CONVERGENCE ANALYSIS ON SVD-BASED ALGORITHMS FOR TENSOR LOW RANK APPROXIMATIONS

YU GUAN

NATIONAL UNIVERSITY OF SINGAPORE

2018

CONVERGENCE ANALYSIS ON SVD-BASED ALGORITHMS FOR TENSOR LOW RANK APPROXIMATIONS

YU GUAN

(MA.Sc., Shanghai University)

A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
DEPARTMENT OF MATHEMATICS
NATIONAL UNIVERSITY OF SINGAPORE
2018

Supervisor:
Associate Professor Chu Delin
Examiners:
Associate Professor Yang Haizhao
Associate Professor Tan Hwee Huat
Professor Li Wen, South China Normal University

DECLARATION

I hereby declare that the thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

GuanYu

YU GUAN

22 August 2018

Acknowledgements

First and foremost, I owe to my deepest gratitude to my supervisor Professor Delin Chu for his encouragement and support throughout the last few years. His professional and precise attitude on research and spirit of assiduous study has impressed and inspired me a lot. Professor Chu has taught me how to choose my profession and how to be a true scholar. He has also provided me with many opportunities. Therefore, I have attended several conferences where I gained a lot from other scholars. Secondly, I am most indebted to Professor Moody Chu. He has kindly offered me a great chance being a visiting scholar under his guidance at North Carolina State University for half a year during my PHD period. He has taught me the most important things about pursuing topics that interested me and helped me build a solid foundation on tensor decompositions and approximations. His hard working, rigorous and persistence attitude, great passion on research have influenced me deeply. As a young scholar, I am trying to catch up with his step and build myself up. Besides, I also want to thank my parents who have been always the most supportive and understanding. There are many friends whom I owe huge debts of gratitude and have supported and helped me both in academic area and usual life: Teng Dan, Qian Lilong, Zhang Xiaowei, Li Ning, Zhang Yangjing, Pang Tongyao, Lam Xinyee, Wang Liuquan, Wu Bin and so on.

Yu Guan, 22 August 2018

Contents

Summary

This thesis is to study a few problems on tensor decompositions and approximations in real space. Among other things, we revisit the classical problem of finding the best rank-R CANDECOMP/PARAFAC(CP) approximation with different cases $R > 1$ and $R = 1$ respectively.

- Best rank-R CP approximation

  For a given order-k tensor T, determine unit vectors $\mathbf{u}_r^{(\ell)} \in \mathbb{R}^{I_\ell}$, $\ell = 1, \dots k$ and scalars λ_r, $r = 1, \dots, R$ to minimize

$$\left\| T - \sum_{r=1}^{R} \lambda_r \mathbf{u}_r^{(1)} \otimes \cdots \otimes \mathbf{u}_r^{(k)} \right\|_F^2 .$$

Unlike the rank-1 approximation is theoretically guaranteed to have a global optimum, general rank-R approximation ($R > 1$) may not exist in real space. So, there should be an added orthogonality requirement to ensure the existence of $R > 1$ case. In contrast to the conventional approach by the so-called alternating least squares (ALS) method that works to adjust one factor a time, proposed SVD-based algorithms improve two factors simultaneously. Convergence analysis both for the generalized Rayleigh quotient and the iterates themselves is the main contribution of this thesis. In addition, we also study the convergence property of a general framework called alternating direction methods (ADM) in this thesis.

List of Figures

Chapter 1

Introduction

1.1 Applications of tensor decompositions

After Hitchcock originally proposed the notion of tensor decomposition in 1927 [92,93], it has been applied in a wide range of areas: signal processing [6,38,48,59–61, 64,73,105,145,160], numerical linear algebra [62,63,77,110,111,124,197], computer vision [87,159,183–188], numerical analysis [19,20,80,81,94,103,104], data mining and analysis [3–5,12,36,42,43,131,134,158,170–172,180] graph analysis [13,113,115], neuroscience [1,2,18,66,67,136–138,141–144], chemometrics [7,9,26,127,163], image processing [53,184,186,187,196], component analysis [101,119], network analysis, scientific computing, telecommunications [57,161,162], independent component analysis (ICA) [58], Newton potential [80,81], stochastic PDEs [69,194] and many other areas [25,27,48,49,54,65,72,88,106,118,155,163]. Moreover, there are several software packages available for tensor structures and decompositions in [8,11,14,15,76,122,152,191,195].

1.2 Main decompositions and approximations

Rank-1 tensor

Rank-1 tensor has the form $\mathbf{u}^{(1)}\otimes\ldots\otimes\mathbf{u}^{(k)} := \left[u^{(1)}_{i_1}\ldots u^{(k)}_{i_k}\right]$, where vectors $\mathbf{u}^{(j)} \in \mathbb{R}^{I_j}$ with elements $[u^{(j)}_{i_j}]$ for $j = 1, \ldots, k$.

Tensor decomposition

Tensor decomposition is to rewrite the given tensor T as the summation of some rank-1 tensors.

Among many kinds of decompositions of high order tensors, the most general two are Tucker decomposition [62,63,92,178] and CP decomposition [32,72,82,107].

- Tucker decomposition

$$T = \sum_{r_1,r_2,\ldots,r_k} \lambda_{r_1,r_2,\ldots,r_k}\mathbf{u}^{(1)}_{r_1} \otimes \cdots \otimes \mathbf{u}^{(k)}_{r_k}, \tag{1.1}$$

  where $\lambda_{r_1,r_2,\ldots,r_k} \in \mathbb{R}$ and $\mathbf{u}^{(\ell)}_{r_\ell} \in \mathbb{R}^{I_\ell}$ are unit vectors for $\ell = 1, \ldots, k$.

- CP decomposition

$$T = \sum_{r} \lambda_r\mathbf{u}^{(1)}_{r} \otimes \cdots \otimes \mathbf{u}^{(k)}_{r}, \tag{1.2}$$

  where $\lambda_r \in \mathbb{R}$ and $\mathbf{u}^{(\ell)}_{r} \in \mathbb{R}^{I_\ell}$ are unit vectors for $\ell = 1, \ldots, k$.

Tucker decomposition also named as higher-order PCA was first introduced by Tucker [176] in 1963 and redefined in [128, 177, 178]. CANDECOMP/PARAFAC (CP) decomposition, also named as polyadic form, CANDECOMP (canonical decomposition) and PARAFAC (parallel factors) were proposed by Hitchock in 1927 [92, 93], Cattell in 1944 [34, 35], Carroll and Chang in 1970 [32] and Harshman in 1970 [82]. Both Tucker decomposition and CP decomposition can be considered to be higher order generalizations of the matrix singular value decomposition (SVD) and principal component analysis (PCA).

Besides CP and Tucker, there are lots of other tensor decompositions related to or transformed by them. For example, Carroll and Chang [32] proposed individual differences in scaling (INDSCAL) in 1970, Harshman [83] named parallel factors for cross products (PARAFAC2) in 1972, Carroll et al. [33] introduced CANDECOMP with linear constraints (CANDELINC) in 1980, Harshman [84] established decomposition into directional components (DEDICOM) in 1978 and he with Lundy [85] presented PARAFAC and Tucker2 (PARATUCK2) in 1996 and so on.

Rank

We clarify the name of "rank" here which refers to the number of rank-1 tensors that generate or approximate the given tensor. It might be the same with (outer-product) tensor rank defined in [40,65,92,121]. Since we care about "low rank", this number will be fixed on low rank approximation problem. Many kinds of ranks are named such as multilinear rank [65], border rank [65], symmetric rank [50], generic rank [65].

Tensor approximation

As finding an exact decomposition of a tensor is NP-hard [86,91], an alternative approach is finding the low rank approximation which seems more computationally feasible. Tensor approximation is to minimize the difference between the given tensor and the summation in the sense of the Frobenius norm after fixing the "rank" whose choice is itself a difficult problem [52,114,121] and affects the quality of approximation. Three well known approximations which receive most concern and interest are rank-1 approximation [63,197], rank-$(r_1, r_2, \ldots, r_k)$ approximation with a full core and k orthogonal side-matrices (in the Tucker/HOOI fashion) and approximations using R outer-product terms (in the CANDECOMP/PARAFAC fashion).

- Best rank-1 approximation

  The problem of finding a best rank-1 approximation to a given order-k tensor

T is to determine unit vectors $\mathbf{u}^{(\ell)} \in \mathbb{R}^{I_\ell}$, $\ell = 1, \dots k$, and a scalar λ such that

$$\left\| T - \lambda \mathbf{u}^{(1)} \otimes \cdots \otimes \mathbf{u}^{(k)} \right\|_F^2 \tag{1.3}$$

is minimized.

- Tucker nearest problem

  For a given order-k tensor T, determine unit vectors $\mathbf{u}_{r_\ell}^{(\ell)} \in \mathbb{R}^{I_\ell}$, $\ell = 1, \dots k$ and scalars $\lambda_{r_1, r_2, \dots, r_k}$ to minimize

  $$\left\| T - \sum_{r_1, r_2, \dots, r_k} \lambda_{r_1, r_2, \dots, r_k} \mathbf{u}_{r_1}^{(1)} \otimes \cdots \otimes \mathbf{u}_{r_k}^{(k)} \right\|_F^2, \tag{1.4}$$

  subject to $\langle \mathbf{u}_{r_i}^{(\ell)}, \mathbf{u}_{r_j}^{(\ell)} \rangle = \delta_{r_i r_j}$ for all $\ell = 1, \dots, k$.

- Best rank-R CP approximation

  For a given order-k tensor T, determine unit vectors $\mathbf{u}_r^{(\ell)} \in \mathbb{R}^{I_\ell}$, $\ell = 1, \dots k$ and scalars λ_r, $r = 1, \dots, R$ to minimize

  $$\left\| T - \sum_{r=1}^{R} \lambda_r \mathbf{u}_r^{(1)} \otimes \cdots \otimes \mathbf{u}_r^{(k)} \right\|_F^2. \tag{1.5}$$

This thesis focuses on best rank-1 approximation (1.3) and low rank CP approximation (1.5). In the following subsection, we will mainly introduce the backgrounds, existing work and algorithms on these two approximations. Other kinds of decompositions and approximations will just be introduced shortly and not covered deeply.

1.3 Low rank approximations

1.3.1 Existence and ill-posedness

Firstly, we discuss the existence of best low rank approximations in 3 cases.

1) Best low rank approximation of a matrix ($k = 2$).

For matrices of order-2 tensors, Eckart and Young [70] showed that a best rank-R approximation always exists and is precisely given by the truncated singular value decomposition (TSVD).

Theorem 1.3.1. *[70] Given a matrix A and its singular value decomposition is $A = U\Sigma V^\top = Udiag(\sigma_1, \ldots, \sigma_R, 0, \ldots, 0)V^\top$, then for any r with $0 \leq r \leq R$, $A_r = \sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^T$ is the global minimum of*

$$A_r = argmin_{rank(B)\leq r} \|A - B\|_F,$$

where $\mathbf{u}_i$ is the i-th column of U while $\mathbf{v}_i$ is the i-th column of V.

2) Best rank-1 approximation of a general tensor.

For a high order tensor, it has been theoretically guaranteed to have a best rank-1 approximation, see [65].

Corollary 1.3.1. *[65] Every tensor has a best rank-1 approximation.*

3) Best rank-R ($R > 1$) approximation of a general tensor.

It has been pointed out that the best low rank approximation for high-order tensors may not exist at all [65, 114, 116, 126, 167, 169]. Here we list two examples. A rank-3 tensor fails to have a best rank-2 approximation and a rank-6 tensor fails to have a best rank-5 approximation.

Example 1 [65] Let $\mathbf{u}_1, \mathbf{v}_1 \in \mathbb{R}^{I_1}$, $\mathbf{u}_2, \mathbf{v}_2 \in \mathbb{R}^{I_2}$, and $\mathbf{u}_3, \mathbf{v}_3 \in \mathbb{R}^{I_3}$ be vectors such that each pair $\mathbf{u}_i, \mathbf{v}_i$ is linearly independent. Define tensor

$$T := \mathbf{u}_1 \otimes \mathbf{u}_2 \otimes \mathbf{v}_3 + \mathbf{u}_1 \otimes \mathbf{v}_2 \otimes \mathbf{u}_3 + \mathbf{v}_1 \otimes \mathbf{u}_2 \otimes \mathbf{u}_3 \in \mathbb{R}^{I_1 \times I_2 \times I_3},$$

and for each $n \in \mathbb{N}$,

$$T_n := n\left(\mathbf{u}_1 + \frac{1}{n}\mathbf{v}_1\right) \otimes \left(\mathbf{u}_2 + \frac{1}{n}\mathbf{v}_2\right) \otimes \left(\mathbf{u}_3 + \frac{1}{n}\mathbf{v}_3\right) - n\mathbf{u}_1 \otimes \mathbf{u}_2 \otimes \mathbf{u}_3.$$

Then T has rank 3 and rank of T_n is at most 2. But $\|T_n - T\| \to 0$ as $n \to \infty$. Therefore, T does not have a best rank-2 approximation.

Example 2 [23] Let $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4$ be linearly independent. Define

$$T := \mathbf{u}_1 \otimes \mathbf{u}_1 \otimes \mathbf{u}_1 + \mathbf{u}_1 \otimes \mathbf{u}_2 \otimes \mathbf{u}_3 + \mathbf{u}_2 \otimes \mathbf{u}_3 \otimes \mathbf{u}_1 + \mathbf{u}_2 \otimes \mathbf{u}_4 \otimes \mathbf{u}_3 + \mathbf{u}_3 \otimes \mathbf{u}_1 \otimes \mathbf{u}_2 + \mathbf{u}_3 \otimes \mathbf{u}_2 \otimes \mathbf{u}_4$$

and, for $\epsilon > 0$,

$$T_\epsilon : \ = \ (\mathbf{u}_2 + \epsilon\mathbf{u}_1) \otimes (\mathbf{u}_2 + \epsilon\mathbf{u}_4) \otimes \epsilon^{-1}\mathbf{u}_3 + (\mathbf{u}_3 + \epsilon\mathbf{u}_1) \otimes \epsilon^{-1}\mathbf{u}_1 \otimes (\mathbf{u}_1 + \epsilon\mathbf{u}_2)$$

$$-\epsilon^{-1}\mathbf{u}_2 \otimes \mathbf{u}_2 \otimes (\mathbf{u}_1 + \mathbf{u}_3 + \epsilon\mathbf{u}_4) - \epsilon^{-1}\mathbf{u}_3 \otimes (\mathbf{u}_1 + \mathbf{u}_2 + \epsilon\mathbf{u}_3) \otimes \mathbf{u}_1$$
$$+\epsilon^{-1}(\mathbf{u}_2 + \mathbf{u}_3) \otimes (\mathbf{u}_2 + \epsilon\mathbf{u}_3) \otimes (\mathbf{u}_1 + \epsilon\mathbf{u}_4).$$

Then rank of T_ϵ is at most 5 and T is rank 6. But $\|T_\epsilon - T\| \to 0$ as $\epsilon \to 0$. T has no optimal approximation by tensors of $rank \leq 5$.

Specifically, Silva and Lim [65] showed that the problem of optimal low rank approximation of higher order tensors is ill-posed for many ranks and arbitrary order ($k \geq 3$). Thus an optimal solution for CP approximation need not exist. Apart from the CP model, such failure also occur in other component models, see [68,117,168]. Moreover, [51] indicated that this phenomenon can extend to symmetric tensors. Many examples of such failure can also be found in [108,153,190]. We conclude the ill-posedness by restating a general result in [65].

Theorem 1.3.2. *[65] For $k \geq 3$ and $I_1, I_2, \ldots, I_k \geq 2$, there exists a tensor $T \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_k}$ of rank $r + s$ that has no optimal rank-r approximation, for any r and $s \geq 1$ satisfying $2s \leq r \leq min\{I_1, I_2, \ldots, I_k\}$.*

1.3.2 Solution

Since the best low rank approximation may not exist, the question is how to avoid this failure. Imposing an extra requirement of orthogonality [40,116] changes the approximation problem (1.5) to an optimization over a compact set, thus it guarantees the existence of global optimum. Kolda [110] investigated various orthogonal conditions related to different definitions of orthogonality, including orthogonality, complete orthogonality and strong orthogonality which will be discussed in detail in Chapter 4. Here we briefly introduce complete orthogonality [40, 110, 114] and semi-orthogonality [40,166,190].

[116] proved the following theorem to show that the CP approximation (1.5) does attain its infimum under semi-orthogonality constraint.

Theorem 1.3.3. *[116] There exists a solution to the CP approximation problem (1.5) subject to semi-orthogonality constraint ($\langle \mathbf{u}_{r_1}^{(i)}, \mathbf{u}_{r_2}^{(i)} \rangle = \delta_{r_1 r_2}$ for one i from*

$\{1, \ldots k\}$).

In 2009, [40] discussed a form of low rank approximation with diagonal core with complete orthogonality constraint, i.e., complete CP approximation and proved its existence of global optimum theoretically.

Theorem 1.3.4. *[40] There exists a solution to the CP approximation problem (1.5) subject to complete orthogonality constraint ($\langle \mathbf{u}_{r_1}^{(i)}, \mathbf{u}_{r_2}^{(i)} \rangle = \delta_{r_1 r_2}$ for all $i = 1, \ldots, k$).*

Imposing orthogonality is not only for the theoretical purpose, but also has many applications in signal processing, wireless communication systems, blind signal separation and identification, and independent component analysis [167, 180].

