}_{t+1}-\mathcal{X}^*, \mathcal{X}^*-\tilde{\mathcal{X}}_{t,k}\rangle,$$

we obtain

$$\|\mathcal{X}_{t+1}-\mathcal{X}^*\|_F^2 \le 2\langle \mathcal{X}_{t+1}-\mathcal{X}^*, \tilde{\mathcal{X}}_{t,k}-\mathcal{X}^*\rangle.$$

To further explore the inner product $\langle \mathcal{X}_{t+1}-\mathcal{X}^*, \tilde{\mathcal{X}}_{t,k}-\mathcal{X}^*\rangle$, by using the expression of $\tilde{\mathcal{X}}_{t,k}$ and the fact of $\nabla F(\mathcal{X}^*)=0$, we split the term $\tilde{\mathcal{X}}_{t,k}-\mathcal{X}^*$ into three terms as follows:

$$\begin{aligned}\tilde{\mathcal{X}}_{t,k}-\mathcal{X}^* = &\big(\mathcal{X}_t-\mathcal{X}^*-\eta\big(\nabla f_{l_t}(\mathcal{X}_t)-\nabla f_{l_t}(\mathcal{X}^*)\big)\big)\\ &\big(\widetilde{\mathcal{X}}_k-\mathcal{X}^*-\eta\big(\nabla f_{l_t}(\widetilde{\mathcal{X}}_k)-\nabla f_{l_t}(\mathcal{X}^*)\big)\big)\\ &+\big(\widetilde{\mathcal{X}}_k-\mathcal{X}^*-\eta\big(\nabla F(\widetilde{\mathcal{X}}_k)-\nabla F(\mathcal{X}^*)\big)\big),\end{aligned}$$

which yields

$$\begin{aligned}&\langle \mathcal{X}_{t+1}-\mathcal{X}^*, \tilde{\mathcal{X}}_{t,k}-\mathcal{X}^*\rangle\\ =&\big\langle \mathcal{X}_{t+1}-\mathcal{X}^*, \mathcal{X}_t-\mathcal{X}^*-\eta\big(\nabla f_{l_t}(\mathcal{X}_t)-\nabla f_{l_t}(\mathcal{X}^*)\big)\big\rangle\\ &-\big\langle \mathcal{X}_{t+1}-\mathcal{X}^*, \widetilde{\mathcal{X}}_k-\mathcal{X}^*-\eta\big(\nabla f_{l_t}(\widetilde{\mathcal{X}}_k)-\nabla f_{l_t}(\mathcal{X}^*)\big)\big\rangle\\ &+\big\langle \mathcal{X}_{t+1}-\mathcal{X}^*, \widetilde{\mathcal{X}}_k-\mathcal{X}^*-\eta\big(\nabla F(\widetilde{\mathcal{X}}_k)-\nabla F(\mathcal{X}^*)\big)\big\rangle.\end{aligned}$$

Now, define two subspaces as follows:

$$\Gamma_t := \mathrm{span}\{\mathcal{X}_{t+1}, \mathcal{X}_t, \mathcal{X}^*\}, \quad \Gamma_t' := \mathrm{span}\{\mathcal{X}_{t+1}, \widetilde{\mathcal{X}}_k, \mathcal{X}^*\}.$$

Thus,

$$\begin{aligned}&\big\langle \mathcal{X}_{t+1}-\mathcal{X}^*, \mathcal{X}_t-\mathcal{X}^*-\eta\big(\nabla f_{l_t}(\mathcal{X}_t)-\nabla f_{l_t}(\mathcal{X}^*)\big)\big\rangle\\ =&\big\langle \mathcal{X}_{t+1}-\mathcal{X}^*, \mathcal{X}_t-\mathcal{X}^*-\eta\mathcal{P}_{\Gamma_t}\big(\nabla f_{l_t}(\mathcal{X}_t)-\nabla f_{l_t}(\mathcal{X}^*)\big)\big\rangle\\ &\big\langle \mathcal{X}_{t+1}-\mathcal{X}^*, \widetilde{\mathcal{X}}_k-\mathcal{X}^*-\eta\big(\nabla f_{l_t}(\widetilde{\mathcal{X}}_k)-\nabla f_{l_t}(\mathcal{X}^*)\big)\big\rangle\\ =&\big\langle \mathcal{X}_{t+1}-\mathcal{X}^*, \widetilde{\mathcal{X}}_k-\mathcal{X}^*-\eta\mathcal{P}_{\Gamma_t'}\big(\nabla f_{l_t}(\widetilde{\mathcal{X}}_k)-\nabla f_{l_t}(\mathcal{X}^*)\big)\big\rangle\\ &\big\langle \mathcal{X}_{t+1}-\mathcal{X}^*, \widetilde{\mathcal{X}}_k-\mathcal{X}^*-\eta\big(\nabla F(\widetilde{\mathcal{X}}_k)-\nabla F(\mathcal{X}^*)\big)\big\rangle\\ =&\big\langle \mathcal{X}_{t+1}-\mathcal{X}^*, \widetilde{\mathcal{X}}_k-\mathcal{X}^*-\eta\mathcal{P}_{\Gamma_t'}\big(\nabla F(\widetilde{\mathcal{X}}_k)-\nabla F(\mathcal{X}^*)\big)\big\rangle.\end{aligned}$$