1.4 Algorithms and convergence analysis

1.4.1 Best rank-1 approximation

Many efforts for finding the best rank-1 approximation of a general tensor (1.3) have been made in the literature, yet the problem is still not settled. See, for example, [24, 95, 109, 112, 114, 189, 197]. The difficulty is partly due to the curse of dimensionality, whence the rapid growth of computational overhead, and partly the nonlinearity, whence the stagnation at a local solution. For example, the alternating least squares (ALS) method works on improving one factor a time. Assuming the form as a high-order power method, the ALS is easy to implement and has been conventionally employed as the workhorse for low rank tensor approximation. However, the method suffers from slow convergence and easy stagnation at a local solution. Thus it is appealing that maybe alternating two factors simultaneously by employing the singular value decomposition (SVD) as the two-in-one optimization mechanism could result in better performance. The idea was mentioned in [63, Section 3.3] with no particular elaboration, and was more carefully postulated in [75] with numerical testing on some synthetic and real data sets of third-order tensors. This approach has the obvious advantage that, starting from the same point, one step of SVD-based iteration is superior to two consecutive steps of ALS iteration. There is no theory at present to support that the improvement by the SVD-based iteration will continue to be superior in the long run. Through numerical experiments, however, it has been suggested that for large scale data the SVD-based method might have better limiting behavior leading to better approximations [75, Section 5].

1.4.2 Best rank-1 approximation of a symmetric tensor

Symmetric tensor

A given tensor $T \in \mathbb{R}^{n\times\cdots\times n}$ with elements $\tau_{i_1,\ldots,i_k}$ is said to be symmetric if $\tau_{i_1,\ldots,i_k} = \tau_{i_{\sigma(1)},\ldots,i_{\sigma(k)}}$ with respect to all possible permutations σ over the integers $\{1,\ldots k\}$.

Consider the best rank-1 approximation of a symmetric tensor (1.3), it was conjectured in [154] and proved that the best symmetric rank-1 approximation to a symmetric tensor is its best rank-1 approximation [198, Theorem 2.1]. The proof was by induction. However, a more correct way of stating this result is that the best rank-1 approximation to a symmetric tensor "can be chosen" symmetric [74, Theorem 9], because there might be non-symmetric best rank-1 approximations [74, Section 4] for a symmetric tensor. Even more precisely, except for symmetric tensors lying on a specific real algebraic variety, a generic symmetric tensor has a unique rank-1 approximation which, hence, is symmetric [74]. With all these being said, we make an interesting remark that perhaps it was Stefan Banach who first noted in the context of homogeneous polynomials [16] that a best rank-1 approximation of a symmetric tensor could be chosen to be symmetric. Discussions on different aspects of rank-1 approximation to symmetric tensors can be found in [17, 50, 99, 109, 148]. Research endeavor on this subject is still ongoing. See, for example, some more recent work in [55, 182].

1.4.3 Existing algorithms for CP approximation

Now we introduce the algorithms about finding the best low rank CP approximation. The first method is to generate singular value decomposition (SVD) for a matrix into high order SVD for a tensor. Eckart and Young [70] showed that the best rank-R approximation of a matrix is given by truncated SVD and the orthogonality is an inherent property. However, this type of result does not hold for higher-order tensors [40, 62, 96, 109, 111, 126, 130, 135, 152, 180]. Second idea is to repeat R times of best rank-1 approximation to form the rank-R approximation since the best rank-1 always exists. The detailed way is firstly to compute an optimal rank-1 approximation and subtract it from the original tensor, yielding the so-called residual tensor. Then repeat the process until the rank-R approximation is obtained. But Kolda [110] provided a counter example to show that this approximation is not the best.

Assuming the number of components is fixed, the ”workhorse” algorithm for computing a low rank CP decomposition has been the alternating least squares (ALS) algorithm [32, 52, 82, 114, 173]. Then, its local convergence has been established in [179] under additional conditions. Later on, it has been proved that both objective value and the iterates generated by the ALS method for low rank approximation converge globally for almost all tensors in [190]. Notice that this low rank CP approximation has addressed semi-orthogonality constraint by proposing polar decomposition [40,157]. Under the same constraint, [166] developed two new numerical approaches – simultaneous matrix diagonalization (SD-CPO) and ALS applied to the combined mode matrices (ALS-CPO).

However, there are some shortcomings about ALS method although it is simple to implement. The convergence is quite slow and it cannot be guaranteed to converge to a global minimum. The final solution is heavily dependent on the starting point which is another interesting problem on how to choose a good starting point. Later on, some papers [174, 175] work on the strategies to improve the efficiency of ALS such as line searches [149, 156], Tikhonov regularization [147]. Recently, the comparison among ALS with other methods such as alternating slicewise diagonalization (ASD) method [98], damped Gauss-Newton (dGN) and PMF3 [151] can be found in [28, 72].

1.4.4 Existing algorithms for Tucker nearest problem

We briefly introduce some popular algorithms which are applied to compute Tucker nearest problem. For example, ALS [32, 82], Tucker1 method [178] also named as higher order singular value decomposition (HOSVD) [62], TUCKALS3 [120] and its extension [101], higher order orthogonal iteration (HOOI) method [63] and its improvement [30], Newton-Grassmann method [71], differential-geometric Newton method [97] and MBI method [37]. For more discussions on the Tucker problem, see [114].

1.5 Road map and our contributions

We provide the outline of this thesis below.

1.5.1 Synopsis of Chapter 2: symmetric best rank-1 approximation

We provide three algorithms based on the singular value decomposition (SVD) that modify two factors a time to find the best rank-1 approximation to a symmetric tensor. Comparing with existing alternating least squares (ALS) technique which improves one factor a time, one step of SVD-based iteration is superior to two steps of ALS iterations. Noting that generically the best rank-1 approximation to a symmetric tensor is symmetric. We prove that not only the generalized Rayleigh quotients generated from the three SVD-based algorithms enjoy monotone convergence, but also that the iterates themselves converge.

1.5.2 Synopsis of Chapter 3: non-symmetric best rank-1 approximation

The problem is to find the best rank-1 approximation to a generic tensor. The main focus of this chapter is on providing a mathematical proof for the convergence of the iterates of SVD-based algorithms. The ALS method is easy to implement, but suffers from slow convergence and easy stagnation at a local solution. It has been suggested that the SVD-algorithm might have a better limiting behavior leading to better approximations. We provide a rigorous mathematical proof for the convergence of iterates and our approach relies on only the continuity of singular vectors and real analysis.

1.5.3 Synopsis of Chapter 4: orthogonal low rank approximation

In this chapter, we study the orthogonal low rank approximation problem of tensors in the general setting in the sense that more than one matrix factor is required to be mutually orthonormal, which includes the completely orthogonal low rank approximation and semi-orthogonal low rank approximation as two special cases. It has been addressed in [190] that ”the question of more than one semi-orthogonal factor matrix, except for the case of complete orthogonality, remains open”. To deal with this open question we present an SVD-based algorithm. Our SVD-based algorithm updates two vectors simultaneously and maintains the required orthogonality conditions by means of the polar decomposition. The convergence behavior of our algorithm is analyzed for both objective function and iterates themselves and is illustrated by numerical experiments.

1.5.4 Synopsis of Chapter 5: general convergence of ADM

For problems involving multiple variables, the notion of solving a sequence of simplified problems by fixing all but one variable a time and alternating among the variables (ADM) has been exploited in a wide range of applications. We propose a general framework that can be applied to prove convergence for many types of alternating direction methods. The conditions are mild and easy to satisfy, so the theory should be of fundamental significance to many algorithms. Its application to a variety of important algorithms is demonstrated below.

Applications to some known cases:

- The Gauss-Seidel method for solving a system of linear equations.
- The power method for finding the dominant eigenvector.
- The alternating least squares method for computing the QR decomposition.
- The alternating projection method for finding structured low rank matrices.
- Best rank-one tensor approximation.
- Tucker nearest problem.
- Structured Kronecker approximation.

1.5.5 Synopsis of Chapter 6: conclusion

1.5.6 Synopsis of Chapter 7: list of author's publications

Chapter 2

Symmetric Best Rank-1 Approximation

2.1 Introduction

A real-valued tensor of order k can be represented by a k-way array

$$T = [\tau_{i_1,\dots,i_k}] \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_k}$$

with elements $\tau_{i_1,\dots,i_k}$ accessed via k indices. A tensor of the form

$$\bigotimes_{\ell=1}^{k} \mathbf{u}^{(\ell)} = \mathbf{u}^{(1)} \otimes \dots \otimes \mathbf{u}^{(k)} := \left[u_{i_1}^{(1)} \dots u_{i_k}^{(k)} \right], \tag{2.1}$$

where elements are the products of entries from vectors $\mathbf{u}^{(\ell)} \in \mathbb{R}^{I_\ell}$, $\ell = 1, \dots, k$, is said to be of rank one. When $I_1 = \dots = I_k$, we have a square tensor. An order-k square tensor T is said to be symmetric if

$$\tau_{i_1,\dots,i_k} = \tau_{i_{\sigma(1)},\dots,i_{\sigma(k)}} \tag{2.2}$$

with respect to all possible permutations σ over the integers $\{1, \dots k\}$. A symmetric rank-1 tensor therefore necessarily implies that $\mathbf{u}^{(\ell)} = c_\ell \mathbf{u}^{(1)}$ for some scalar c_ℓ, $\ell = 2, \dots, k$. In this case, we denote $I_1 = \dots = I_k = n$ and write $\bigotimes_{\ell=1}^{k} \mathbf{u} = \mathbf{u}^k$.

The problem of finding a best rank-1 approximation to T is to determine unit vectors $\mathbf{u}^{(\ell)} \in \mathbb{R}^{I_\ell}$, $\ell = 1, \dots k$, and a scalar λ such that the functional

$$f\left(\lambda, \mathbf{u}^{(1)}, \dots, \mathbf{u}^{(k)}\right) := \left\| T - \lambda \bigotimes_{\ell=1}^{k} \mathbf{u}^{(\ell)} \right\|_F^2 = \sum_{i_1, i_2, \dots, i_k} \left(\tau_{i_1,\dots,i_k} - \lambda u_{i_1}^{(1)} \dots u_{i_k}^{(k)} \right)^2 \tag{2.3}$$

is minimized. For any fixed unit vectors $\mathbf{u}^{(1)},\ldots,\mathbf{u}^{(k)}$, the optimal value of λ for (2.3) is given precisely by the length of the projection of the "vector" T onto the direction of the "unit vector" $\bigotimes_{\ell=1}^{k}\mathbf{u}^{(\ell)}\in\mathbb{R}^{I_1\times I_2\times\ldots\times I_k}$, i.e.,

$$\lambda=\lambda\left(\mathbf{u}^{(1)},\ldots,\mathbf{u}^{(k)}\right)=\left\langle T,\bigotimes_{\ell=1}^{k}\mathbf{u}^{(\ell)}\right\rangle. \tag{2.4}$$

Thus, minimizing the orthogonal component of T, as is desired in (2.3), is equivalent to maximizing the length $|\lambda|$ of the parallel component. In [197], the expression (2.4) is called the generalized Rayleigh quotient of T corresponding to $\{\mathbf{u}^{(1)},\ldots,\mathbf{u}^{(k)}\}$. Many approaches for finding the extreme values of (2.4) have been proposed in the literature. See, for example, [24, 75, 95, 109, 112, 114]. Switching the signs of the variables $\mathbf{u}^{(\ell)}$, if necessary, we may restrict our attention without loss of generality to the case that $\lambda > 0$ only.

2.1.1 Summary

This chapter contains two parts. First, we offer a simple argument that the symmetry of the best rank-1 approximation for a generic symmetric tensor can easily be understood with the notion of conventional singular value decomposition (SVD) for matrices. Second, we turn that argument into iterative SVD-based algorithms for computing the symmetric best rank-1 approximation. Our main focus is on the second part where we offer a convergence analysis that is new in the literature.

2.1.2 Outline of the chapter

This chapter is organized as follows. We begin with the introduction of some notations and basic facts in Section 2.2. Then, using well known properties of the SVD, we argue in a very concise way for the symmetry of the best rank-1 approximation in Section 2.3. Depending on how the permutations are chosen when applying the SVD successively to increase the objective value, we propose three algorithms for computing the best rank-1 approximation in Section 2.4. Among these, Algorithm 1 with cyclic permutation and Algorithm 2 with pre-assigned random permutation are formulated due to their theoretical simplicity. In turn, they highly motivate Algorithm 3 with post-assigned random permutation which is the easiest to implement. Convergence analysis is given in Section 2.5 where we first prove that the generalized Rayleigh quotients of all three algorithms converge monotonically and then detail the limiting behavior of the iterates generated by Algorithm 3 and Algorithm 1. Some numerical examples together with some interesting observations are presented in Section 2.6.

2.2 Basics

Tensors are multi-dimensional arrays. Thus, there are multiple ways to define tensor multiplications. Their appearances are often rather complex and perplexing. To facilitate the subsequence discussion, we first introduce a simple notation system that generalizes what we already know from the matrix theory. We also establish a few useful tools.

2.2.1 Multi-indexing

Suppose that the set $[\![k]\!]$ is partitioned as the union of two disjoint nonempty subsets $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_s\}$ and $\boldsymbol{\beta} = \{\beta_1, \ldots, \beta_t\}$, where $s + t = k$. We sometimes abbreviate $\boldsymbol{\beta} = \{\boldsymbol{\alpha}\}^{\mathrm{C}}$ since $\boldsymbol{\beta}$ is a complement of $\boldsymbol{\alpha}$. Choosing various ways to partition $[\![k]\!]$ offers us a convenient tool to dissect a high-dimensional array T and exam its cross-sections from different perspectives. For instance, an element in the tensor $T\mathbb{R}^{I_1 \times I_2 \times \ldots \times I_k}$ can be identified as $\tau_{[\mathcal{I}|\mathcal{J}]}^{(\boldsymbol{\alpha},\boldsymbol{\beta})}$ where $\mathcal{I} := (i_1, \ldots, i_s)$ and $\mathcal{J} := (j_1, \ldots, j_t)$ contain those indices at locations $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, respectively. Each index in the arrays $\mathcal{I}$ and $\mathcal{J}$ should be within the corresponding range of integers, e.g., $i_1 \in [\![I_{\alpha_1}]\!]$ and so on. In a sense, the subsets $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are to replace the role of row or column in matrices for tensors.

Operator product. Given a partitioning $(\boldsymbol{\alpha},\boldsymbol{\beta})$, the $\boldsymbol{\beta}$-product of T with a tensor $A \in \mathbb{R}^{I_{\alpha_1}\times\ldots\times I_{\alpha_s}}$ is defined to be

$$\mathscr{T}_{\boldsymbol{\beta}}(A) := T \circledast_{\boldsymbol{\beta}} A = \left[\langle \tau^{(\boldsymbol{\alpha},\boldsymbol{\beta})}_{[:|\mathcal{J}]}, A\rangle\right] \in \mathbb{R}^{I_{\beta_1}\times\ldots\times I_{\beta_t}}, \tag{2.5}$$

where

$$\langle \tau^{(\boldsymbol{\alpha},\boldsymbol{\beta})}_{[:|\mathcal{J}]}, A\rangle := \sum_{i_1=1}^{I_{\alpha_1}} \cdots \sum_{i_s=1}^{I_{\alpha_s}} \tau^{(\boldsymbol{\alpha},\boldsymbol{\beta})}_{[i_1,\ldots,i_s|\mathcal{J}]} a_{i_1,\ldots,i_s} \tag{2.6}$$

is the Frobenius inner product generalized to multi-dimensional arrays. In terms of the multi-index notation, the $\mathcal{J}$-th entry of $\mathscr{T}_{\boldsymbol{\beta}}(A)$ can be interpreted as

$$(\mathscr{T}_{\boldsymbol{\beta}}(A))_{\mathcal{J}} = \sum_{\mathcal{I}} \tau^{(\boldsymbol{\beta},\boldsymbol{\alpha})}_{[\mathcal{J}|\mathcal{I}]} a_{\mathcal{I}}, \tag{2.7}$$

where the summation of $\mathcal{I}$ runs through appropriate ranges of the indices $i_1,\ldots,i_{\alpha_s}$. In this way, the tensor T is considered as matrix representation of the linear map $\mathscr{T}_{\boldsymbol{\beta}}$ from $\mathbb{R}^{I_{\alpha_1}\times\ldots\times I_{\alpha_s}}$ to $\mathbb{R}^{I_{\beta_1}\times\ldots\times I_{\beta_t}}$ and the tensor-to-tensor operation $\circledast_{\boldsymbol{\beta}}$ defined in (2.5) generalizes the conventional matrix-to-vector multiplication.

2.2.2 Basic Lemmas

Each of the following sequence of results is elementary but together they are helpful tool for algebraic manipulations throughout the discussion.

Lemma 2.2.1. *Given a general tensor $T \in \mathbb{R}^{I_1\times I_2\times\ldots\times I_k}$, a partitioning $[\![k]\!] = \boldsymbol{\alpha}\cup\boldsymbol{\beta}$, and vectors $\mathbf{u}^{(\ell)} \in \mathbb{R}^{I_\ell}$, $\ell = 1,\ldots,k$, then it holds that*

$$\left\langle T, \bigotimes_{\ell=1}^{k} \mathbf{u}^{(\ell)} \right\rangle = \left\langle T \circledast_{\boldsymbol{\beta}} \bigotimes_{i=1}^{s} \mathbf{u}^{(\alpha_i)}, \bigotimes_{j=1}^{t} \mathbf{u}^{(\beta_j)} \right\rangle. \tag{2.8}$$

Proof. Based on the definition (2.5), the right hand side of (2.8) is simply a rearrangement of terms in the summation by the associative law. □

Lemma 2.2.2. *Given a general tensor $T \in \mathbb{R}^{I_1\times I_2\times\ldots\times I_k}$, arbitrary vectors $\mathbf{u}^{(\alpha_i)} \in \mathbb{R}^{I_{\alpha_i}}$, $i = 1,\ldots,k-2$, and $\mathbf{v} \in \mathbb{R}^{I_{\beta_2}}$, then*

$$\left(T \circledast_{\{\beta_1,\beta_2\}} \bigotimes_{i=1}^{k-2} \mathbf{u}^{(\alpha_i)} \right) \mathbf{v} = \left(T \circledast_{\{\beta_1,\alpha_j\}} \bigotimes_{i=1}^{j-1} \mathbf{u}^{(\alpha_i)} \otimes \mathbf{v} \otimes \bigotimes_{i=j+1}^{k-2} \mathbf{u}^{(\alpha_i)} \right) \mathbf{u}^{(\alpha_j)} \tag{2.9}$$

for any $j \in [\![k-2]\!]$.

Proof. The notion of a tensor entry $\tau^{(\boldsymbol{\alpha},\boldsymbol{\beta})}_{[\mathcal{I}|\mathcal{J}]}$ defined earlier will be informative here for tracking which index is being associated with which location. The μ-th entry of the vector $\left(T \circledast_{\boldsymbol{\beta}} \bigotimes_{i=1}^{k-2} \mathbf{u}^{(\alpha_i)}\right) \mathbf{v}$ is given by

$$\begin{aligned}
& \sum_{\nu=1}^{I_{\beta_2}} \left(\sum_{i_1=1}^{I_{\alpha_1}} \cdots \sum_{i_{k-2}=1}^{I_{\alpha_{k-2}}} \tau^{(\boldsymbol{\alpha},\boldsymbol{\beta})}_{[i_1,\ldots,i_{k-2}|\mu,\nu]} u^{(\alpha_1)}_{i_1} \ldots u^{(\alpha_{k-2})}_{i_{k-2}} \right) v_\nu \\
= & \sum_{i_1=1}^{I_{\alpha_1}} \cdots \sum_{i_{k-2}=1}^{I_{\alpha_{k-2}}} \sum_{\nu=1}^{I_{\beta_2}} \tau^{(\boldsymbol{\alpha} \bigcup \{\beta_2\},\{\beta_1\})}_{[i_1,\ldots,i_{k-2},\nu|\mu]} u^{(\alpha_1)}_{i_1} \ldots u^{(\alpha_{k-2})}_{i_{k-2}} v_\nu \\
= & \sum_{i_j=1}^{I_{\alpha_j}} \left(\sum_{i_1=1}^{I_{\alpha_1}} \cdots \sum_{i_{k-2}=1}^{I_{\alpha_{k-2}}} \sum_{\nu=1}^{I_{\beta_2}} \tau^{(\boldsymbol{\alpha} \bigcup \{\beta_2\}-\{\alpha_j\},\{\beta_1,\alpha_j\})}_{[i_1,\ldots,i_{k-2},\nu|\mu,i_j]} u^{(\alpha_1)}_{i_1} \ldots u^{(\alpha_{k-2})}_{i_{k-2}} v_\nu \right) u^{(\alpha_j)}_{i_j},
\end{aligned}$$

where the last equality is obtained by the associative law so that the summation inside the parentheses contains no $u^{(\alpha_j)}_{i_j}$ terms. □

Lemma 2.2.3. *Given a symmetric tensor $T \in \mathbb{R}^{n \times \cdots \times n}$ of order k and two fixed positive integers s and t with $k = s + t$, let $\mathbf{v}^{(\ell)} \in \mathbb{R}^n$, $\ell = 1, \ldots, s$, be arbitrary vectors. Then the product $T \circledast_{\boldsymbol{\beta}} \bigotimes_{i=1}^{s} \mathbf{v}^{(\rho_i)}$ is a symmetric tensor of order t and is independent of any permutation ρ of $[\![s]\!]$ and any subset $\boldsymbol{\beta} \subset [\![k]\!]$ with cardinality t.*

Proof. Let $[\![k]\!] = \boldsymbol{\alpha} \cup \boldsymbol{\beta}$ be a partitioning where, without loss of generality, indices in $\boldsymbol{\beta}$ are arranged in ascending order. Then for $\ell_j \in [\![n]\!]$, $j = 1, \ldots, t$, we have

$$\begin{aligned}
\left(T \circledast_{\boldsymbol{\beta}} \bigotimes_{i=1}^{s} \mathbf{v}^{(\rho_i)} \right)_{\ell_1,\ldots,\ell_t} & = \sum_{i_1=1}^{n} \cdots \sum_{i_s=1}^{n} \tau^{(\boldsymbol{\alpha},\boldsymbol{\beta})}_{[i_1,\ldots,i_s|\ell_1,\ldots,\ell_t]} v^{(\rho_1)}_{i_1} \ldots v^{(\rho_s)}_{i_s} \\
& = \sum_{i_1=1}^{n} \cdots \sum_{i_s=1}^{n} \tau_{i_1,\ldots,i_s,\ell_1,\ldots,\ell_t} v^{(\rho_1)}_{i_1} \ldots v^{(\rho_s)}_{i_s} \\
& = \sum_{i_{\rho^{-1}(1)}=1}^{n} \cdots \sum_{i_{\rho^{-1}(s)}=1}^{n} \tau_{i_{\rho^{-1}(1)},\ldots,i_{\rho^{-1}(s)},\ell_1,\ldots,\ell_t} v^{(1)}_{i_{\rho^{-1}(1)}} \ldots v^{(s)}_{i_{\rho^{-1}(s)}}.
\end{aligned}$$

In the above, the symmetry of T implies that the location of $\boldsymbol{\beta}$ is immaterial and thus the second equation is obtained by moving $\ell_1, \ldots, \ell_t$ to the end of the index array, whereas ρ^{-1} denotes the inverse of the permutation ρ. By renaming $i_{\rho^{-1}(j)}$ as i_j, $j = 1, \ldots, s$, we see that the reference to ρ is also immaterial. □

Corollary 2.2.1. *Under the same condition of Lemma 2.2.3, the associative law*

$$T\circledast \bigotimes_{i=1}^{s} \mathbf{v}^{(\rho_i)} = \left(T\circledast \bigotimes_{i=1}^{j} \mathbf{v}^{(\rho_i)} \right) \circledast \bigotimes_{i=j+1}^{s} \mathbf{v}^{(\rho_i)} \tag{2.10}$$

holds for any $j \in [\![s]\!]$ and any permutation ρ of $[\![s]\!]$.

When the subset $\boldsymbol{\beta} \subset [\![k]\!]$ is of cardinality 2, then Lemma 2.2.3 can be generalized to arbitrary tensor of order $k-2$.

Lemma 2.2.4. *Given a symmetric tensor $T \in \mathbb{R}^{n\times\cdots\times n}$ of order k is symmetric and a subset $\boldsymbol{\beta} \subset [\![k]\!]$ with cardinality 2, then, with respect to any tensor $S \in \mathbb{R}^{n\times\cdots\times n}$ of order $k-2$, the product $T\circledast_{\boldsymbol{\beta}} S$ is a symmetric matrix.*

Proof. For convenience, write $M := T\circledast_{\boldsymbol{\beta}} S$. By the definition (2.5), M is a matrix. Observer that

$$m_{ij} = \langle \tau_{[:|i,j]}, S\rangle = \langle \tau_{[:|j,i]}, S\rangle = m_{ji},$$

because $\tau_{[:|i,j]} = \tau_{[:|j,i]}$ by the symmetry of T. □

For latter usage in our proof for convergence, we also need the following results from real analysis.

Lemma 2.2.5. *Let $\{a_k\}$ be a bounded sequence of real numbers with the property $|a_{k+1} - a_k| \to 0$ as $k \to \infty$. If the accumulation points for the sequence are isolated, then $\{a_k\}$ converges to a unique limit point.*

Proof. Suppose $\{a_{\alpha_k}\}$ and $\{a_{\beta_k}\}$ are two subsequences of $\{a_k\}$ which converge, respectively, to two distinct limit points, x and y. Let z denote any fixed real number between x and y. For a positive number r, let $B_x(r)$ denote the neighborhood $[x-r, x+r]$ of x.

For any $0 < \epsilon < \frac{1}{4}\min\{|x-z|, |y-z|\}$, there exists a large enough integer $K = K(\epsilon)$ such that $a_{\alpha_k} \in B_x(\epsilon)$, $a_{\beta_k} \in B_y(\epsilon)$, and $|a_{k+1} - a_k| < \epsilon$ for all $k \geq K$. Infinitely many elements of $\{a_k\}$ must leave $B_x(\epsilon)$ to enter $B_y(\epsilon)$ and vise versa. By doing so, there is an infinite subsequence of $\{a_k\}$ contained in $B_z(\epsilon)$. This shows that z is also an accumulation point. Since z is arbitrary, we have shown any number between x and y is an accumulate point. This contradicts the assumption that the accumulation points are isolated. □

Lemma 2.2.5 asserts the uniqueness. A slight variation requiring a weaken assumption and resulting only a local convergence will also fit our need. The proof of following Lemma can be found in [140, Lemma 4.10], [198, Proposition 3.2] and [79, Lemma 6], but we prove it in another way.

Lemma 2.2.6. *Assume that a^* is an isolated accumulation point of a bounded sequence $\{a_k\}$ such that for every subsequence $\{a_{k_j}\}$ converging to a^*, there is an infinite subsequence $\{a_{k_{j_i}}\}$ such that $|a_{k_{j_i}+1} - a_{k_{j_i}}| \to 0$. Then the whole sequence $\{a_k\}$ converges to a^*.*

Proof. There are some ambiguities in the original proof [140, Lemma 4.10]. See also [198, Proposition 3.2]. We take this opportunity to clarify the dubiety.

We prove by contradiction. Suppose that the sequence $\{a_k\}$ does not converge to a^*. Since a^* is isolated, there exists a neighborhood $N_\epsilon(a^*) := \{x \in \mathbb{R} | |x - a^*| \leq \epsilon\}$ such that a^* is the only accumulation point of the sequence $\{a_k\}$. Choose $\{a_{k_j}\}$ be an arbitrary subsequence of $\{a_k\}$ contained in $N_\epsilon(a^*)$. For each j, let $\{a_{k_j}, a_{k_j+1}, \ldots, a_{\ell_j}\}$ be the largest consecutive segment of $\{a_k\}$ that starts at a_{k_j} and stays inside $N_\epsilon(a^*)$, i.e.,

$$\ell_j := \max\{\ell | \, |a_i - a^*| \leq \epsilon, i = k_j, k_j + 1, \ldots, \ell\}.$$

If ℓ_j is infinite, $|a_k - a^*| \leq \epsilon$ hold for all k large enough and for arbitrary ϵ which implies that $\{a_k\}$ converges to a^*. But this contradicts with assumption.

If ℓ_j is finite, by construction, the subsequence $\{a_{\ell_j}\}$ has the property

$$|a_{\ell_j} - a^*| \leq \epsilon, \quad |a_{\ell_j+1} - a^*| > \epsilon.$$

Since $\{a_{\ell_j}\}$ is a bounded sequence, a subsequence $\{a_{\ell_{j_i}}\}$ must converge. Notice that $\{a_{\ell_{j_i}}\}$ is contained in $N_\epsilon(a^*)$, and by the property of isolation, the limit point must be a^*. Therefore $|a_{\ell_{j_i}} - a^*| < \frac{\epsilon}{2}$ when i is large enough. In this way, we have found a convergent subsequence $\{a_{\ell_{j_i}}\}$, but element by element we always have the gap

$$|a_{\ell_{j_i}+1} - a_{\ell_{j_i}}| \geq |a_{\ell_{j_i}+1} - a^*| - |a_{\ell_{j_i}} - a^*| \geq \frac{\epsilon}{2}.$$

This is a contradiction to $|a_{\ell_{j_i}+1} - a_{\ell_{j_i}}| \to 0$. Therefore, the whole sequence $\{a_k\}$ converges to a^*. □

2.3 Symmetric best rank-1 approximation

We now argue that the best rank-1 approximation to a generic symmetric tensor is symmetric. We shall not assume a priori that the best rank-1 approximation is unique, nor that a symmetric best rank-1 approximation always exists. All we need is the following fundamental fact from matrix theory.

Lemma 2.3.1. *Given a matrix $A \in \mathbb{R}^{m\times n}$, then the global maximum of the generalized Rayleigh quotient*

$$\max_{\substack{\mathbf{y} \in \mathbb{R}^m, \|\mathbf{y}\| = 1 \\ \mathbf{z} \in \mathbb{R}^n, \|\mathbf{z}\| = 1}} \mathbf{y}^\top A\mathbf{z} \tag{2.11}$$

is precisely the largest singular value σ_1 of A, where the global maximizer $(\mathbf{y}_1, \mathbf{z}_1)$ consists of precisely the corresponding left and right singular vectors. The best rank-1 approximation to A is given by $\sigma_1\mathbf{y}_1\mathbf{z}_1^\top$. In the event that $A \in \mathbb{R}^{m\times m}$ is symmetric and that the largest singular value of A is simple, then $\mathbf{y} = \pm\mathbf{z}$ depending on the sign[1] *of the dominant eigenvalue $\lambda_1 = \pm\sigma_1$ and, hence, the best rank-1 approximation to A is symmetric.*

The condition that the largest singular value of A is simple is generic in the sense that the symmetric matrices with multiply eigenvalues form an algebraic variety of codimension two [56]. Consequently, the symmetric matrices that do not

[1] We shall use the symbol ± to indicate a proper sign selection in the subsequent discussion when there is no need to specify the sign.

have a unique symmetric best rank-1 approximation form an algebraic variety of codimension one [74, Lemma 3].

Built upon Lemma 2.3.1, we explain in the argument below the kind of generic property we need for a symmetric tensor. We justify the symmetry by comparing two components a time in the rank-1 tensor.

Suppose that $\lambda \bigotimes_{\ell=1}^{k} \overline{\mathbf{u}}^{(\ell)}$ is the best rank-1 approximation to a given order-k symmetric tensor T. By (2.4), the generalized Rayleigh quotient $\lambda = \left\langle T, \bigotimes_{\ell=1}^{k} \overline{\mathbf{u}}^{(\ell)} \right\rangle$ is positive and maximal. Consider the case $\boldsymbol{\beta} = \{1,2\}$. By Lemma 2.2.1, we can write

$$\lambda = \left\langle T \circledast_{\boldsymbol{\beta}} \bigotimes_{\ell=3}^{k} \overline{\mathbf{u}}^{(\ell)}, \overline{\mathbf{u}}^{(1)} \otimes \overline{\mathbf{u}}^{(2)} \right\rangle.$$

By Lemma 2.2.4, the matrix $C := T \circledast_{\boldsymbol{\beta}} \bigotimes_{\ell=3}^{k} \overline{\mathbf{u}}^{(\ell)}$ is symmetric. Assume that the largest singular value, which is λ, of C is simple. Then, by Lemma 2.3.1, we conclude that $\overline{\mathbf{u}}^{(1)} = \pm \overline{\mathbf{u}}^{(2)}$. Moving to the choice $\boldsymbol{\beta} = \{2,3\}$ and assuming again that λ is simple for the newly defined matrix C, we than have $\overline{\mathbf{u}}^{(2)} = \pm \overline{\mathbf{u}}^{(3)}$. Continuing this process, we conclude that $\overline{\mathbf{u}}^{(1)}, \ldots, \overline{\mathbf{u}}^{(k)}$ differ from each other by at most a negative sign. At the end, we may write $\lambda \bigotimes_{\ell=1}^{k} \overline{\mathbf{u}}^{(\ell)} = \pm \lambda \overline{\mathbf{u}}^{(1)^{k}}$. So the best rank-1 approximation to a symmetric tensor is necessarily symmetric.

2.4 Computation

The argument in the preceding section motivates an SVD-based way to calculate the symmetric best rank-1 approximation by iterations. The idea of using the SVD instead of the ALS is not new. It has been proposed for general tensors in [75], but so far as we know no convergence analysis has ever been established. The main contribution of this chapter is to furnish the proof of convergence for symmetric tensors.

2.4.1 Update with cyclic progression

The most basic approach is outlined in Algorithm 1. For efficiency, we also propose a modification by random permutations in Algorithm 2 followed by a more simplified Algorithm 3. Two types of dynamics are involved in all algorithms. One is the dynamics of the objective values, of which the analysis is straightforward. The other is the dynamics of the iterates, which is much harder to characterize. We will discuss the convergence in the next section.

To convey the idea, it is convenient to adopt the subscript $_{[p]}$ in Algorithm 1 to indicate the quantity at the p-th iteration. Each sweep of p at Line 1 in Algorithm 1 involves k pairs of $\boldsymbol{\beta}$ ranging circularly from $(1,2),(2,3),\ldots,(k,1)$. Thus, each $\mathbf{u}_{[p+1]}^{(\ell)}$ is updated twice. The first updates for $\ell=2,\ldots,k$, denoted by $\widehat{\mathbf{u}}_{[p+1]}^{(\ell)}$ at Line 10, are not essential and can be completely removed from the algorithm without affecting the calculation, but their presences help bridge the monotonicity in theory. The update $\widehat{\mathbf{u}}_{[p+1]}^{(1)}$ is temporarily overwritten as $\mathbf{u}_{[p+1]}^{(1)}$ at Line 9 for the computation of $C_{[p]}^{(\ell)}$ at Line 4 for $\ell=2,\ldots,k-1$, but will be updated again at Line 17. The switch of sign at Line 7 conditioned upon Line 6 is to ensure that the iterates will be aligned in one direction and thus avoid jumping back and forth. Also, by Lemma 2.2.3, the reference to $\boldsymbol{\beta}_\ell$ in the multiplication by $\circledast_{\boldsymbol{\beta}_\ell}$ at Line 4 is entirely unnecessary. We include it in the description to help keep track of the procedure. We register the intermediate values $\lambda_{[p+1]}^{(\ell)}$ as well, even though only $\lambda_{[p+1]}^{(k)}$ at the final stage is crucial.