Therefore,

$$\begin{aligned}&\frac{1}{2}\|\mathcal{X}_{t+1}-\mathcal{X}^*\|_F\\ \le&\big\|\mathcal{X}_t-\mathcal{X}^*-\eta P_{\Gamma_t}\big(\nabla f_{l_t}(\mathcal{X}_t)-\nabla f_{l_t}(\mathcal{X}^*)\big)\big\|_F\\ &+\big\|\widetilde{\mathcal{X}}_k-\mathcal{X}^*-\eta P_{\Gamma_t'}\big(\nabla f_{l_t}(\widetilde{\mathcal{X}}_k)-\nabla f_{l_t}(\mathcal{X}^*)\big)\big\|_F\\ &+\big\|\widetilde{\mathcal{X}}_k-\mathcal{X}^*-\eta P_{\Gamma_t'}\big(\nabla F(\widetilde{\mathcal{X}}_k)-\nabla F(\mathcal{X}^*)\big)\big\|_F.\end{aligned}$$

Taking the expectation on both sides of the inequality results in the following expression:

$$\begin{aligned}&\frac{1}{2}\mathbb{E}_{l_t}\|\mathcal{X}_{t+1}-\mathcal{X}^*\|_F\\ \le&\mathbb{E}_{l_t}\big\|\mathcal{X}_t-\mathcal{X}^*-\eta P_{\Gamma_t}\big(\nabla f_{l_t}(\mathcal{X}_t)-\nabla f_{l_t}(\mathcal{X}^*)\big)\big\|_F\\ &+\mathbb{E}_{l_t}\big\|\widetilde{\mathcal{X}}_k-\mathcal{X}^*-\eta P_{\Gamma_t'}\big(\nabla f_{l_t}(\widetilde{\mathcal{X}}_k)-\nabla f_{l_t}(\mathcal{X}^*)\big)\big\|_F\\ &+\mathbb{E}_{l_t}\big\|\widetilde{\mathcal{X}}_k-\mathcal{X}^*-\eta P_{\Gamma_t'}\big(\nabla F(\widetilde{\mathcal{X}}_k)-\nabla F(\mathcal{X}^*)\big)\big\|_F.\end{aligned}$$

Since

$$\eta < \frac{3\delta_{3r}^- + \omega}{3\delta_{3r}^+\delta_{3r}^-} < \frac{2}{\delta_{3r}^2},$$

by using (3.12) and (3.13), we have

$$\mathbb{E}_{l_t}\|\mathcal{X}_{t+1} - \mathcal{X}^*\|_F \leq 2\rho_{3r}(\|\mathcal{X}_t - \mathcal{X}^*\|_F + 2\|\widetilde{\mathcal{X}}_k - \mathcal{X}^*\|_F).$$

Here

$$\rho_{3r} = \sqrt{1 + \eta^2\delta_{3r}^+\delta_{3r}^- - 2\eta\delta_{3r}^-}, \quad \delta_{3r}^- = 2(1 - \delta_{3r}), \quad \delta_{3r}^+ = 2(1 + \delta_{3r}).$$

Let $\theta = 2\rho_{3r}$, by recursively using the above inequality over t, and noting that $\widetilde{\mathcal{X}}_k = \mathcal{X}_0$ and $\widetilde{\mathcal{X}}_{k+1} = \mathcal{X}_n$, we can deduce

$$\mathbb{E}_{l_t}\|\widetilde{\mathcal{X}}_{k+1} - \mathcal{X}^*\|_F = \mathbb{E}_{l_t}\|\mathcal{X}_n - \mathcal{X}^*\|_F \leq \theta\|\mathcal{X}_{n-1} - \mathcal{X}^*\|_F + 2\theta\|\widetilde{\mathcal{X}}_k - \mathcal{X}^*\|_F,$$

from which, we further have

$$\begin{aligned}
&\mathbb{E}_{l_t}\|\widetilde{\mathcal{X}}_{k+1} - \mathcal{X}^*\|_F \\
&\leq \theta\mathbb{E}_{l_t}\|\mathcal{X}_{n-1} - \mathcal{X}^*\|_F + 2\theta\|\widetilde{\mathcal{X}}_k - \mathcal{X}^*\|_F \\
&\leq \theta^2\|\mathcal{X}_{n-2} - \mathcal{X}^*\|_F + 2(\theta^2 + \theta)\|\widetilde{\mathcal{X}}_k - \mathcal{X}^*\|_F.
\end{aligned}$$

Following the same argument, we have

$$\begin{aligned}
&\mathbb{E}_{l_t}\|\widetilde{\mathcal{X}}_{k+1} - \mathcal{X}^*\|_F \\
&\leq \theta^n\|\mathcal{X}_0 - \mathcal{X}^*\|_F + 2\left(\sum_{i=1}^{n}\theta^i\right)\|\widetilde{\mathcal{X}}_k - \mathcal{X}^*\|_F \\
&= \frac{-3\theta^{n+1} + \theta^n + 2\theta}{1 - \theta}\|\widetilde{\mathcal{X}}_k - \mathcal{X}^*\|_F \leq \|\widetilde{\mathcal{X}}_k - \mathcal{X}^*\|_F.
\end{aligned}$$

The last inequality follows from

$$\kappa = \frac{-3\theta^{n+1} + \theta^n + 2\theta}{1 - \theta} < 1,$$

since $\delta_{3r} < 1/71$ and

$$\frac{3\delta_{3r}^- - \omega}{3\delta_{3r}^+\delta_{3r}^-} < \eta < \frac{3\delta_{3r}^- + \omega}{3\delta_{3r}^+\delta_{3r}^-},$$

where $\omega = \sqrt{71\delta_{3r}^2 - 72\delta_{3r} + 1}$. □

Obviously, the linear convergence of the TSVRG method for the tensor recovery problem with the CP or Tucker rank constraint follows immediately.