The above algorithm is different from the alternating least squares (ALS) approach that has been popular for computing the best rank-1 approximation [50, 109, 197]. The most significant difference is that, since the dominant singular vector $\mathbf{u}_{[p+1]}^{(\ell)}$ of the matrix $C_{[p]}^{(\ell)}$ gives rise to the absolute maximal value $\lambda_{[p+1]}^{(\ell)}$ for the functional

$$g(\mathbf{x},\mathbf{y}) := \left\langle T, \bigotimes_{i=1}^{\ell-1} \mathbf{u}_{[p+1]}^{(i)} \otimes \mathbf{x} \otimes \mathbf{y} \otimes \bigotimes_{i=\ell+2}^{k} \mathbf{u}_{[p]}^{(i)} \right\rangle \tag{2.12}$$

among all possible vectors $\mathbf{x}$ and $\mathbf{y}$, the mechanism of updating $\mathbf{x}$ and $\mathbf{y}$ simultaneously in Algorithm 1 is going to increase the generalized Rayleigh quotient faster

Algorithm 1 (Best rank-1 approximation via SVD updating with cyclic progression.)

Input: An order-k, n-dimensional, symmetric tensor T and k starting unit vectors $\mathbf{u}_{[0]}^{(1)}, \ldots, \mathbf{u}_{[0]}^{(k)} \in \mathbb{R}^n$

Output: A local best rank-1 approximation to T

```
1:  for $p = 0, 1, \cdots,$ do
2:    for $\ell = 1, 2, \cdots, k-1,$ do
3:      $\boldsymbol{\beta}_\ell = (\ell, \ell+1)$
4:      $C_{[p]}^{(\ell)} = T \circledast_{\boldsymbol{\beta}_\ell} \bigotimes_{i=1}^{\ell-1} \mathbf{u}_{[p+1]}^{(i)} \otimes \bigotimes_{i=\ell+2}^{k} \mathbf{u}_{[p]}^{(i)}$
5:      $[\mathbf{u}, s, \mathbf{v}] = \mathsf{svds}(C_{[p]}^{(\ell)}, 1)$   {Dominant singular value triplet via Matlab routine svds}
6:      if $u_1 < 0$ then
7:        $\mathbf{u} = -\mathbf{u}$   {Assume the generic case that $u_1 \neq 0$; otherwise, use another entry.}
8:      end if
9:      $\mathbf{u}_{[p+1]}^{(\ell)} := \mathbf{u}$   {If $\ell = 1$, this is $\widehat{\mathbf{u}}_{[p+1]}^{(1)}$; otherwise this is the second update $\mathbf{u}_{[p+1]}^{(\ell)}$, if $2 \leq \ell < k$.}
10:     $\widehat{\mathbf{u}}_{[p+1]}^{(\ell+1)} := \mathbf{u}$   {Skipping this step will not affect $C_{[p]}^{(\ell+1)}$ at Line 4.}
11:     $\lambda_{[p+1]}^{(\ell)} := s$
12:   end for
13:   $\boldsymbol{\beta}_k = (k, 1)$
14:   $C_{[p]}^{(k)} = T \circledast_{\boldsymbol{\beta}_k} \bigotimes_{i=2}^{k-1} \mathbf{u}_{[p+1]}^{(i)}$
15:   $[\mathbf{u}, s, \mathbf{v}] = \mathsf{svds}(C_{[p]}^{(k)}, 1)$   {Dominant singular value triplet via Matlab routine svds}
16:   $\mathbf{u}_{[p+1]}^{(k)} := \mathbf{u}$   {Adjust the sign properly as in Line 6.}
17:   $\mathbf{u}_{[p+1]}^{(1)} := \mathbf{u}$
18:   $\lambda_{[p+1]}^{(k)} := s$
19: end for
```

Algorithm 2 (Best rank-1 approximation via SVD updating with randomization.)

Input: An order-k, n-dimensional, symmetric tensor T and k starting unit vectors $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(k)} \in \mathbb{R}^n$

Output: A local best rank-1 approximation to T

```
 1: $t \leftarrow 0$
 2: $\lambda_0 \leftarrow \left\langle T, \bigotimes_{\ell=1}^{k} \mathbf{u}^{(\ell)} \right\rangle$
 3: **repeat**
 4:   $t \leftarrow t + 1$
 5:   $\sigma \leftarrow$ random permutation of $\{1, \ldots, k\}$
 6:   $\boldsymbol{\beta}_t \leftarrow (\sigma_{k-1}, \sigma_k)$          {Randomly select two factors}
 7:   $C_t \leftarrow T \circledast_{\boldsymbol{\beta}_t} \bigotimes_{i=1}^{k-2} \mathbf{u}^{(\sigma_i)}$
 8:   $[\mathbf{u}_t, s_t, \mathbf{v}_t] = \mathsf{svds}(C_t, 1)$  {Dominant singular value triplet via Matlab routine
      svds}
 9:   **if** $(\mathbf{u}_t)_1 < 0$ **then**
10:     $\mathbf{u}_t = -\mathbf{u}_t$
11:   **end if**
12:   $\lambda_t \leftarrow s_t$
13:   $\mathbf{u}^{(\sigma_{k-1})}, \mathbf{u}^{(\sigma_k)} \leftarrow \mathbf{u}_t$
14: **until** $\lambda_t$ meets convergence criteria
```

than the combination of two applications of ALS approach to $\mathbf{x}$ followed by $\mathbf{y}$. The gain is also better than the maximum of updating $\mathbf{x}$ or $\mathbf{y}$ separately as that discussed in [75, Preposition 4].

2.4.2 Update with random permutation

An alternative way to cut short the iterates required by the ℓ-loop in Algorithm 1 is to shuffle the columns $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(k)}$ by a random permutation σ and generate a matrix C for updating. This randomized procedure is modified at Line 7 in Algorithm 2. To avoid confusion with data generated from Algorithm 1, we employ a slightly different notation when describing this algorithm. Again, by Lemma 2.2.3, the reference to $\boldsymbol{\beta}_t$ is irrelevant. For simplicity, we always choose to update the last two vectors $\mathbf{u}^{(\sigma_{k-1})}, \mathbf{u}^{(\sigma_k)}$ after the permutation. A classical result in probability theory asserts that the expected number of trials for a permutation to recur is $\frac{k(k-1)}{2}$. But by the time that a repetition occurs, the vectors $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(k)}$ should have been updated. If the convergence is ever to happen, the effect of reshuffling will gradually diminish.

2.4.3 Update with post-randomization

To carry out the permutation $\mathbf{u}^{(\sigma_i)}$ at Line 7 in Algorithm 2 is still cumbersome. Since the purpose of permutation is simply to mingle the vectors, we may consider the alternative by postponing the permutation to the end of calculation as is indicated in Algorithm 3. It can be argued that Algorithm 2 would be equivalent to Algorithm 3 in the sense that, if one could foresee the future permutation at Line 7 and prearrange the columns in the order $\left\{\mathbf{u}^{(\sigma_1)}, \ldots, \mathbf{u}^{(\sigma_k)}\right\}$ before Line 5 in Algorithm 3, then both algorithms would be using the same C_t. In reality, of course, such a rearrangement does not happen, so we distinguish the progress of the generalized Rayleigh quotient by a different notation μ_t. Though $\mu_0 = \lambda_0$ to begin with, this μ_t in general is not the same as the λ_t generated by Algorithm 2 when $t \geq 1$. Note the simplification at Line 5 in Algorithm 3 which utilizes only the first $k-2$ vectors. The permutation at Line 12 will help intermingle the vectors before the next step.

Another difference between Algorithm 2 and Algorithm 3 deserves noting. In Algorithm 2, the replacement at Line 13 does not interfere with the vectors $\mathbf{u}^{(\sigma_i)}$, $i \in [\![k-2]\!]$, used to define C_t at Line 7. But in Algorithm 3, the replacement at Line 12 may affect $0, 1$ or 2 many of the first $k-2$ vectors used to define C_t at Line 5.

Indeed, with probability $\frac{(k-2)(k-3)}{k(k-1)}$ the perturbation σ will ask to replace 2 such vectors, which is high when k is large. We may thus consider a subclass of Algorithm 3 by requiring the update at Line 12 be limited to $[\![k-2]\!]$. The limit points

Algorithm 3 (Best rank-1 approximation via SVD updating with post-randomization.)

Input: An order-k, n-dimensional, symmetric tensor T and k starting unit vectors $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(k)} \in \mathbb{R}^n$

Output: A local best rank-1 approximation to T

```
1:  $t \leftarrow 0$
2:  $\mu_0 \leftarrow \left\langle T, \bigotimes_{\ell=1}^{k} \mathbf{u}^{(\ell)} \right\rangle$
3:  repeat
4:      $t \leftarrow t + 1$
5:      $C_t \leftarrow T \circledast \bigotimes_{i=1}^{k-2} \mathbf{u}^{(i)}$
6:      $[\mathbf{u}_t, s_t, \mathbf{v}_t] = \mathsf{svds}(C_t, 1)$  {Dominant singular value triplet via Matlab routine svds}
7:      $\sigma \leftarrow$ random permutation of $\{1, \ldots, k\}$
8:      if $(\mathbf{u}_t)_1 < 0$ then
9:          $\mathbf{u}_t = -\mathbf{u}_t$
10:     end if
11:     $\mu_t \leftarrow s_t$
12:     $\mathbf{u}^{(\sigma_{k-1})}, \mathbf{u}^{(\sigma_k)} \leftarrow \mathbf{u}_t$      {Randomly replace two factors}
13: until $\mu_t$ meets convergence criteria
```

of $\{\mathbf{u}_t\}$ by this subclass iteration form a subset of those by the unmodified Algorithm 3. Our numerical experiments suggest that both versions have the same set of limit points.

2.4.4 Symmetric update

Finally, since the best rank-1 approximation of symmetric tensors is symmetric, all factors should be the same eventually. It is tempting to exploit the mechanism of keeping the symmetry at every iteration once an SVD is done. We outline the procedure in Algorithm 4. The contrast is at Line 5 where C_t is calculated based on

Algorithm 4 (Best rank-1 approximation via symmetric SVD.)

Input: An order-k, n-dimensional, symmetric tensor T and a starting unit vector $\mathbf{u}_0 \in \mathbb{R}^n$

Output: A local best rank-1 approximation to T

```
$t \leftarrow 0$
$\mu_0 \leftarrow \left\langle T, \bigotimes_{\ell=1}^{k} \mathbf{u}_0 \right\rangle$
repeat
    $t \leftarrow t + 1$
    $C_t \leftarrow T \circledast \bigotimes_{i=1}^{k-2} \mathbf{u}_0$        {Using the same factor for all}
    $[\mathbf{u}_t, s_t, \mathbf{v}_t] = \mathsf{svds}(C_t, 1)$  {Dominant singular value triplet via Matlab routine svds}
    if $(\mathbf{u}_t)_1 < 0$ then
        $\mathbf{u}_t = -\mathbf{u}_t$
    end if
    $\mu_t \leftarrow s_t$
    $u_0 \leftarrow \mathbf{u}_t$
until $\mu_t$ meets convergence criteria
```

one single factor $\mathbf{u}_t$. Indeed, a similar idea has been proposed in [109] as the symmetric high-order power method whose performance has been reported as poor. We shall demonstrate in our numerical experiment that Algorithm 4 does not perform competitively either. Though interesting, this algorithm is of little importance to us. We mention it in passing and we do not consider it as a contribution.

2.5 Convergence analysis

In this section, we analyze the convergence for the above algorithms. We first show the monotonicity of the generalized Rayleigh quotients. Most importantly, we argue that the iterates themselves also converge. The latter answers an open question on convergence analysis for the SVD-based methods, as it offers an understanding of what was declared as "we do not have a complete understanding when this will happen" in [75, Page 947]. We concentrate mainly on Algorithm 3 for its practicality in implementation. We also include the behavior of Algorithm 1 for its elegance in theory. The analysis of Algorithm 2 is left to interested readers.

2.5.1 Convergence of objective values

Because the SVD is involved, where the dominant singular value and singular vector are selected at each update, all three algorithms enjoy the property that the corresponding sequences of the generalized Rayleigh quotients are bounded and monotone increasing.

Lemma 2.5.1. *The scalars* $\left\{\lambda^{(\ell)}_{[p]}\right\}$ *generated in Algorithm 1 form a monotone convergent sequence for each* $\ell = 1, \ldots, k$ *and all converge to the same value.*

Proof. It suffices to prove the assertion for the case $\lambda^{(k)}_{[p]}$ because the following argument shows that all other cases are sandwiched in between.

By applying Lemma 2.3.1 to each of the matrices $C^{(\ell)}_{[p]}$ consecutively, we observe that at any stage of p the inequalities

$$\begin{aligned}
\lambda^{(k)}_{[p]} &= \left|\left\langle T, \bigotimes_{\ell=1}^{k} \mathbf{u}^{(\ell)}_{[p]} \right\rangle\right| = \left|\left\langle T \circledast_{\boldsymbol{\beta}_1} \bigotimes_{\ell=3}^{k} \mathbf{u}^{(\ell)}_{[p]}, \mathbf{u}^{(1)}_{[p]} \otimes \mathbf{u}^{(2)}_{[p]} \right\rangle\right| \\
&\leq \left|\left\langle T \circledast_{\boldsymbol{\beta}_1} \bigotimes_{\ell=3}^{k} \mathbf{u}^{(\ell)}_{[p]}, \widehat{\mathbf{u}}^{(1)}_{[p+1]} \otimes \widehat{\mathbf{u}}^{(2)}_{[p+1]} \right\rangle\right| \\
&= \lambda^{(1)}_{[p+1]} = \left|\left\langle T \circledast_{\boldsymbol{\beta}_2} \widehat{\mathbf{u}}^{(1)}_{[p+1]} \otimes \bigotimes_{\ell=4}^{k} \mathbf{u}^{(\ell)}_{[p]}, \widehat{\mathbf{u}}^{(2)}_{[p+1]} \otimes \mathbf{u}^{(3)}_{[p]} \right\rangle\right| \\
&\leq \left|\left\langle T \circledast_{\boldsymbol{\beta}_2} \widehat{\mathbf{u}}^{(1)}_{[p+1]} \otimes \bigotimes_{\ell=4}^{k} \mathbf{u}^{(\ell)}_{[p]}, \mathbf{u}^{(2)}_{[p+1]} \otimes \widehat{\mathbf{u}}^{(3)}_{[p+1]} \right\rangle\right| \\
&= \lambda^{(2)}_{[p+1]} \leq \ldots \\
&\leq \lambda^{(k-1)}_{[p+1]} = \left|\left\langle T, \widehat{\mathbf{u}}^{(1)}_{[p+1]} \otimes \bigotimes_{\ell=2}^{k-1} \mathbf{u}^{(\ell)}_{[p+1]} \otimes \widehat{\mathbf{u}}^{(k)}_{[p+1]} \right\rangle\right|
\end{aligned}$$

$$
\begin{aligned}
&= \left| \left\langle T \circledast_{\boldsymbol{\beta}_k} \bigotimes_{\ell=2}^{k-1} \mathbf{u}^{(\ell)}_{[p+1]}, \widehat{\mathbf{u}}^{(1)}_{[p+1]} \otimes \widehat{\mathbf{u}}^{(k)}_{[p+1]} \right\rangle \right| \\
&\leq \left| \left\langle T \circledast_{\boldsymbol{\beta}_k} \bigotimes_{\ell=2}^{k-1} \mathbf{u}^{(\ell)}_{[p+1]}, \mathbf{u}^{(1)}_{[p+1]} \otimes \mathbf{u}^{(k)}_{[p+1]} \right\rangle \right| = \lambda^{(k)}_{[p+1]}
\end{aligned}
$$

are always maintained. The monotone sequence $\left\{ \lambda^{(k)}_{[p]} \right\}$ is bounded above by $\|T\|_F$, so it must converge. The inequalities sandwich the sequences one after another, so their limit points must be the same. □

Lemma 2.5.2. *The scalars $\{\lambda_t\}$ generated in Algorithm 2 form a monotone convergent sequence.*

Proof. By applying Lemma 2.3.1, it is still true that at any stage of t we always have

$$
\begin{aligned}
\lambda_t &= \left| \left\langle T, \bigotimes_{\ell=1}^{k} \mathbf{u}^{(\ell)} \right\rangle \right| = \left| \left\langle T \circledast_{\boldsymbol{\beta}_t} \bigotimes_{i=1}^{k-2} \mathbf{u}^{(\sigma_i)}, \mathbf{u}^{(\sigma_{k-1})} \otimes \mathbf{u}^{(\sigma_k)} \right\rangle \right| \\
&\leq \left| \left\langle T \circledast_{\boldsymbol{\beta}_t} \bigotimes_{i=1}^{k-2} \mathbf{u}^{(\sigma_i)}, \mathbf{u}_{t+1} \otimes \mathbf{u}_{t+1} \right\rangle \right| = \lambda_{t+1},
\end{aligned}
$$

since $\mathbf{u}_{t+1}$ is the dominant singular vector of the matrix $T \circledast_{\boldsymbol{\beta}_t} \bigotimes_{i=1}^{k-2} \mathbf{u}^{(\sigma_i)}$. Again, the monotone sequence $\{\lambda_t\}$ is bounded above by $\|T\|_F$, so it must converge. □

Lemma 2.5.3. *The scalars $\{\mu_t\}$ generated in Algorithm 3 form a monotone convergent sequence.*

Proof. Suppose that $C_t = T \circledast \bigotimes_{i=1}^{k-2} \mathbf{u}^{(i)}$ has been defined in terms of vectors $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(k-2)}$ from the previous step. Suppose also that its dominant singular vector $\mathbf{u}_t$ has been calculated. Then

$$
\mu_t = \left| \left\langle T \circledast \bigotimes_{i=1}^{k-2} \mathbf{u}^{(i)}, \mathbf{u}_t \otimes \mathbf{u}_t \right\rangle \right|. \tag{2.13}
$$