In the next section, we focus on two types of tensor data that are selected for clarity of presentation, and with the aim of confirming our theoretical results.

4. Numerical Experiments

In this section, we present numerical experiments to demonstrate the performance of the TSVRG method for tensor recovery tasks constrained by the CP or Tucker rank. All experiments use third-order tensor data (i.e. $d = 3$), including both synthetic and real datasets. For

comprehensive and complete comparisons, we first evaluate recovery performance based on the CP rank constraint of the tensor with TIHT [7]. Then, we compare the performance based on the Tucker rank constraint of the tensor with TIHT and StoIHT [6]. In the numerical results, the red, green, and blue lines represent the TIHT, StoIHT, and TSVRG methods, respectively. Furthermore, we use the Barzilai-Borwein (BB) [2, 25] technique for automatic adjustment of step sizes η according to the information of the first two iterations, helping the algorithm to converge to the optimal solution faster. For the k-th iteration, we set

$$\eta_k = \frac{\|\widetilde{\mathcal{X}}_k - \widetilde{\mathcal{X}}_{k-1}\|_F^2}{\langle \widetilde{\mathcal{X}}_k - \widetilde{\mathcal{X}}_{k-1}, g_k - g_{k-1}\rangle}, \quad g_k = \nabla F(\widetilde{\mathcal{X}}_k).$$

The BB strategy has been highly effective in practical applications [1, 12, 27]. And it is not convergent in the process of proving [3] in general. The same case appears in our paper. Specifically, η_k in the numerical experiment is within the range of η in Theorem 3.1. Each experimental result shows that SVRG with the BB strategy is convergent.

All methods are executed on a desktop running Windows 11 and MATLAB (R2023a) with an Intel(R) Core(TM) i7-10700 CPU at 2.90 GHz and 64 GB RAM. For parameter settings, the maximum number of iterations $K = 500$ is used for both synthetic and real data across all methods. We randomly select $30\%, 50\%, 70\%, 90\%$ of the entries in the tensor as observed data, denoted as the sample ratio (SR). Experimental data (both synthetic and video data) and parameters (such as CP or Tucker rank r and the corresponding SR) are based on the guidance provided in [6, 7].

We assess the performance of tensor recovery using two common criteria: the relative square error (RSE) and the peak signal-to-noise ratio (PSNR), defined as follows:

$$\text{RSE} := \frac{\|\mathcal{X}^* - \mathcal{X}\|_F}{\|\mathcal{X}\|_F}, \quad \text{PSNR} := 10\log_{10}\left(\frac{\|\mathcal{X}\|_\infty^2}{1/(i_1 i_2 \cdots i_d)\|\mathcal{X}^* - \mathcal{X}\|_F^2}\right),$$

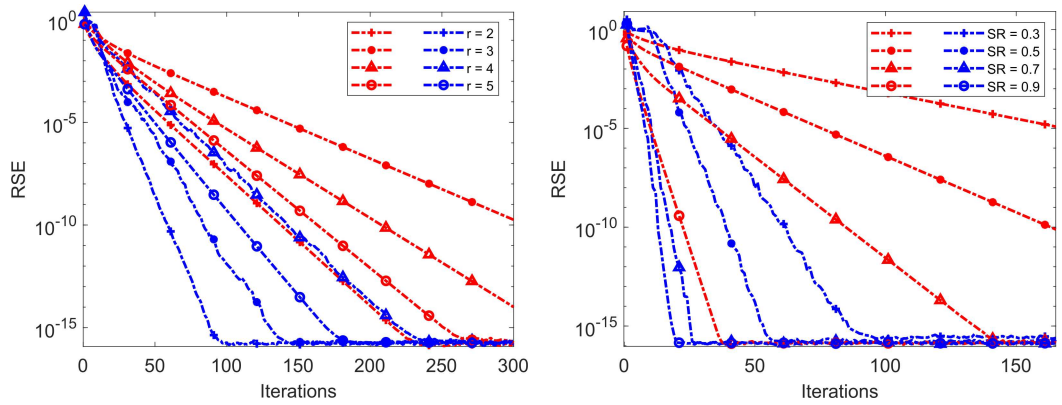
where $\mathcal{X}^*, \mathcal{X}$ respectively represent the recovered tensor and original tensor. We consider recovery successful if RSE $\leq 10^{-3}$. In each figure of the numerical experiments, we also observe varying levels of the "convergence horizon", which refers to the RSE values after the algorithm has reached a relatively stable state. Another metric to assess the recovery effect is PSNR. Each run is repeated 5 times, and we report the average results. In addition, we show the execution time of each algorithm on real data, measured in seconds. For simplicity, we have rounded the running time and the PSNR to two decimal places.

4.1. Performance comparison with CP rank restraint

In this subsection, we demonstrate the effectiveness of the TSVRG method compared to TIHT using both synthetic and real data under CP rank constraints. The "thresholding" step $\mathcal{H}_r$ in Eq. (2.5) is performed using the `cpd` function from the Tensorlab 3.0 package.

Firstly, for verification with synthetic data, we generate $n_1 \times n_2 \times n_3$ tensors of CP rank r, whose elements adhere to Gaussian distributions. The positions of the observed entries on $n_1 \times n_2 \times n_3$ are randomly and uniformly selected. The tensor size is set to $n_1 = n_2 = n_3 = 10$, with different SR $= \{30\%, 50\%, 70\%, 90\%\}$ and the CP rank r ranges from 2 to 5 to demonstrate performance compared to TIHT. Fig. 4.1 presents results that illustrate the RSE for different ranks as well as different SR values.

In our first experiment, we report the RSE results of TIHT and TSVRG against varying CP ranks r with SR $= 0.5$ in Fig. 4.1(a). In Fig. 4.1(b), we set the CP rank $r = 2$ and vary the SR,



(a) (b)

Fig. 4.1. RSE versus iteration for $n_1 = n_2 = n_3 = 10$. (a): Various CP rank with SR $= 0.5$; (b): Various SR with CP rank $r = 2$.

showing the RSE of TIHT and TSVRG for different SR values. The differences in initial values arise from the random selection of observed locations, which is a phenomenon that is observed consistently in later experiments.

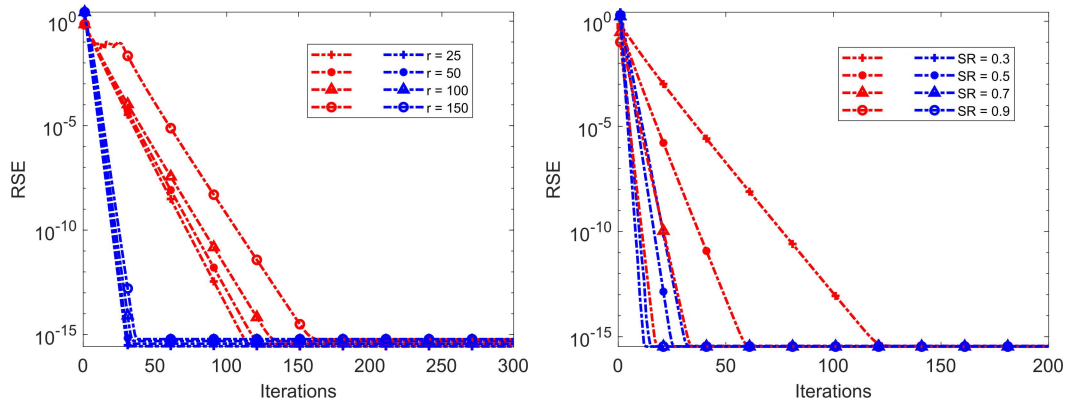
As expected, both TIHT and TSVRG demonstrate that tensors with lower CP ranks show a more rapid convergence trend. However, this does not imply a slower convergence tendency for tensors with larger CP ranks. For example, the convergence rate for $r = 5$ is more effective than that for $r = 4$ in terms of TSVRG. Furthermore, the convergence rates of both methods indicate that TSVRG outperforms TIHT under the same rank conditions. Most importantly, the convergence horizon and the convergence rate of TSVRG are superior to those of TIHT across all demonstrated ranks.

In Fig. 4.1(b), we observe that higher SR values lead to faster convergence tendencies, as predicted. The convergence horizon of TSVRG is better than that of TIHT for all SR values. The convergence rate of TSVRG outperforms TIHT for each different SR. Specifically, TSVRG exceeds TIHT in the convergence rate for SR=$30\%, 50\%, 70\%$. Overall, the recovery performance of TSVRG is better than that of TIHT in synthetic data under CP rank constraints.

To validate the effectiveness of TSVRG on relatively large-scale synthetic data, we generate a tensor of size $n_1 = n_2 = n_3 = 200$ with CP rank r, and compare its performance with TIHT. Fig. 4.2 presents the RSE versus the number of iterations in two experimental settings: (a) varying CP ranks with a fixed SR value and (b) varying SR values with a fixed CP rank.

In Fig. 4.2(a), we fix the SR to 0.3 and vary CP rank $r = \{25, 50, 100, 150\}$. As observed, TSVRG exhibits significantly faster convergence than TIHT across all CP ranks. For example, at $r = 25$, TSVRG achieves an RSE near 10^{-15} in fewer than 20 iterations, while TIHT requires more than 100 iterations to reach a similar level. Even for higher ranks such as $r = 150$, TSVRG maintains a rapid decline in RSE and achieves a lower convergence horizon than TIHT. These results demonstrate that TSVRG is more efficient with respect to varying CP ranks.

In Fig. 4.2(b), we fix the CP rank to $r = 50$ and vary SR $= \{30\%, 50\%, 70\%, 90\%\}$. As expected, both algorithms benefit from a relatively large SR value, with faster convergence observed at higher SR values. However, TSVRG consistently outperforms TIHT in all SR values. In particular, in SR $= 0.3$, TSVRG reduces the convergence horizon by approximately 40 iterations, while TIHT converges much more slowly. At SR $= 0.9$, both algorithms converge quickly,



(a) (b)

Fig. 4.2. RSE versus iteration for $n_1 = n_2 = n_3 = 200$. (a): Various CP rank with SR = 0.3; (b): Various SR with CP rank $r = 50$.

though TSVRG still maintains a slight advantage in both the rate and convergence horizon. Overall, the experimental results clearly show that TSVRG achieves superior convergence speed and recovery accuracy compared to TIHT under a wide range of CP ranks and SR values.

Next, we verify the recovery effectiveness of the TSVRG method using real data. We select the real tensor data (candle video) from [7]. For computational efficiency, we used a smaller image, resulting in a candle video size of $30 \times 30 \times 10$, where the 10 images are part of a larger image. The positions of the observed entries are generated randomly, with a CP rank of 15 and an SR of 0.8. Both PSNR and RSE are used for comparison.

Quantitative comparisons, shown in Fig. 4.3(a), indicate that TSVRG has a faster convergence rate and a lower convergence horizon than TIHT. Furthermore, we observe that the proposed TSVRG algorithm escapes local minima within 100 iterations, validating the theoretical advantage of SVRG over GD or SGD. As illustrated by the images in Figs. 4.3(c) and 4.3(d), the first frame recovered by TSVRG in the 150-th iteration exhibits higher visual quality compared to TIHT.