To proceed to the next step, some of these columns $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(k-2)}$ might be replaced by $\mathbf{u}_t$. Let $\boldsymbol{\Theta}^o$ and $\boldsymbol{\Theta}^+$ denote subsets of $[\![k-2]\!]$ containing indices of those vectors that will not been changed and are to be updated, respectively. Depending on the current permutation σ_t, there are three possible scenarios – the cardinality $|\mathbf{\Theta}^+|$ of the set $\mathbf{\Theta}^+$ can be 0, 1, or 2. By Lemmas 2.2.1 and 2.2.3, we may write

$$\begin{aligned}
\mu_t &= \left|\left\langle T, \bigotimes_{i\in\mathbf{\Theta}^o} \mathbf{u}^{(i)} \otimes \bigotimes_{j\in\mathbf{\Theta}^+} \mathbf{u}^{(j)} \otimes \mathbf{u}_t^2 \right\rangle\right| = \left|\left\langle T \circledast \bigotimes_{i\in\mathbf{\Theta}^o} \mathbf{u}^{(i)} \otimes \mathbf{u}_t^{|\mathbf{\Theta}^+|}, \bigotimes_{j\in\mathbf{\Theta}^+} \mathbf{u}^{(j)} \otimes \mathbf{u}_t^{2-|\mathbf{\Theta}^+|} \right\rangle\right| \\
&\leq \left|\left\langle T \circledast \bigotimes_{i\in\mathbf{\Theta}^o} \mathbf{u}^{(i)} \otimes \mathbf{u}_t^{|\mathbf{\Theta}^+|}, \mathbf{u}_{t+1} \otimes \mathbf{u}_{t+1} \right\rangle\right| = \mu_{t+1}. \qquad (2.14)
\end{aligned}$$

In the above, we have adopted the notion that a factor of either $\bigotimes_{j\in\emptyset} \mathbf{u}^{(j)}$ or $\mathbf{u}_t^0$ means that it does not occur in the multiplication. □

2.5.2 Convergence of iterates in Algorithm 3

The above argument about the monotonicity of values $\lambda_{[p]}^{(k)}$, λ_t, or μ_t is interesting and important, but certainly not enough, because the convergence of objective values does not guarantee the convergence of iterates to a global minimizer nor even to a stationary point [114]. What need be done for both algorithms is to argue that generically the iterates of vectors themselves also converge. Since Algorithm 3 is our ultimate choice of scheme in view of its simplicity and effectiveness, we give a detailed account of its dynamical behavior in this subsection.

For clarity, enumerate the column vectors at the end of each t-loop by $\left\{\mathbf{u}_t^{(1)}, \ldots, \mathbf{u}_t^{(k)}\right\}$ in accordance with their original order. By construction, for $t \geq 1$, at least two of these vectors are identical and the rest are unchanged from the previous step. The goal is to prove that these columns converge to the same vector as t goes to infinity, regardless how the random permutation σ which varies in t takes place. Toward this end, we establish a sequence of results.

Since the dominant singular vectors $\mathbf{u}_t$ are always normalized to unit length, the sequence $\{\mathbf{u}_t\}$ must have a convergent subsequence. We first argue that its limit point must propagate to all elements of $\left\{\mathbf{u}_{t_i}^{(1)}, \ldots, \mathbf{u}_{t_i}^{(k)}\right\}$ in the following sense.

Lemma 2.5.4. *If* $\left\{\mathbf{u}_{t_i}\right\}$ *is a convergent subsequence and* $\lim_{i\to\infty}\mathbf{u}_{t_i} = \overline{\mathbf{u}}$, *then* $\lim_{i\to\infty}\mathbf{u}_{t_i}^{(\ell)} = \overline{\mathbf{u}}$ *for all* $\ell \in [\![k]\!]$.

Proof. Let $\left\{\sigma^{[t_i]}\right\}$ denote the corresponding sequence of permutations used in the algorithm for generating $\left\{\mathbf{u}_{t_i}\right\}$. Note that each time at least two vectors in the set $\left\{\mathbf{u}_{t_i}^{(1)}, \ldots, \mathbf{u}_{t_i}^{(k)}\right\}$ are identical with $\mathbf{u}_{t_i}$. Specifically, by Line 12 in the algorithm, we have $\mathbf{u}_{t_i}^{\left(\sigma_{k-1}^{[t_i]}\right)} = \mathbf{u}_{t_i}^{\left(\sigma_{k}^{[t_i]}\right)} = \mathbf{u}_{t_i}$. As $\mathbf{u}_{t_i}$ gets close to $\overline{\mathbf{u}}$, so do these two vectors. However, since $\left\{\sigma^{[t_i]}\right\}$ is of uniform distribution varying through all possible permutations as i goes to infinity, the locations of the so called "these two vectors" must also pervade through all possible pairs in the set $[\![k]\!]$. That is, all vectors in $\left\{\mathbf{u}_{t_i}^{(1)}, \ldots, \mathbf{u}_{t_i}^{(k)}\right\}$ are close to $\overline{\mathbf{u}}$ when i is large enough. □

Lemma 2.5.5. *For almost all symmetric tensors T, the accumulation points of the sequence $\{\mathbf{u}_t\}$ generated by Algorithm 3 are geometrically isolated.*

Proof. Suppose that $\overline{\mathbf{u}}$ is an accumulation point. By Corollary 2.2.1, we may perform the following operation, and by Lemma 2.5.4, $\overline{\mathbf{u}}$ is a solution to the nonlinear equation

$$T \circledast \mathbf{u}^{k-2} = \left\langle T, \mathbf{u}^k \right\rangle \mathbf{u}^2. \tag{2.15}$$

The equation (2.15) is a polynomial system of degree $k+2$ in the unknown $\mathbf{u} \in \mathbb{R}^n$ with leading coefficient T. By the theory of parameter continuation [164, Theorem 7.1.1], we know that for almost all symmetric tensor T, except for an affine algebraic subset of codimension one, the solutions to (2.15) are isolated. □

We stress that the polynomial system (2.15) might have multiple solutions. We are interested in the real solution that maximizes the generalized Rayleigh quotient $\left\langle T, \mathbf{u}^k \right\rangle$. The monotone behavior of $\{\lambda_t\}$ or $\{\mu_t\}$ seems to suggest that this is happening. However, we can conclude only that a local maximum is being realized by the iteration.

Theorem 2.5.1. *For almost all symmetric tensors T, the sequence $\{\mathbf{u}_t\}$ of dominant singular vectors generated in Algorithm 3 converges.*

Proof. Suppose that $\{\mathbf{u}_{t_i}\}$ is any subsequence converging to $\bar{\mathbf{u}}$. Suppose also by Lemma 2.5.5 that $\bar{\mathbf{u}}$ is isolated. By Lemma 2.5.4, the corresponding subsequences $\left\{\mathbf{u}_{t_i}^{(\ell)}\right\}$ converge to $\bar{\mathbf{u}}$ for all $\ell \in [\![k]\!]$. By construction (Line 5 in Algorithm 3), the subsequence $\{C_{t_i+1}\}$ of matrices converges. By continuity, the subsequence $\{\mathbf{u}_{t_i+1}\}$ must also converge to $\bar{\mathbf{u}}$. In particular, $\|\mathbf{u}_{t_i+1} - \mathbf{u}_{t_i}\| \to 0$. The condition in Lemma 2.2.6 therefore is satisfied. It follows that the whole sequence $\{\mathbf{u}_t\}$ converges to $\bar{\mathbf{u}}$. □

We conclude this part with a simple illustration. Given a symmetric tensor T of order 3 and an arbitrary unit vector $\mathbf{u}_0$, Algorithm 3 can be cast as fixed-point iteration defined by

$$\mathbf{u}_{t+1} = F(\mathbf{u}_t), \tag{2.16}$$

where $F(\mathbf{u})$ represents the dominant singular vector of $T \circledast \mathbf{u}$ (with consistent sign at the first entry). Then, by Theorem 2.5.1, the sequence $\{\mathbf{u}_t\}$ converges.

2.5.3 Convergence of iterates in Algorithm 1

Algorithm 1 is somewhat too conservative for computation in practice. However, its fundamental structure is the basis of Algorithm 3. In this subsection, we prove its convergence.

The difficulty of Algorithm 1 is at the complexity that $\mathbf{u}_{[p+1]}^{(\ell)}$ depends on both the new vectors $\left\{\mathbf{u}_{[p+1]}^{(i)}\right\}$, $i = 1, \ldots, \ell - 1$, and the old vectors $\left\{\mathbf{u}_{[p]}^{(i)}\right\}$, $i = \ell + 2, \ldots, k$. We first make the following observation about the collective behavior.

Lemma 2.5.6. *There is a subsequence $\left\{\mathbf{u}_{[p_j]}^{(\ell)}\right\}$ generated by Algorithm 1 that converges to the same limit point for all $\ell = [\![k]\!]$.*

Proof. For each fixed ℓ, it is always true that $\|C_{[p]}^{(\ell)}\|_F \leq \|T\|_F$ for all p because $\| \bigotimes_{i=1}^{\ell-1} \mathbf{u}_{[p+1]}^{(i)} \otimes \bigotimes_{i=\ell+2}^{k} \mathbf{u}_{[p]}^{(i)} \|_F = 1$. The Bolzano-Weierstrass theorem guarantees a convergent subsequence. There are only finitely many ℓ. Selecting a subsequence of a subsequence, if necessary, we can find a common subset $\{p_j\}$ of nonnegative integers so that $\left\{C^{(\ell)}_{[p_j]}\right\}$ converges simultaneously for all $\ell \in [\![k]\!]$. We claim that the assertion holds for the subsequence $\left\{\mathbf{u}^{(\ell)}_{[p_j]}\right\}$.

For clarity, we accomplish the proof in two steps. First, we argue that the sequences $\left\{\mathbf{u}^{(\ell)}_{[p_j]}\right\}$ converge simultaneously for all $\ell \in [\![k]\!]$. Second, we argue that they converge to the same limit point.

By continuity, the sequences of the corresponding dominant singular vectors $\left\{\widehat{\mathbf{u}}^{(1)}_{[p_j+1]}\right\}$ and $\left\{\mathbf{u}^{(\ell)}_{[p_j+1]}\right\}$ of $\left\{C^{(\ell)}_{[p_j]}\right\}$ converge simultaneously for all $\ell \in [\![k]\!]$. Denote $\lim_{j\to\infty} C^{(\ell)}_{[p_j]} = C^{(\ell)}$, $\lim_{j\to\infty} \widehat{\mathbf{u}}^{(1)}_{[p_j+1]} = \mathbf{u}^{(1)}_{\sharp}$ and $\lim_{j\to\infty} \mathbf{u}^{(\ell)}_{[p_j+1]} = \mathbf{u}^{(\ell)}_{\sharp}$ for $\ell = 2, \dots, k$. The subscript ${}_{\sharp}$ is a handy way to remind us that these are the limit points corresponding to the subsequence ${}_{[p_j+1]}$. We shall assume the generic condition that $C^{(\ell)}$ is nonsingular for all ℓ.

To prove the simultaneous convergence, we consider separate cases:

Case 1. For $\ell = 4, \dots k$, let $\eta = \ell - 2$ so that $2 \leq \eta \leq k-2$. By using Lemma 2.2.2, we obtain the equalities

$$\begin{aligned} C^{(\eta)}_{[p_j]}\mathbf{u}^{(\eta)}_{[p_j+1]} &= \left(T \circledast_{\boldsymbol{\beta}_\eta} \left(\widehat{\mathbf{u}}^{(1)}_{[p_j+1]} \otimes \dots \otimes \mathbf{u}^{(\eta-1)}_{[p_j+1]} \otimes \mathbf{u}^{(\eta+2)}_{[p_j]} \otimes \mathbf{u}^{(\eta+3)}_{[p_j]} \otimes \dots \otimes \mathbf{u}^{(k)}_{[p_j]}\right)\right) \mathbf{u}^{(\eta)}_{[p_j+1]} \\ &= \left(T \circledast_{\boldsymbol{\beta}_{\eta+1}} \left(\widehat{\mathbf{u}}^{(1)}_{[p_j+1]} \otimes \dots \otimes \mathbf{u}^{(\eta-1)}_{[p_j+1]} \otimes \mathbf{u}^{(\eta)}_{[p_j+1]} \otimes \mathbf{u}^{(\eta+3)}_{[p_j]} \otimes \dots \otimes \mathbf{u}^{(k)}_{[p_j]}\right)\right) \mathbf{u}^{(\eta+2)}_{[p_j]} \\ &= C^{(\eta+1)}_{[p_j]}\mathbf{u}^{(\eta+2)}_{[p_j]}. \end{aligned} \tag{2.17}$$

Taking the limits on both sides of (2.17), together with the non-singularity of $C^{(\eta+1)}$, we see that $\lim_{j\to\infty} \mathbf{u}^{(\ell)}_{[p_j]}$ exists for $\ell = 4, \dots, k$. The algorithm entails that $\mathbf{u}^{(1)}_{[p_j]} = \mathbf{u}^{(k)}_{[p_j]}$, so the convergence of $\mathbf{u}^{(1)}_{[p_j]}$ is a by-product.

Case 2. For $\ell = 3$, consider the identities

$$\begin{aligned} C^{(1)}_{[p_j]}\widehat{\mathbf{u}}^{(1)}_{[p_j+1]} &= \left(T \circledast_{\boldsymbol{\beta}_1} \left(\mathbf{u}^{(3)}_{[p_j]} \otimes \mathbf{u}^{(4)}_{[p_j]} \otimes \dots \otimes \mathbf{u}^{(k)}_{[p_j]}\right)\right) \widehat{\mathbf{u}}^{(1)}_{[p_j+1]} \\ &= \left(T \circledast_{\boldsymbol{\beta}_2} \left(\widehat{\mathbf{u}}^{(1)}_{[p_j+1]} \otimes \mathbf{u}^{(4)}_{[p_j]} \otimes \dots \otimes \mathbf{u}^{(k)}_{[p_j]}\right)\right) \mathbf{u}^{(3)}_{[p_j]} = C^{(2)}_{[p_j]}\mathbf{u}^{(3)}_{[p_j]}. \end{aligned} \tag{2.18}$$

Taking the limits at both ends of (2.18), together with $C^{(2)}$ being nonsingular, we see that $\lim_{j\to\infty} \mathbf{u}^{(3)}_{[p_j]}$ exists.

Case 3. For $\ell = 2$, observe the relationship

$$C^{(k)}_{[p_j-1]}\mathbf{u}^{(k)}_{[p_j]} = \pm\lambda^{(k)}_{[p_j]}\mathbf{u}^{(k)}_{[p_j]}$$

$$\begin{aligned} &= \left(T \circledast_{\boldsymbol{\beta}_k} \left(\mathbf{u}_{[p_j]}^{(2)} \otimes \ldots \otimes \mathbf{u}_{[p_j]}^{(k-1)}\right)\right) \mathbf{u}_{[p_j]}^{(k)} \\ &= \left(T \circledast_{\boldsymbol{\beta}_1} \left(\mathbf{u}_{[p_j]}^{(3)} \otimes \ldots \otimes \mathbf{u}_{[p_j]}^{(k)}\right)\right) \mathbf{u}_{[p_j]}^{(2)} = C_{[p_j]}^{(1)} \mathbf{u}_{[p_j]}^{(2)}. \end{aligned} \tag{2.19}$$

Though we do not know the convergence of $C_{[p_j-1]}^{(k)}$, the convergence of $\lambda_{[p_j]}^{(k)}$, $\mathbf{u}_{[p_j]}^{(k)}$ and $C_{[p_j]}^{(1)}$ implies that $\lim_{j\to\infty} \mathbf{u}_{[p_j]}^{(2)}$ exists, which in turn implies that $C_{[p_j-1]}^{(k)}$ converges.