By the 150-th iteration, the execution times for TIHT and TSVRG are 304.05 seconds and 161.30 seconds, respectively. The PSNR values for TIHT and TSVRG are 25.70 dB and 28.76 dB, respectively. These experimental results confirm that our TSVRG method outperforms TIHT for both synthetic and real data under CP rank constraints.

4.2. Performance comparison with Tucker rank restraint

In this subsection, we demonstrate the efficacy of TSVRG compared to TIHT and StoTIHT using synthetic and real data under Tucker rank constraints. The "thresholding" step $\mathcal{H}_r$ in Eq. (2.5) is performed using the `mlsvd` function from the Tensorlab 3.0 package.

For verification with synthetic data, we generate a tensor of size $n_1 \times n_2 \times n_3$ with Tucker rank r. The entries of this tensor follow Gaussian distributions, and the positions of the observed entries are selected randomly and uniformly. The tensor size is set to $n_1 = n_2 = 5$, $n_3 = 6$ with SR values of $\{30\%, 50\%, 70\%, 90\%\}$. The Tucker ranks considered are $r = (1, 1, 2; 1, 2, 2; 2, 2, 2; 2, 2, 3)$, to validate the effectiveness of TSVRG by comparing it with TIHT and StoTIHT.

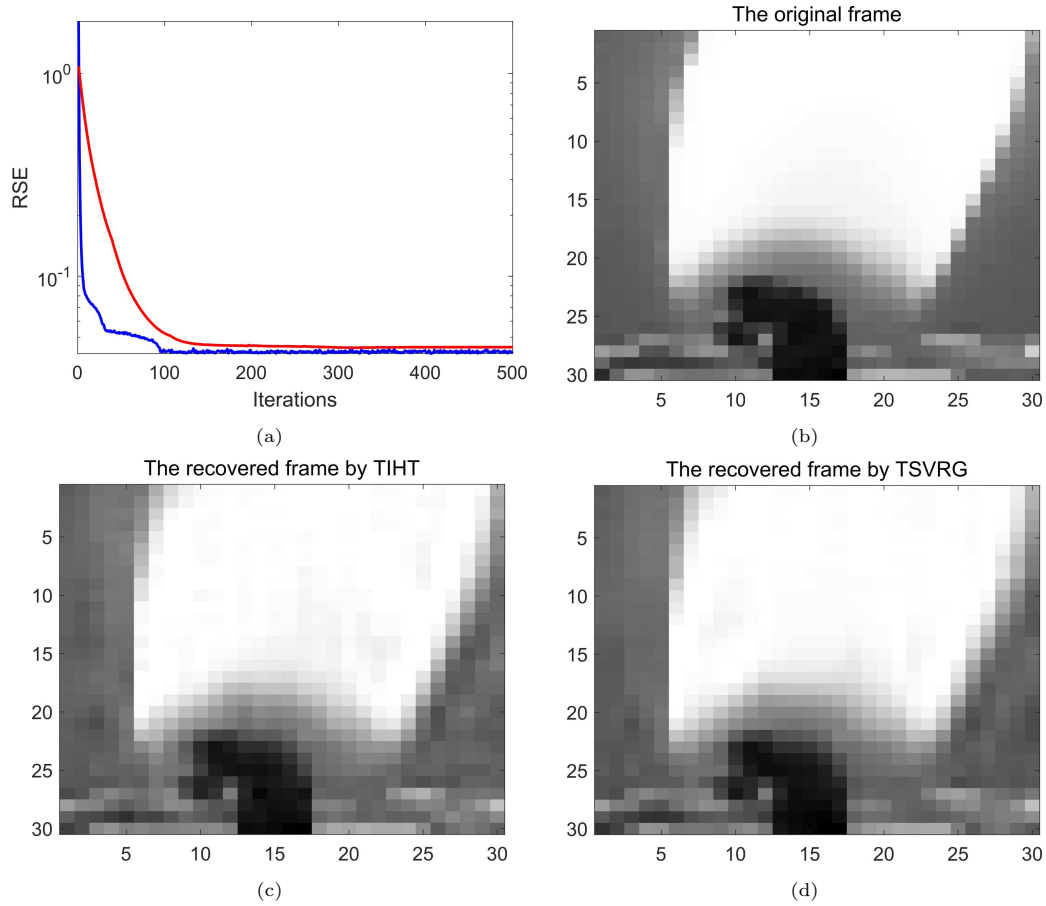
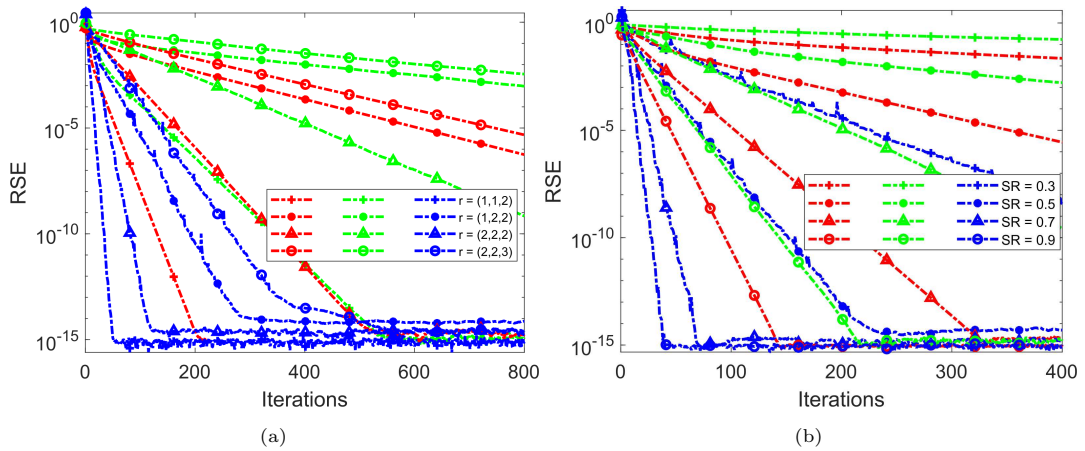