Now we prove that sequences converge to the same limit point. Denote $\lim_{j\to\infty} \mathbf{u}_{[p_j]}^{(\ell)} = u_{\natural}^{(\ell)}$ for $\ell = 1, \ldots, k$. It thus becomes clear that

$$C^{(\ell)} = T \circledast_{\boldsymbol{\beta}_\ell} \left(\mathbf{u}_{\sharp}^{(1)} \otimes \ldots \otimes \mathbf{u}_{\sharp}^{(\ell-1)} \otimes \mathbf{u}_{\natural}^{(\ell+2)} \otimes \mathbf{u}_{\natural}^{(\ell+3)} \otimes \ldots \otimes \mathbf{u}_{\natural}^{(k)}\right). \tag{2.20}$$

Analogous to (2.17), we mention also the identity

$$\begin{aligned} C_{[p_j]}^{(k-1)} \mathbf{u}_{[p_j+1]}^{(k-1)} &= \pm\lambda_{[p_j+1]}^{(k-1)} \mathbf{u}_{[p_j+1]}^{(k-1)} \\ &= \left(T \circledast_{\boldsymbol{\beta}_{k-1}} \left(\widehat{\mathbf{u}}_{[p_j+1]}^{(1)} \otimes \mathbf{u}_{[p_j+1]}^{(2)} \otimes \ldots \otimes \mathbf{u}_{[p_j+1]}^{(k-2)}\right)\right) \mathbf{u}_{[p_j+1]}^{(k-1)} \\ &= \left(T \circledast_{\boldsymbol{\beta}_k} \left(\mathbf{u}_{[p_j+1]}^{(2)} \otimes \ldots \otimes \mathbf{u}_{[p_j+1]}^{(k-2)} \otimes \mathbf{u}_{[p_j+1]}^{(k-1)}\right)\right) \widehat{\mathbf{u}}_{[p_j+1]}^{(1)} = C_{[p_j]}^{(k)} \widehat{\mathbf{u}}_{[p_j+1]}^{(1)}. \end{aligned}$$

It follows, by construction and continuity, that we have the relationships

$$\pm\widetilde{\lambda}\mathbf{u}_{\natural}^{(1)} = \pm\widetilde{\lambda}\mathbf{u}_{\natural}^{(k)} = C^{(1)}\mathbf{u}_{\natural}^{(2)}, \tag{2.21}$$

$$C^{(\ell)}\mathbf{u}_{\sharp}^{(\ell)} = \pm\widetilde{\lambda}\mathbf{u}_{\sharp}^{(\ell)} = C^{(\ell+1)}\mathbf{u}_{\natural}^{(\ell+2)}, \quad \ell = 1, \ldots, k-2, \tag{2.22}$$

$$C^{(k-1)}\mathbf{u}_{\sharp}^{(k-1)} = \pm\widetilde{\lambda}\mathbf{u}_{\sharp}^{(k-1)} = C^{(k)}\mathbf{u}_{\sharp}^{(1)}, \tag{2.23}$$

$$C^{(k)}\mathbf{u}_{\sharp}^{(k)} = \pm\widetilde{\lambda}\mathbf{u}_{\sharp}^{(k)}. \tag{2.24}$$

By continuity again, note that $\widetilde{\lambda}$ is the dominant singular value of all matrices $C^{(\ell)}$, $\ell = 1, \ldots, k$. Under the assumption that $\widetilde{\lambda}$ is simple, the corresponding singular vector is unique up to a sign change. However, because in Lines 6 to 8 of Algorithm 1 we have required that the first entry of the dominant singular vector be positive, such a sign change does not exist. Recursively, we obtain the relationships

$$\begin{cases} \mathbf{u}_{\sharp}^{(1)} = \mathbf{u}_{\natural}^{(1)} = \mathbf{u}_{\natural}^{(k)} = \mathbf{u}_{\natural}^{(2)}, & \\ \mathbf{u}_{\sharp}^{(\ell)} = \mathbf{u}_{\natural}^{(\ell+2)}, & \ell = 1, \ldots, k-2, \\ \mathbf{u}_{\sharp}^{(1)} = \mathbf{u}_{\sharp}^{(k-1)} = \mathbf{u}_{\sharp}^{(k)}. & \end{cases} \tag{2.25}$$

These relationships allow us to write

$$\begin{aligned}\widetilde{\lambda} &= \left|\left\langle C^{(1)}, \mathbf{u}_{\sharp}^{(1)} \otimes \mathbf{u}_{\sharp}^{(1)} \right\rangle\right| = \left|\left\langle T \circledast_{\boldsymbol{\beta}_1} \left(\mathbf{u}_{\natural}^{(3)} \otimes \mathbf{u}_{\natural}^{(4)} \otimes \ldots \otimes \mathbf{u}_{\natural}^{(k)}\right), \mathbf{u}_{\natural}^{(1)} \otimes \mathbf{u}_{\natural}^{(2)} \right\rangle\right| \\ &= \left|\left\langle T \circledast_{\boldsymbol{\beta}_2} \left(\mathbf{u}_{\natural}^{(1)} \otimes \mathbf{u}_{\natural}^{(4)} \otimes \ldots \otimes \mathbf{u}_{\natural}^{(k)}\right), \mathbf{u}_{\natural}^{(2)} \otimes \mathbf{u}_{\natural}^{(3)} \right\rangle\right| \\ &= \left|\left\langle T \circledast_{\boldsymbol{\beta}_2} \left(\mathbf{u}_{\sharp}^{(1)} \otimes \mathbf{u}_{\natural}^{(4)} \otimes \ldots \otimes \mathbf{u}_{\natural}^{(k)}\right), \mathbf{u}_{\natural}^{(2)} \otimes \mathbf{u}_{\natural}^{(3)} \right\rangle\right|.\end{aligned}$$

But we also have

$$\widetilde{\lambda} = \left|\left\langle C^{(2)}, \mathbf{u}_{\sharp}^{(2)} \otimes \mathbf{u}_{\sharp}^{(2)} \right\rangle\right| = \left|\left\langle T \circledast_{\boldsymbol{\beta}_2} \left(\mathbf{u}_{\sharp}^{(1)} \otimes \mathbf{u}_{\natural}^{(4)} \otimes \ldots \otimes \mathbf{u}_{\natural}^{(k)}\right), \mathbf{u}_{\sharp}^{(2)} \otimes \mathbf{u}_{\sharp}^{(2)} \right\rangle\right|.$$

By the uniqueness of the dominant singular vector, since the first entry is kept positive, we conclude that

$$\mathbf{u}_{\sharp}^{(2)} = \mathbf{u}_{\natural}^{(2)}. \tag{2.26}$$

Repeating this process, we can prove that $\mathbf{u}_{\sharp}^{(\ell)} = \mathbf{u}_{\natural}^{(\ell)}$ for $\ell = 2, \ldots, k-1$. Together with (2.25), we finally prove that the sequence $\left\{\mathbf{u}_{[p_j]}^{(\ell)}\right\}$ converges to the same limit point for all $\ell = 1, \ldots k$. □

Corollary 2.5.1. *If the sequences $\left\{C_{[p_j]}^{(\ell)}\right\}$ of matrices converge simultaneously for all $\ell \in [\![k]\!]$, then they converge to the same limit point for all $\ell \in [\![k]\!]$.*

Proof. The assertion follows from (2.20) and the fact that $\mathbf{u}_{\sharp}^{(\ell)} = \mathbf{u}_{\natural}^{(\ell)}$ for $\ell \in [\![k]\!]$ which has been proved in the previous lemma. □

In the proof of Lemma 2.5.6, we make use of simultaneously convergent subsequences $\left\{C_{[p_j]}^{(\ell)}\right\}$ to argue the simultaneously convergent subsequences $\left\{\mathbf{u}_{[p_j]}^{(\ell)}\right\}$. We can also reverse the argument.

Corollary 2.5.2. *If subsequences $\left\{\mathbf{u}_{[p_j]}^{(\ell)}\right\}$ converge simultaneously for all $\ell \in [\![k]\!]$, then so do subsequences $\left\{C_{[p_j]}^{(\ell)}\right\}$ and $\left\{\mathbf{u}_{[p_j+1]}^{(\ell)}\right\}$.*

Proof. The simultaneous convergence of $\left\{\mathbf{u}_{[p_j]}^{(\ell)}\right\}$ for $\ell = 3, \ldots k$ implies that the subsequence $\left\{C_{[p_j]}^{(1)}\right\}$ converges. By continuity, $\left\{\widehat{\mathbf{u}}_{[p_j+1]}^{(1)}\right\}$ converges. But then by definition, $\left\{C^{(2)}_{[p_j]}\right\}$ converges and, thus, so does $\left\{\mathbf{u}^{(2)}_{[p_j+1]}\right\}$. Cycling through the ℓ-loop in Algorithm 1, the assertion is proved. □

Theorem 2.5.2. *For almost all symmetric tensors T, the sequences $\{\mathbf{u}^{(\ell)}_{[p]}\}$ generated in Algorithm 1 converge to the same limit point.*

Proof. Let $\left\{\mathbf{u}^{(\ell)}_{[p_j]}\right\}$ be any simultaneously convergent subsequences. By Corollary 2.5.2, the corresponding subsequences $\left\{C^{(1)}_{[p_j]}\right\}$ converge simultaneously. Using the same argument in the proof of Lemma 2.5.6, we see that subsequences $\left\{\mathbf{u}^{(\ell)}_{[p_j]}\right\}$ and $\left\{\mathbf{u}^{(\ell)}_{[p_j+1]}\right\}$ converge to the same limit point for all $\ell \in [\![k]\!]$. The limit point must satisfy the polynomial system (2.15), whence we assume is geometrically isolated. By Lemma 2.2.6, we obtain the convergence. □

2.6 Numerical examples

Note that all the experiments in this thesis are performed on a MacBook with 2.3 GHz Intel Core i7 processor and 16 GB 1600 MHz DDR3 memory running MATLAB with version R2015a (8.5.0.19613).

In this section, we carry out some numerical experiments to demonstrate the working of our algorithms. We concentrate mostly on Algorithm 3. Because the size of data grows rapidly as n^k, we will not list the test data. At present, we pay no attention to fine tune the implementation for efficiency either. We simply describe how we set up our experiments and present some empirical observations.

Example 1. Our first concern is the complexity analysis of the algorithm. Given the abilities of high performance (vector or parallel) processors today, simple floating-point operations (flops) counts are not at all valid any more. On the other hand, on a dedicated machine, the CPU time should be approximately proportional to the number of flops, albeit the I/O will also cost time. We decide to measure the CPU time required in each of the major components in the algorithm. We divide the measurement as follows:

- T_{Outer} = the time needed to form the outer product $\bigotimes_{i=1}^{k-2} \mathbf{u}^{(i)}$.

- T_C = the time needed to perform the tensor multiplication $\circledast$ for creating C_t in Line 5.

- T_{SVD} = the time needed for the SVD at Line 6.

- T_{Total} = the total execution time, including every other possible details such as I/O.

For the case $k = 3$, we vary the dimension $n = 2^p$ for $p = 4, \ldots, 9$. For the case $k = 4$, an order-4 tensor of dimension $n = 2^8$ requires 32GB bytes. So we limit ourselves to $p = 4, \ldots, 7$ only. Each case of p is repeatedly tested 20 times with random starting unit vectors and we plot the average as the running time in Figure 2.1.

It should not be surprising that the overhead T_{Outer} remains almost constant for the case $k = 3$ because no outer product is needed except for swapping columns at Line 12. However, we find that neither T_{Outer} does vary significantly even for the case $k = 4$. What is interesting is that for small size problems, say, $n \leq 32$, the overhead T_{Total} is attributed mainly to T_{SVD}. But when n is sufficiently large, while the SVD should cost more time, the cost T_C of the tensor product $\circledast$ outweighs T_{SVD} of the SVD. It is seen in Figure 2.1 that when $n = 2^9$ and $k = 3$ or when $n = 2^7$ when $k = 4$, the main contribution to T_{Total} is from T_C.

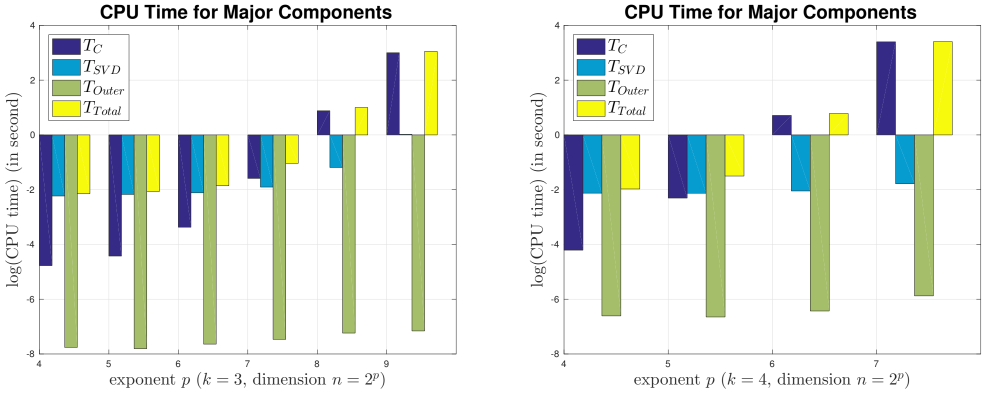

Figure 2.1: Breakdown of CPU time needed for calculation of major parts in Algorithm 3.

Example 2. We are curious about the performance of the SVD-based algorithms when comparing among themselves as well some of the popular methods. For this purpose, we apply the same stopping criteria to all methods — the iteration terminates when three consecutive generalized Rayleigh quotients do not vary more than the tolerance 10^{-8}. We measure the CPU time needed by our Algorithms 1-4, as well as the conventional ALS and symmetric ALS [109]. The problem sizes are chosen in the same way as in Example 1. We execute each algorithm by 20 runs with random initial unit vectors. They may converge to different limit points, but we give every algorithm the same criteria to reach convergence. We compute the average time. In all tests, we find that Algorithm 3 is fastest especially for large p in Figure 2.2. Algorithm ALS and Algorithm 2 perform better when p is small. Compared to randomise methods Algorithm 2 and Algorithm 3, Algorithm 1 is less effective for both small and large p.

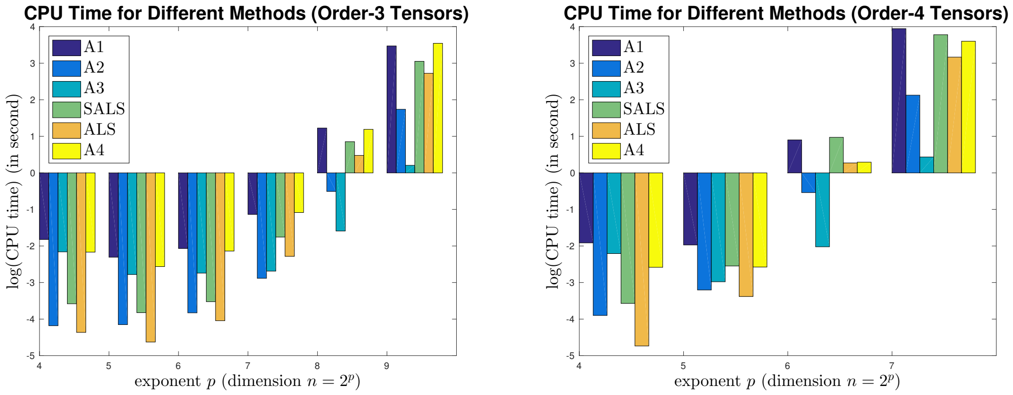

Figure 2.2: Breakdown of CPU time for comparison among different methods.

In Example 1, we find that the $\boldsymbol{\beta}$-product $T \circledast_{\boldsymbol{\beta}} S$ is the most expensive part of the calculation for our Algorithm 3. Since the conventional ALS method involves a similar calculation, we make a rough complexity comparison between Algorithm 3 and the conventional ALS method. In the ALS method, forming $\bigotimes_{i=1}^{k-1} \mathbf{u}^{(i)}$ requires n^{k-1} entry-to-entry multiplications and there are k layers to form the inner product $T \circledast \bigotimes_{i=1}^{k-1} \mathbf{u}^{(i)}$. Together with the required normalization, the ALS method requires $n^k + n^{k-1} + n$ scalar multiplications per update. In contrast, the SVD-based method requires $n^k + n^{k-2}$ scalar multiplications per $\boldsymbol{\beta}$-product if $k > 3$, and n^3 if $k = 3$. Additionally, the svds involved in the SVD-based method requires more overhead than the normalization involved in the ALS method. It is difficult to estimate how many iterations are required inside the svds to generate the dominant singular value triplet, but the total cost should not be worse than $O(n^3)$. We thus estimate that the SVD-based method requires $O(n^k+n^{k-2}+n^3)$ scalar multiplications per update. These estimates seem compatible per update, however, the numerical evidence in Example 2 clearly demonstrates that overall Algorithm 3 is significantly faster than the ALS method.

Though they are not in the same category, it might be interesting to compare the above-mentioned complexities of the iterative methods in one update to those of finite algorithms. The so called SeROAP method proposed in [55] requires $O(\frac{2p(n^{k+1}-n^2)}{n-1})$ scalar multiplications (where p is a user-defined parameter) for the decreasing order phase and $O(\frac{2(n^{k+1}-n^2)}{n-1})$ scalar multiplications for the projection phase. Similarly, the ST-HOSVD and T-HOSVD methods proposed in [182] require $O(\frac{n^{k+2}+n^k-n^2-1}{n-1})$ and $O(\frac{kn^{k+2}+(1-k)n^{k+1}-n}{n-1})$ operations, respectively. These finite algorithms give rise to some good approximations, but have no mechanism for further improvement. So, they might serve as a good starting point for further iteration such as by our SVD-based methods and the ALS method. We have not explored this hybrid approach in this study.

It is worthy of a further remark on whether or how a scheme preserves symmetry. At first glance, it might seem that in our algorithms A1, A2, and A3, we have imposed symmetry because we update two vectors simultaneously by the same dominant singular vector. The fact is that because only two vectors are updated a time while others are not affected, the symmetry is not required or even expected. One of the most important points in our theory is that at the end all factors in iterates converge to the same vector and, hence, symmetry shows up. On the other hand, as is reported in Figure 2.2, imposing symmetry in the ALS scheme, which is the SALS method, or imposing total symmetry in the SVD-based scheme, which is the A4 algorithm, has no advantage over the conventional ALS or the randomized A3 algorithm.

Finally, we need to point out that, although the symmetric limit point (by non-symmetric algorithms) for symmetric tensors is expected in theory, we have to set some stopping criteria for the iteration in practice. Consequently, the numerical results returned by non-symmetric algorithms might not be perfectly symmetric. In contrast, the symmetric ALS method in [109] and our totally symmetric SVD-based method (A4) always keep the symmetry in each step. They are slower, but have the advantage of keeping the symmetry in bay when comparing to non-symmetric algorithms.

Example 3. Even though we have taken the advantage of the SVD that produces the best approximation per iteration, its effect is limited to the locality. We generate 15 random vectors $\mathbf{x}_i \in \mathbb{R}^{10}$ and combine them into $T = \sum_{i=1}^{15} \mathbf{x}_i^7$. As these vectors $\{\mathbf{x}_i\}$ are linearly dependent, we no longer have a good way to estimate the rank of T [50]. Still, its best rank-1 approximation is guaranteed to exist. Applying Algorithm 3 with 20 distinct sets of random starting unit vectors, we plot history of iterations for each of the 20 tests. We continue to observe properties such as convergence and monotonicity discussed earlier in this chapter. However, numerical results in Figure 2.3 indicate the possibility of having multiple local solutions. See Lemma 2.5.5. The number of locally best rank-1 approximations should be the same of real solutions to the polynomial system (2.15), but that depends on T. While the SVD-based algorithms seem capable of capturing the solution with "larger" objective values in most of the trials, this experiment demonstrates that the notion of the best rank-1 approximation should be interpreted only locally.