Fig. 4.3. The size of Candle video $30 \times 30 \times 10$, SR=80%, CP rank $r = 15$. (a): RSE vs iteration; (b): The original image corresponding to the first frame; (c)-(d): The recovered frame by TIHT and by TSVRG in the 150 iteration.

As shown in Fig. 4.4, we illustrate the RSE versus iteration for different Tucker ranks with SR $= 0.5$ in Fig. 4.4(a) and different SR values with Tucker rank $r = (1,\ 2,\ 2)$ in Fig. 4.4(b). The results indicate that the convergence rate of TSVRG is superior to that of TIHT and StoTIHT, as seen in Fig. 4.4(a). Specifically, both the convergence rate and the convergence horizon show that TSVRG outperforms TIHT and StoTIHT for the same Tucker rank. In Fig. 4.4(b), the convergence rate of TSVRG is evidently superior to both TIHT and StoTIHT, and the convergence horizon of TSVRG is slightly better than the other two methods. Overall, TSVRG demonstrates a clear advantage over the other two algorithms in synthetic data under Tucker rank constraints.

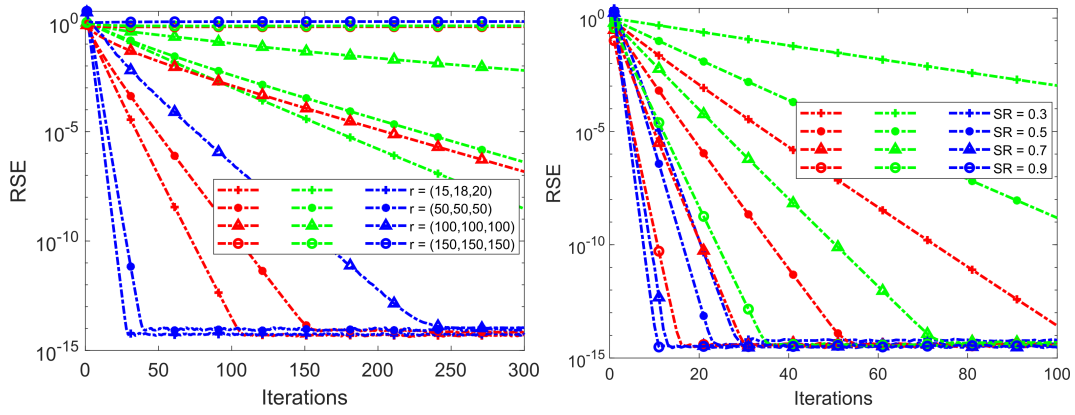
To validate the effectiveness of TSVRG on relatively large-scale synthetic data, we generate a tensor of size $n_1 = n_2 = n_3 = 200$ with Tucker rank r, and compare its performance with TIHT and StoTIHT.

As shown in Fig. 4.5, we illustrate the RSE versus iteration for various Tucker ranks and different SR values. Specifically, Fig. 4.5(a) shows the performance under different Tucker ranks $r = \{15, 18, 20; 50, 50, 50; 100, 100, 100; 150, 150, 150\}$ (The Tucker rank $(15, 18, 20)$ is adopted from [23].) with a fixed SR $= 0.3$. Changes in the Tucker rank have a pronounced impact on



(a) (b)

Fig. 4.4. RSE versus iteration for $n_1 = n_2 = 5, n_3 = 6$. (a): Various Tucker rank with SR $= 0.5$; (b): Various SR with Tucker rank $r = (1, 2, 2)$.



(a) (b)

Fig. 4.5. RSE versus iteration for $n_1 = n_2 = n_3 = 200$. (a): Various Tucker rank with SR $= 0.3$; (b): Various SR with Tucker rank $r = (15, 18, 20)$.

the convergence behavior of TSVRG, TIHT, and StoTIHT. TSVRG consistently achieves faster convergence compared to the other two methods.

Fig. 4.5(b) presents the results under varying SR $= \{30\%, 50\%, 70\%, 90\%\}$ with a fixed Tucker rank $r = (15, 18, 20)$. It can be observed that higher sample ratios lead to faster convergence and final convergence horizon across all methods. Notably, TSVRG exhibits the most rapid decrease in RSE under all SR settings, again confirming its superiority in both convergence rate and final recovery accuracy. These results further validate the efficiency of TSVRG under different sampling conditions.

Furthermore, we will confirm the efficacy of TSVRG in real data with Tucker rank restraint. Similarly to the case of CP rank restraint, we use the Candle video data set. The size of the Candle video and the method for selecting observed positions are consistent with the CP rank experiments. Furthermore, we adopt a Tucker rank of $r = (8, 8, 2)$ and set the sample ratio to 0.5.

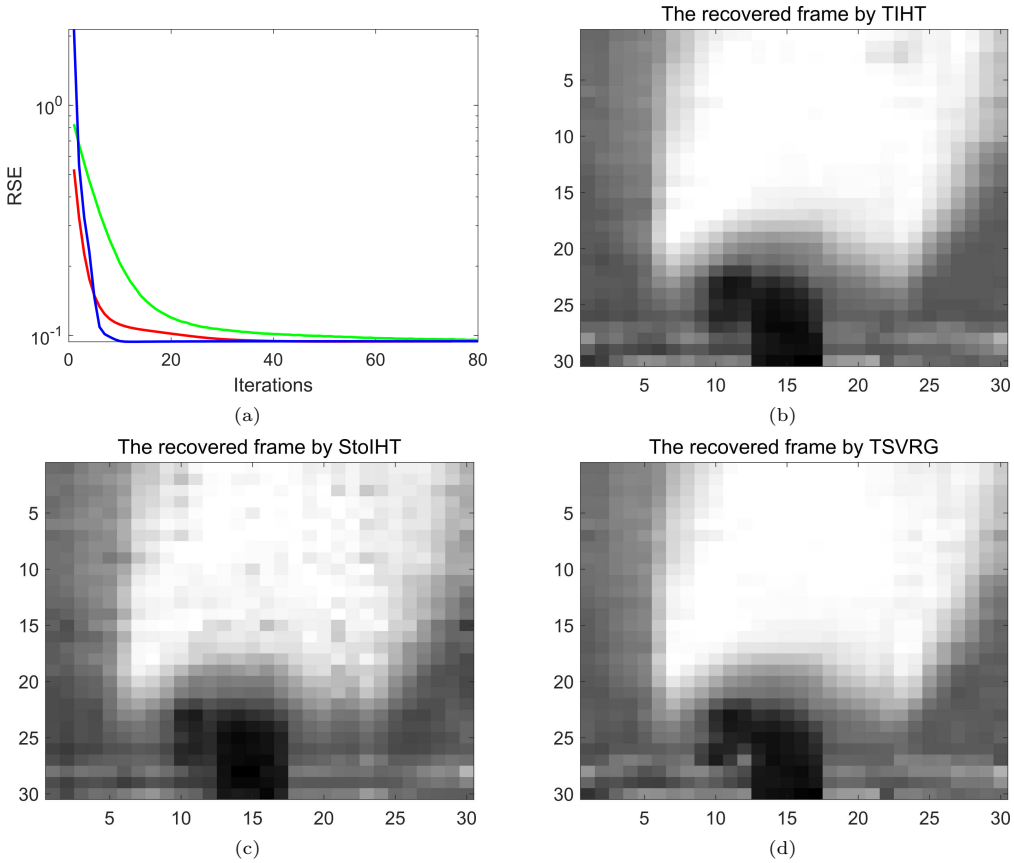


Fig. 4.6. The size of Candle video $30 \times 30 \times 10, \mathrm{SR} = 50\%$, Tucker rank $r = (8, 8, 2)$. The original image corresponding to the first frame as same as the first image in Fig. 4.3. (a): RSE vs iteration; (b)-(d): The recovered frame by TIHT, by StoTIHT and by TSVRG in the 20 iteration.

As shown in Fig. 4.6(a), the quantitative comparisons demonstrate that TSVRG achieves a lower convergence horizon within a limited number of iterations and a faster convergence rate compared to TIHT and StoTIHT. Furthermore, we present the recovered results of the first frame in the 20-th iteration in Figs. 4.6(b)-4.6(d).

By the 20-th iteration, the execution times of the TIHT, StoTIHT, and TSVRG algorithms are 0.08 s, 0.03 s, and 0.06 s, respectively. The PSNR values for these three algorithms are 22.56 dB, 21.20 dB, and 23.04 dB, respectively. It can be observed that our algorithm achieves the highest PSNR in slightly less time within 20 iterations. These experimental results verify that TSVRG outperforms both TIHT and StoTIHT for synthetic and real data under Tucker rank restraint.

In all experiments, we demonstrate that the TSVRG algorithm consistently outperforms other TIHT algorithms based on GD and SGD, whether under CP or Tucker rank restraint.

5. Conclusion

We present a straightforward, yet highly effective, TSVRG method for addressing the tensor recovery problem. TSVRG demonstrates a significantly beneficial balance of efficiency and

precision compared to other greedy algorithms. It is worth noting that the condition for linear convergence may not be the most efficient in the present version, and there is potential for finding a more effective condition by further relaxation. Additionally, all numerical experiments indicate that the TSVRG method is slightly faster in terms of time consumption under the same parameter settings. However, we have not yet provided an exact approximate form to express the computational complexity of applying the SVRG method to a tensor recovery problem with CP rank restraint or Tucker rank restraint. Addressing these incomplete aspects will be the direction of our future efforts.

Acknowledgments. The work of L. Li was supported in part by the Shenzhen University (Grant No. 868-000002020258). The work of C. Xu was supported in part by the NSF of China (Grant No. 62372302). The work of J. Lu was supported in part by the National Natural Science Foundation of China (Grant Nos. U21A20455, 12571569), in part by the Natural Science Foundation of Guangdong Province of China (Grant Nos. 2023A1515011691, 2024A1515011913), in part by the Shenzhen Basis Research Project of China (Grant No. JCYJ20210324094006017), in part by the Shenzhen Key Laboratory of Advanced Machine Learning and Applications (Grant No. SYSPG20241211173920042). The work of N. Han was supported in part by the NSF of China (Grant No. 12201456). The work of L. Shen was supported in part by the NSF of China (Grant No. DMS-2208385).

References

[1] A. Abubakar, P. Kumam, H. Mohammad, and A. Awwal, A Barzilai-Borwein gradient projection method for sparse signal and blurred image restoration, *J. Franklin Inst.*, **357**:11 (2020), 7266–7285.