Example 4. To experiment the sensitivity of a rank-1 tensor subject to perturbations, we randomly generate six vectors $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_5 \in \mathbb{R}^{10}$ from the identical and independent standard normal distribution. Define $T_0 = \mathbf{x}_0^7$. This rank-1, order-7, dimension-10, and symmetric tensor T_0 will be fixed as our target. Define the unit tensor $B := \frac{\sum_{i=1}^{5} \mathbf{x}_i^7}{\| \sum_{i=1}^{5} \mathbf{x}_i^7 \|_F}$ which generically is of rank 5 [50]. We perturb T_0 via an

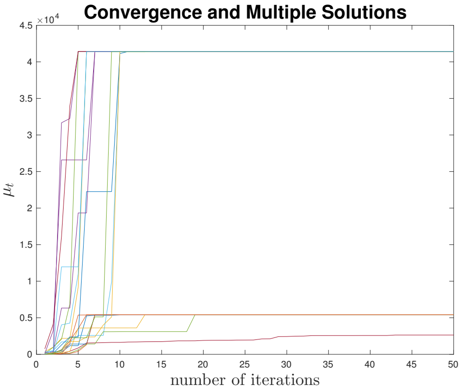

Figure 2.3: History of monotone convergence and existence of multiple best rank-1 approximations.

additive noise of the form

$$T_\sigma = T_0 + \sigma B$$

where $\sigma \in [0, 2]$ signifies the magnitude of the noise. By gradually increasing the strength of perturbation, we compute the rank-1 approximation tensor $\overline{T}_\sigma$ of T_σ by Algorithm 3. The noise level $\|T_\sigma - T_0\|_F = \sigma$ is low relative to $\|T_0\|_F$, but the added noise certainly disrupts the rank. Our goal is to compare the difference between $\overline{T}_\sigma$ and the original T_0 as well as the computed generalized Rayleigh quotient $\mu(\overline{T}_\sigma)$. Plotted in Figure 2.4 are the relative differences, showing that the computed rank-1 tensor $\overline{T}_\sigma$ is a reasonable approximation to T_0, but the discretion is large enough to suggest that it is not recovering T_0 exactly. It is interesting to note that, despite of the high nonlinearity involved, the quantities $|\mu(\overline{T}_\sigma) - \mu(T_0)|$ and $\|\overline{T}_\sigma - T_0\|_F$ are almost linearly in σ.

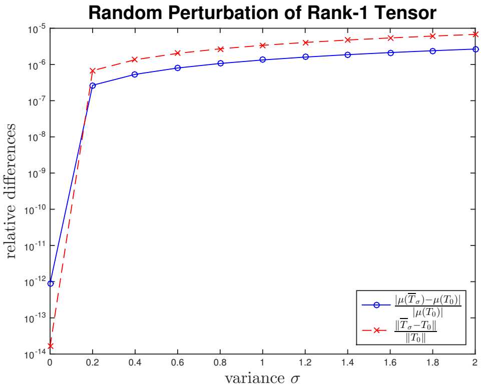

Figure 2.4: Relative difference $\frac{|\mu(\overline{T}_\sigma)-\mu(T_0)|_F}{|\mu(T_0)|_F}$ *and* $\frac{\|\overline{T}_\sigma-T_0\|}{\|T_0\|}$ *between the computed rank-1 tensor* $\overline{T}_\sigma$ *and the original* T_0.

Chapter 3

Non-Symmetric Best Rank-1 Approximation

3.1 Introduction

Finding the best rank-1 approximation to a generic tensor is similar to the symmetric case in Chapter 2 which is to determine unit vectors $\mathbf{u}^{(\ell)} \in \mathbb{R}^{I_\ell}$, $\ell = 1, \dots k$, and a scalar $\lambda \in \mathbb{R}$ such that

$$\left\| T - \lambda \mathbf{u}^{(1)} \otimes \ldots \otimes \mathbf{u}^{(k)} \right\|_F^2 \tag{3.1}$$

is minimized for a given generic tensor $T \in \mathbb{R}^{I_1 \times \ldots \times I_k}$. For any fixed unit vectors $\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(k)}$, the optimal value of λ for (3.1) is given precisely by the length of the projection of the "vector" T onto the direction of the "unit vector" $\bigotimes_{\ell=1}^{k} \mathbf{u}^{(\ell)} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_k}$, i.e.,

$$\lambda = \lambda(\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(k)}) = \langle T, \bigotimes_{\ell=1}^{k} \mathbf{u}^{(\ell)} \rangle. \tag{3.2}$$

Thus, minimizing the orthogonal component of T, as is desired in (3.1), is equivalent to maximizing the length $|\lambda|$ of the parallel component. In [197], the expression (3.2) is called the generalized Rayleigh quotient of T relative to $\{\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(k)}\}$. Switching the signs of the variables $\mathbf{u}^{(\ell)}$ if necessary, we may restrict our attention without loss of generality to the case $\lambda > 0$ only.

3.1.1 Summary

This chapter is not concerned about how fast the different algorithms perform, nor what quality they achieve. Rather, we are curious about the more fundamental question of whether the iteration converges at all. Recall that the convergence theory for the ALS method was established much later than the method had been put into practice [52,179,189]. A similar concern is raised for the SVD-based algorithm — the convergence of the generalized Rayleigh quotients is obvious, but the convergence analysis for the iterates themselves has been elusive in the literature [75, Page 947]. In this chapter, we provide a rigorous mathematical proof for the convergence of iterates from a specific SVD-based algorithm, which thus complements the theory. We learn recently that an independent work in the report [193] also investigates the convergence theory by using the Łojasiewicz gradient inequality [44, 132, 133]. Indeed, we have employed a similar technique in proving the global convergence of the ALS method in [189]. The tactics we develop in this chapter for the SVD-based algorithm are using an entirely different approach. Our approach relies only on the continuity of singular vectors and real analysis, which, in our opinion, is much more straightforward.

3.1.2 Outline of the chapter

This chapter is organized as follows. We begin with a brief review of the basic operation in Section 3.2 to prepare for the discussion. We describe two variants of SVD-based algorithms in Section 3.3. The difference is at where the SVD is to be applied. Our main result is presented in Section 3.4 where we explain the meaning of a tensor being generic and argue the convergence for the most basic algorithm. Finally, though it is not the main objective of this chapter, we carry out some exploratory experiments in Section 3.5 to compare performance between proposed and other types of SVD-based algorithms.

3.2 Basics

We briefly review some notations in section 2.2 for later useage in this chapter. Given a fixed partitioning $[\![k]\!] = \boldsymbol{\alpha} \cup \boldsymbol{\beta}$, we shall regard an order-k tensor $T \in \mathbb{R}^{I_1\times\ldots\times I_k}$ as a "matrix representation" of a linear operator mapping order-s tensors to order-t tensors [189]. Specifically, we identify T with the linear map

$$\mathscr{T}_{\boldsymbol{\beta}} : \mathbb{R}^{I_{\alpha_1}\times\ldots\times I_{\alpha_s}} \to \mathbb{R}^{I_{\beta_1}\times\ldots\times I_{\beta_t}}, \tag{3.3}$$

such that for any $S \in \mathbb{R}^{I_{\alpha_1}\times\ldots\times I_{\alpha_s}}$, we have

$$\mathscr{T}_{\boldsymbol{\beta}}(S) := T \circledast_{\boldsymbol{\beta}} S = [\langle \tau_{[:|\ell_1,\ldots,\ell_t]}, S\rangle] \in \mathbb{R}^{I_{\beta_1}\times\ldots\times I_{\beta_t}} \tag{3.4}$$

where

$$\langle \tau_{[:|\ell_1,\ldots,\ell_t]}, S\rangle := \sum_{i_1=1}^{I_{\alpha_1}} \ldots \sum_{i_s=1}^{I_{\alpha_s}} \tau_{[i_1,\ldots,i_s|\ell_1,\ldots,\ell_t]} s_{i_1,\ldots,i_s} \tag{3.5}$$

is the Frobenius inner product generalized to multi-dimensional arrays. The $\boldsymbol{\beta}$-product defined by (3.4) is a natural generalization of the usual matrix-vector multiplication in the sense that if an order-2 tensor $T \in \mathbb{R}^{m\times n}$ is regarded as a matrix, and if the column is identified by the pointer $\boldsymbol{\alpha} = \{1\}$ and the row by $\boldsymbol{\beta} = \{2\}$ so that $\tau_{ij} = \tau^{(\{2\},\{1\})}_{[i|j]}$, then with respect to given column vectors $\mathbf{z} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^m$ we can write

$$\left\{\begin{array}{rcl} T\mathbf{z} & = & T \circledast_2 \mathbf{z}, \\ T^\top \mathbf{y} & = & T \circledast_1 \mathbf{y}. \end{array}\right. \tag{3.6}$$

This notation is handy in our convergence analysis later.

Mode products. For any matrix $M = [m_{tj}] \in \mathbb{R}^{\gamma\times I_d}$, the mode-$d$ matrix product [1] is defined to be [14, 114]

$$P = T \times_d M = \left[\left\langle \tau^{(\{d\},\{d\}^C)}_{[:|\mathcal{J}]}, m^{(1,2)}_{[t|:]} \right\rangle\right] \in \mathbb{R}^{I_1\times\ldots\times I_{d-1}\times\gamma\times I_{d+1}\times\ldots\times I_k}, \tag{3.7}$$

that is,

$$p^{(\{d\},\{d\}^C)}_{[t|\mathcal{J}]} = p_{j_1,\ldots,j_{d-1},t,j_{d+1},\ldots j_k} := \sum_{\ell=1}^{I_d} m_{t,\ell} \tau_{j_1,\ldots,j_{d-1},\ell,j_{d+1},\ldots j_k}. \tag{3.8}$$

[1] Given a vector $v \in \mathbb{R}^{I_d}$, the so called mode-d vector product, denoted by $T\overline{\times}_d v$, is equal to $T \circledast_{\{d\}^C} v$ which is in $\mathbb{R}^{I_1\times I_{d-1}\times I_{d+1}\times\ldots\times I_k}$.

We should carefully differentiate $\boldsymbol{\beta}$-product $\circledast_{\boldsymbol{\beta}}$ from this d-mode product $\times_d$. First, notice that the d-mode product of a tensor with a matrix maintains the same order of the original tensor, but the $\boldsymbol{\beta}$-product reduces the order from $k = s+t$ to t. Second, notice that the d-mode product of a tensor with a vector is indeed a contraction. When S is of the form $S = \mathbf{u}^{(1)}\otimes\ldots\otimes\mathbf{u}^{(s)}$, then

$$(T\circledast_{\boldsymbol{\beta}} S)_{\ell_1,\ldots,\ell_t} = \sum_{i_1=1}^{I_{\alpha_1}}\cdots\sum_{i_s=1}^{I_{\alpha_s}} \tau_{[i_1,\ldots,i_s|\ell_1,\ldots,\ell_t]} u_{i_1}^{(1)}\ldots u_{i_s}^{(s)} = T\times_{\alpha_1}\mathbf{u}^{(1)}\times_{\alpha_2}\mathbf{u}^{(2)}\ldots\times_{\alpha_s}\mathbf{u}^{(s)}.$$

However, we are not aware of a consistent way to define the d-mode product when S is a general order-s tensor.

The following basic facts will be used in the subsequent discussion.

Lemma 3.2.1. *Given a general tensor $T \in \mathbb{R}^{I_1\times I_2\times\ldots\times I_k}$, a partitioning $[\![k]\!] = \boldsymbol{\alpha}\cup\boldsymbol{\beta}$, and vectors $\mathbf{u}^{(\ell)} \in \mathbb{R}^{I_\ell}$, $\ell = 1,\ldots,k$, then it holds that*

$$\langle T, \bigotimes_{\ell=1}^{k}\mathbf{u}^{(\ell)}\rangle = \langle T\circledast_{\boldsymbol{\beta}}\bigotimes_{i=1}^{s}\mathbf{u}^{(\alpha_i)}, \bigotimes_{j=1}^{t}\mathbf{u}^{(\beta_j)}\rangle. \tag{3.9}$$

Lemma 3.2.2. *Given a general tensor $T \in \mathbb{R}^{I_1\times I_2\times\ldots\times I_k}$, arbitrary vectors $\mathbf{u}^{(\alpha_i)} \in \mathbb{R}^{I_{\alpha_i}}$, $i \in [\![k-2]\!]$, $\mathbf{v} \in \mathbb{R}^{I_{\beta_2}}$, and $\mathbf{w} \in \mathbb{R}^{I_{\beta_1}}$, then*

$$(T\circledast_{\{\beta_1,\beta_2\}}\bigotimes_{i=1}^{k-2}\mathbf{u}^{(\alpha_i)})\circledast_{\beta_1}\mathbf{v} = (T\circledast_{\{\beta_1,\alpha_j\}}\bigotimes_{i=1}^{j-1}\mathbf{u}^{(\alpha_i)}\otimes\mathbf{v}\otimes\bigotimes_{i=j+1}^{k-2}\mathbf{u}^{(\alpha_i)})\circledast_{\beta_1}\mathbf{u}^{(\alpha_j)} \tag{3.10}$$

$$(T\circledast_{\{\beta_1,\beta_2\}}\bigotimes_{i=1}^{k-2}\mathbf{u}^{(\alpha_i)})\circledast_{\beta_2}\mathbf{w} = (T\circledast_{\{\alpha_j,\beta_2\}}\bigotimes_{i=1}^{j-1}\mathbf{u}^{(\alpha_i)}\otimes\mathbf{w}\otimes\bigotimes_{i=j+1}^{k-2}\mathbf{u}^{(\alpha_i)})\circledast_{\beta_2}\mathbf{u}^{(\alpha_j)} \tag{3.11}$$

for any $j \in [\![k-2]\!]$.

Then we restate Lemma 2.2.6 and Lemma 2.3.1 here as they are critical lemmas for proving the convergence.

Lemma 3.2.3. *[140, Lemma 4.10] Assume that a^* is an isolated accumulation point of a sequence $\{a_k\}$ such that for every subsequence $\{a_{k_j}\}$ converging to a^*, there is an infinite subsequence $\{a_{k_{j_i}}\}$ such that $|a_{k_{j_i}+1} - a_{k_{j_i}}| \to 0$. Then the whole sequence $\{a_k\}$ converges to a^*.*

Lemma 3.2.4. *[70] Given a matrix $A \in \mathbb{R}^{m\times n}$, then the global maximum of the generalized Rayleigh quotient*

$$\max_{\substack{\mathbf{y}\in\mathbb{R}^m,\,\|\mathbf{y}\|=1\\ \mathbf{z}\in\mathbb{R}^n,\,\|\mathbf{z}\|=1}} \mathbf{y}^\top A\mathbf{z} \tag{3.12}$$

is precisely the largest singular value σ_1 of A, where the global maximizer $(\mathbf{y}_1,\mathbf{z}_1)$ consists of precisely the corresponding left and right singular vectors. The best rank-1 approximation to A is given by $\sigma_1\mathbf{y}_1\mathbf{z}_1^\top$.

3.3 SVD-based best rank-1 approximation

We now investigate a possible application of the singular value decomposition (SVD) to the best rank-1 approximation of a generic tensor. We shall explain the kind of generic property we need in the context. Our goal is to achieve the best rank-1 approximation by improving two components a time via the SVD.

3.3.1 SVD certification

Suppose that $\lambda\bigotimes_{\ell=1}^{k}\overline{\mathbf{u}}^{(\ell)}$ is the best rank-1 approximation to a given order-k tensor T. By (3.2), the generalized Rayleigh quotient $\lambda=\langle T,\bigotimes_{\ell=1}^{k}\overline{\mathbf{u}}^{(\ell)}\rangle$ is positive and maximal. Consider an arbitrary partitioning $[\![k]\!]=\boldsymbol{\alpha}\cup\boldsymbol{\beta}$ with the cardinality $|\boldsymbol{\beta}|=2$. By Lemma 3.2.1, we can write

$$\lambda=\langle T \circledast_{\boldsymbol{\beta}} \bigotimes_{i=1}^{k-2}\overline{\mathbf{u}}^{(\alpha_i)},\bigotimes_{j=1}^{2}\overline{\mathbf{u}}^{(\beta_j)}\rangle.$$

The product $C_{\boldsymbol{\beta}}:=T\circledast_{\boldsymbol{\beta}}\bigotimes_{i=1}^{k-2}\overline{\mathbf{u}}^{(\alpha_i)}$ is a matrix in $\mathbb{R}^{I_{\beta_1}\times I_{\beta_2}}$. Since λ is the maximal generalized Rayleigh quotient, by Lemma 3.2.4, we conclude that $\overline{\mathbf{u}}^{(\beta_1)}$ and $\overline{\mathbf{u}}^{(\beta_2)}$ must be the left and the right singular vectors associated with the largest singular value λ of $C_{\boldsymbol{\beta}}$ for any $\boldsymbol{\beta}$. This is the SVD certification of the best rank-1 approximation to a given tensor T.