[2] J. Barzilai and J. Borwein, Two-point step size gradient methods, *IMA J. Numer. Anal.*, **8**:1 (1988), 141–148.

[3] R. Fletcher, On the Barzilai-Borwein method, in: *Optimization and Control with Applications. Applied Optimization*, Vol. 96, Springer, (2005), 235–256.

[4] M. Fornasier, H. Rauhut, and R. Ward, Low-rank matrix recovery via iteratively reweighted least squares minimization, *SIAM J. Optim.*, **21** (2011), 1614–1640.

[5] L. Grasedyck, Hierarchical singular value decomposition of tensors, *SIAM J. Matrix Anal. Appl.*, **31**:4 (2010), 2029–2054.

[6] R. Grotheer, S. Li, A. Ma, D. Needell, and J. Qin, Stochastic iterative hard thresholding for low-Tucker-rank tensor recovery, in: *2020 Information Theory and Applications Workshop*, IEEE, (2020), 1–5.

[7] R. Grotheer, S. Li, A. Ma, D. Needell, and J. Qin, Iterative hard thresholding for low CP-rank tensor models, *Linear Multilinear Algebra*, **70**:22 (2021), 7452–7468.

[8] N. Han, J. Nie, J. Lu, and M. Ng, Stochastic variance reduced gradient for affine rank minimization problem, *SIAM J. Imaging Sci.*, **17**:2 (2024), 1118–1144.

[9] J. Huang, F. Zhang, B. Safaei, Z. Qin, and F. Chu, The flexible tensor singular value decomposition and its applications in multisensor signal fusion processing, *Mech. Syst. Signal Process.*, **220** (2024), 111662.

[10] R. Johnson and T. Zhang, Accelerating stochastic gradient descent using predictive variance reduction, in: *Proceedings of the 27th International Conference on Neural Information Processing Systems*, Vol. 1, Curran Associates, Inc., (2013), 315–323.

[11] T. Kolda and B. Bader, Tensor decompositions and applications, *SIAM Rev.*, **51** (2009), 455–500.

[12] J. Liang, Y. Xu, C. Bao, Y. Quan, and H. Ji, Barzilai-Borwein-based adaptive learning rate for deep learning, *Pattern Recognit. Lett.*, **128** (2019), 197–203.

[13] J. Liu, P. Musialski, P. Wonka, and J. Ye, Tensor completion for estimating missing values in visual data, *IEEE Trans. Pattern Anal. Mach. Intell.*, **35**:1 (2013), 208–220.

[14] X. Liu, S. Aeron, V. Aggarwal, and X. Wang, Low-tubal-rank tensor completion using alternating minimization, *IEEE Trans. Inf. Theory*, **66**:3 (2020), 1714–1737.

[15] Y. Liu, Z. Long, H. Huang, and C. Zhu, Low CP rank and Tucker rank tensor completion for estimating missing components in image data, *IEEE Trans. Circuits Syst. Video Technol.*, **30**:4 (2020), 944–954.

[16] Z. Long, Y. Liu, L. Chen, and C. Zhu, Low rank tensor completion for multiway visual data, *Signal Process.*, **155** (2019), 301–316.

[17] Z. Long, C. Zhu, J. Liu, and Y. Liu, Bayesian low rank tensor ring for image recovery, *IEEE Trans. Image Process.*, **30** (2021), 3568–3580.

[18] J. Lu, C. Xu, Z. Hu, X. Liu, Q. Jiang, D. Meng, and Z. Lin, A new nonlocal low-rank regularization method with applications to magnetic resonance image denoising, *Inverse Problems*, **38**:6 (2022), 065012.

[19] D. Qiu, M. Ng, and X. Zhang, Low-rank matrix completion with Poisson observations via nuclear norm and total variation constraints, *J. Comput. Math.*, **42**:6 (2024), 1427–1451.

[20] H. Rauhut, R. Schneider, and Ž. Stojanac, Low-rank tensor recovery via iterative hard thresholding, *Linear Algebra Appl.*, **523** (2017), 220–262.

[21] H. Sato, H. Kasai, and B. Mishra, Riemannian stochastic variance reduced gradient with retraction and vector transport, *SIAM J. Optim.*, **29**:201 (2019), 1444–1472.

[22] F. Shang, B. Wei, H. Liu, Y. Liu, P. Zhou, and M. Gong, Efficient gradient support pursuit with less hard thresholding for cardinality-constrained learning, *IEEE Trans. Neural Netw. Learn. Syst.*, **33**:12 (2022), 7806–7817.

[23] Q. Shi, H. Lu, and Y. Cheung, Tensor rank estimation and completion via CP-based nuclear norm, in: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, ACM, (2017), 949–958.

[24] G. Song, M. Ng, and X. Zhang, Tensor completion by multi-rank via unitary transformation, *Appl. Comput. Harmon. Anal.*, **65** (2023), 348–373.

[25] C. Tan, S. Ma, Y. Dai, and Y. Qian, Barzilai-Borwein step size for stochastic gradient descent, in: *Proceedings of the 30th International Conference on Neural Information Processing Systems*, Curran Associates, Inc., (2016), 685–693.

[26] Y. Ying, L. Chen, and G. Chen, A temporal-aware POI recommendation system using context-aware tensor decomposition and weighted HITS, *Neurocomputing*, **242** (2017), 195–205.

[27] T. Yu, X. Liu, Y. Dai, and J. Sun, Stochastic variance reduced gradient methods using a trust-region-like scheme, *J. Sci. Comput.*, **87** (2021), 5.