We are thus motivated to formulate an SVD-based approach to calculate the best rank-1 approximation by iterations. Depending on the choice of $\boldsymbol{\beta}$ which dictates where the certification is to be checked, the approach may appear in different variants. For order-4 tensors, for example, only the two pairs $\boldsymbol{\beta} = (1, 2)$ and $(3, 4)$ are alternatingly checked in [193], whereas all six combinations in the order $\boldsymbol{\beta} = (1, 2), (3, 4), (1, 3), (2, 4), (1, 4), (2, 3)$ are checked in [75]. We propose two alternatives. In Algorithm 1, we circulate through k pairs of $\boldsymbol{\beta}$ in the order $(1, 2), (2, 3), \ldots, (k - 1, k)$ and $(1, k)$. In Algorithm 2, we propose a random choice of $\boldsymbol{\beta} = (\sigma_{k-1}, \sigma_k)$ where σ is an arbitrary permutation of $[\![k]\!]$. We do not think that there is a significant difference in the performance among the variants, but the true verdict is yet to be further investigated. In all algorithms, the most fundamental concern is a proof of convergence for generic tensors.

3.3.2 Algorithm description

The most basic SVD-based approach is outlined in Algorithm 5. Two types of dynamics are involved in this and all other algorithms. One is the dynamics of the objective values, of which the analysis is straightforward. The other is the dynamics of the iterates, which is much harder to characterize. We will discuss the convergence in the next section.

To convey the idea, we adopt the subscript $_{[p]}$ in Algorithm 5 to indicate the quantity at the p-th iteration. Each sweep of p at Line 1 in Algorithm 5 involves k pairs of $\boldsymbol{\beta}$ ranging circularly from $(1, 2), (2, 3), \ldots, (k - 1, k)$ and $(1, k)$. It is tricky that last pair has to be in the order $(1, k)$, as the reversal $(k, 1)$ will not work. Each $\mathbf{u}_{[p+1]}^{(\ell)}$ is updated twice. The first updates for $\ell = 2, \ldots, k$, denoted by $\widehat{\mathbf{u}}_{[p+1]}^{(\ell)}$ at Line 10, are not essential and can be completely removed from the algorithm, but its presence helps bridge the monotonicity. The update $\widehat{\mathbf{u}}_{[p+1]}^{(1)}$ is temporarily overwritten as $\mathbf{u}_{[p+1]}^{(1)}$ at Line 9 for the computation of $C_{[p]}^{(\ell)}$ at Line 4 for $\ell = 2, \ldots, k-1$, but will be updated again at Line 17. The switch of signs at Line 7 conditioned upon Line 6 is to ensure that the iterates are aligned in one direction and thus avoid discontinuous

Algorithm 5 (Best rank-1 approximation via SVD updating with cyclic progression.)

Input: An order-k tensor $T \in \mathbb{R}^{I_1 \times \dots \times I_k}$ and k starting unit vectors $\mathbf{u}^{(\ell)}_{[0]} \in \mathbb{R}^{I_\ell}$, $\ell \in [\![k]\!]$

Output: A local best rank-1 approximation to T

```
1: for $p = 0, 1, \cdots,$ do
2:     for $\ell = 1, 2, \cdots, k-1,$ do
3:         $\boldsymbol{\beta}_\ell = (\ell, \ell+1)$
4:         $C^{(\ell)}_{[p]} = T \circledast_{\boldsymbol{\beta}_\ell} \bigotimes_{i=1}^{\ell-1} \mathbf{u}^{(i)}_{[p+1]} \otimes \bigotimes_{i=\ell+2}^{k} \mathbf{u}^{(i)}_{[p]}$    {A matrix of size $I_\ell \times I_{\ell+1}$}
5:         $[\mathbf{u}, s, \mathbf{v}] = \mathsf{svds}(C^{(\ell)}_{[p]}, 1)$    {Dominant singular value triplet via Matlab
           routine svds; assume uniqueness}
6:         if $u_1 < 0$ then
7:             $\mathbf{u} = -\mathbf{u}, \mathbf{v} = -\mathbf{v}$    {Assume the generic case that $\mathbf{u}_1 \neq 0$; otherwise, use
               another entry.}
8:         end if
9:         $\mathbf{u}^{(\ell)}_{[p+1]} := \mathbf{u}$    {If $\ell = 1$, this is $\widehat{\mathbf{u}}^{(1)}_{[p+1]}$; otherwise this is the second update
           $\mathbf{u}^{(\ell)}_{[p+1]}$, if $2 \leq \ell < k$.}
10:        $\widehat{\mathbf{u}}^{(\ell+1)}_{[p+1]} := \mathbf{v}$    {Skipping this step will not affect $C^{(\ell+1)}_{[p]}$ at Line 4.}
11:        $\lambda^{(\ell)}_{[p+1]} := s$
12:    end for
13:    $\boldsymbol{\beta}_k = (1, k)$    {Not $(k, 1)$!}
14:    $C^{(k)}_{[p]} = T \circledast_{\boldsymbol{\beta}_k} \bigotimes_{i=2}^{k-1} \mathbf{u}^{(i)}_{[p+1]}$    {A matrix of size $I_1 \times I_k$}
15:    $[\mathbf{u}, s, \mathbf{v}] = \mathsf{svds}(C^{(k)}_{[p]}, 1)$    {Dominant singular value triplet via Matlab routine
       svds; assume uniqueness}
16:    $\mathbf{u}^{(k)}_{[p+1]} := \mathbf{v}$    {After adjusting the signs of $\mathbf{u}$ and $\mathbf{v}$ properly as in Line 6.}
17:    $\mathbf{u}^{(1)}_{[p+1]} := \mathbf{u}$
18:    $\lambda^{(k)}_{[p+1]} := s$
19: end for
```

jumps. The continuity of the dominant singular value and the associated singular vector is critical to convergence. The intermediate values $\lambda_{[p+1]}^{(\ell)}$ are registered in the algorithm as well, even though only $\lambda_{[p+1]}^{(k)}$ at the final stage is crucial.

The above algorithm is still alternating in nature, but is different from the alternating least squares (ALS) approach that has been popular for computing the best rank-1 approximation [50, 109, 197]. The most significant difference is that, since the dominant singular vector $\mathbf{u}_{[p+1]}^{(\ell)}$ and $\mathbf{v}_{[p+1]}^{(\ell)}$ of the matrix $C_{[p]}^{(\ell)}$ gives rise to the absolute maximal value $\lambda_{[p+1]}^{(\ell)}$ for the functional

$$g(\mathbf{x}, \mathbf{y}) := \langle T, \bigotimes_{i=1}^{\ell-1} \mathbf{u}_{[p+1]}^{(i)} \otimes \mathbf{x} \otimes \mathbf{y} \otimes \bigotimes_{i=\ell+2}^{k} \mathbf{u}_{[p]}^{(i)} \rangle \tag{3.13}$$

among all possible vectors $\mathbf{x}$ and $\mathbf{y}$, the mechanism of updating $\mathbf{x}$ and $\mathbf{y}$ simultaneously in Algorithm 5 is going to increase the generalized Rayleigh quotient faster than the combination of two applications of ALS approach to $\mathbf{x}$ followed by $\mathbf{y}$ in one step, provided that the initial information is the same. The two-in-one gain is also better than the maximum of updating $\mathbf{x}$ or $\mathbf{y}$ separately [75, Proposition 4]. We stress that such an advantage happens only when the comparison is made at the same point. There is no general theory at present to support that the SVD update will continue to be superior to the power update in the long run, once they depart toward different directions from the same starting point.

Other than for systematically bookkeeping the progression of $\boldsymbol{\beta}$, there is no particular reason that we have to cycle through the ℓ-loop as is indicated in Algorithm 5. An alternative way is to shuffle the columns $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(k)}$ by a random permutation σ and generate a matrix C for updating. This randomized procedure is modified at Line 7 in Algorithm 6. To avoid confusion with data generated from Algorithm 5, we employ a slightly different notation when describing the progression in this algorithm. For simplicity, we always choose to update the last two vectors $\mathbf{u}^{(\sigma_{k-1})}, \mathbf{u}^{(\sigma_k)}$ after the permutation. It is known in probability theory that the expected number of trials for a permutation to recur is $\frac{k(k-1)}{2}$. Nonetheless, by the time that a repetition

Algorithm 6 (Best rank-1 approximation via SVD updating with randomization.)

Input: An order-k generic tensor T and k starting unit vectors $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(k)} \in \mathbb{R}^n$

Output: A local best rank-1 approximation to T

```
1: $t \leftarrow 0$
2: $\lambda_0 \leftarrow \langle T, \bigotimes_{\ell=1}^{k} \mathbf{u}^{(\ell)} \rangle$
3: repeat
4:     $t \leftarrow t + 1$
5:     $\sigma \leftarrow$ random permutation of $\{1, \ldots, k\}$
6:     $\boldsymbol{\beta}_t \leftarrow (\sigma_{k-1}, \sigma_k)$
7:     $C_t \leftarrow T \circledast_{\boldsymbol{\beta}_t} \bigotimes_{i=1}^{k-2} \mathbf{u}^{(\sigma_i)}$
8:     $[\mathbf{u}_t, s_t, \mathbf{v}_t] = \mathsf{svds}(C_t, 1)$ {Dominant singular value triplet via Matlab routine
       svds, assume uniqueness}
9:     if $(\mathbf{u}_t)_1 < 0$ then
10:        $\mathbf{u} = -\mathbf{u}_t, \mathbf{v} = -\mathbf{v}_t$ {Assume the general case that $(\mathbf{u}_t)_1 \neq 0$; otherwise, use
           another entry}
11:    end if
12:    $\lambda_t \leftarrow s_t$
13:    $\mathbf{u}^{(\sigma_{k-1})} \leftarrow \mathbf{u}_t, \mathbf{u}^{(\sigma_k)} \leftarrow \mathbf{v}_t$
14: until $\lambda_t$ meets convergence criteria
```

of permutation $\boldsymbol{\beta}_t$ occurs, the vectors $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(k)}$ should have been changed. Repeating the random permutations sufficiently many times should get the iteration to move forward. The concern of reiterating with the same matrix C_t at Line 7 should be nominal. It is interesting to note from our numerical experiments in Section 3.5 that this randomized algorithm turns out to be the most efficient when comparing with other variants.

3.4 Convergence analysis

In this section, we analyze the convergence for the above algorithms. First, because the SVD at each update always selects the dominant singular value and the corresponding left and right singular vectors, each of the two algorithms enjoys the property that the corresponding sequence of the generalized Rayleigh quotients is bounded and monotone increasing. The convergence of the objective values (3.2) is obvious.

Lemma 3.4.1. *The scalars $\{\lambda_{[p]}^{(\ell)}\}$ generated in Algorithm 5 form a monotone convergent sequence for each $\ell = 1, \ldots, k$ and all converge to the same value.*

Lemma 3.4.2. *The scalars $\{\lambda_t\}$ generated in Algorithm 6 form a monotone convergent sequence.*

It remains to prove the convergence of iterates themselves under generic conditions [193, Assumption 3.1]. What happens is that there are cases where the iterates do not converge [114], but these cases form algebraic varieties, i.e., zeros of a certain polynomial system, that are of measure zero in the space of general tensors. The complement of this zero measure set is open and dense under the Zariski topology [164], which is what we referred to as generic. To avoid using jargons from algebraic geometry, we shall be more specific in the following argument when generic properties are required.

We learn recently that authors of the report [193] independently prove the convergence of their variant of an SVD-based algorithm by exploiting the monotone convergence of values $\lambda_{[p]}^{(k)}$ and λ_t. Their proof relies on the framework developed in [10] and utilizes the the Łojasiewicz gradient inequality. A similar idea has been employed in our earlier work in [189]. Our contribution in this chapter is a new, shorter, and more direct proof. In either case, the analysis should fulfill what was declared as ”we do not have a complete understanding when this will happen” in [75, Page 947].

3.4.1 Convergence of Algorithm 5

The two SVD-based algorithms outlined in the proceeding section differ by the way $\boldsymbol{\beta}$ is specified. To convey our idea, we first characterize the limiting behavior of Algorithm 5 where $\boldsymbol{\beta}$ is changed systematically in a cyclic pattern. Observe that for each fixed ℓ, because $\|\mathbf{u}_{[p]}^{(\ell)}\|_2 = 1$ for all p, the collection $\{\mathbf{u}_{[p]}^{(\ell)}\}$ must have a convergent subsequence. There are only finitely many ℓ. Selecting a subsequence of a subsequence if necessary, we can find a common subset $\{p_j\}$ of nonnegative integers so that $\{\mathbf{u}_{[p_j]}^{(\ell)}\}$ converges simultaneously for all $\ell \in [\![k]\!]$.

Lemma 3.4.3. *If subsequences $\{\mathbf{u}_{[p_j]}^{(\ell)}\}$ generated by Algorithm 5 converge simultaneously for all $\ell \in [\![k]\!]$, then so do subsequences $\{C_{[p_j]}^{(\ell)}\}$ and $\{\mathbf{u}_{[p_j+1]}^{(\ell)}\}$.*

Proof. The simultaneous convergence of $\{\mathbf{u}_{[p_j]}^{(\ell)}\}$ for $\ell = 3, \ldots, k$ implies that the subsequence $\{C_{[p_j]}^{(1)}\}$ converges. By the continuity inherited in the SVD [29, 192], the subsequence of the left singular vectors $\{\widehat{\mathbf{u}}_{[p_j+1]}^{(1)}\}$ of $C_{[p_j]}^{(1)}$ converges also since we have already aligned them in one direction. But then by definition, $\{C_{[p_j]}^{(2)}\}$ converges and, thus, so does $\{\mathbf{u}_{[p_j+1]}^{(2)}\}$. We can repeat this argument by cycling through the ℓ-loop in Algorithm 5. At the end, the matrices $\{C_{[p_j]}^{(k)}\}$ together with the corresponding left singular vectors $\{\mathbf{u}_{[p_j+1]}^{(1)}\}$ and the right singular vectors $\{\mathbf{u}_{[p_j+1]}^{(k)}\}$ must also converge. □

Denote the respective limit points of the above subsequences by

$$\begin{cases} \lim_{j\to\infty} \mathbf{u}_{[p_j]}^{(\ell)} & = & \mathbf{u}_{\natural}^{(\ell)}, \\ \lim_{j\to\infty} C_{[p_j]}^{(\ell)} & = & C^{(\ell)}, \\ \lim_{j\to\infty} \widehat{\mathbf{u}}_{[p_j+1]}^{(\ell)} & = & \widehat{\mathbf{u}}_{\sharp}^{(\ell)}, \\ \lim_{j\to\infty} \mathbf{u}_{[p_j+1]}^{(\ell)} & = & \mathbf{u}_{\sharp}^{(\ell)}, \end{cases} \qquad \ell \in [\![k]\!], \tag{3.14}$$

where the subscript $_\sharp$ is a handy way to distinguish, at least for now, that the limit points corresponding to the subsequence $_{[p_j+1]}$ might be different from those, denoted by the subscript $_\natural$, of the original subsequence $_{[p_j]}$. By the way $C_{[p_j]}^{(\ell)}$ is defined, it follows that

$$C^{(\ell)} = T\circledast_{\boldsymbol{\beta}_\ell} (\bigotimes_{i=1}^{\ell-1} \mathbf{u}_\sharp^{(i)} \otimes \bigotimes_{i=\ell+2}^{k} \mathbf{u}_\natural^{(i)}). \tag{3.15}$$

We already point out in Lemma 3.4.1 that all matrices $C^{(\ell)}$, $\ell \in \llbracket k \rrbracket$ share the same dominant singular value $\widetilde{\lambda}$. We now explore the relationships among dominant singular vectors of all $C^{(\ell)}$ matrices.

With respect to a given convergent subsequence generated by Algorithm 5, the following result asserts that all dominant (left) singular vectors are the same.

Lemma 3.4.4. *Assume that T is such that, with respect to the given simultaneously convergent subsequences $\{\mathbf{u}_{[p_j]}^{(\ell)}\}$ generated by Algorithm 5, the dominant singular value $\widetilde{\lambda}$ of the corresponding limit point $C^{(\ell)} \in \mathbb{R}^{I_\ell \times I_{\ell+1}}$ defined in (3.14) is simple for all $\ell \in \llbracket k \rrbracket$. Then the limit points defined in (3.14) satisfy the relationships that*

$$\left\{ \begin{array}{rcl} \mathbf{u}_\natural^{(\ell)} & = & \widehat{\mathbf{u}}_\sharp^{(\ell)} = \mathbf{u}_\sharp^{(\ell)}, \\ C^{(\ell)} & = & T\circledast_{\boldsymbol{\beta}_\ell} (\bigotimes_{i=1}^{\ell-1} \mathbf{u}_\natural^{(i)} \otimes \bigotimes_{i=\ell+2}^{k} \mathbf{u}_\natural^{(i)}), \end{array} \right. \qquad \ell \in \llbracket k \rrbracket. \tag{3.16}$$

Proof. For convenience, we employ the abbreviations $\circledast_1$ and $\circledast_2$ to indicate, respectively, the row-matrix and matrix-column multiplications already delineated in (3.6). In reality, it must be noted that we are dealing with multiplications of matrices and vectors of different sizes.

By using Lemma 3.2.2, we first observe the equalities

$$\begin{array}{rcl} C_{[p_j]}^{(\ell)} \circledast_1 \mathbf{u}_{[p_j+1]}^{(\ell)} & = & \lambda_{[p_j+1]}^{(\ell)} \widehat{\mathbf{u}}_{[p_j+1]}^{(\ell+1)} \\ & = & (T\circledast_{\boldsymbol{\beta}_\ell} (\widehat{\mathbf{u}}_{[p_j+1]}^{(1)} \otimes \bigotimes_{i=2}^{\ell-1} \mathbf{u}_{[p_j+1]}^{(i)} \otimes \bigotimes_{i=\ell+2}^{k} \mathbf{u}_{[p_j]}^{(i)})) \circledast_1 \mathbf{u}_{[p_j+1]}^{(\ell)} \\ & = & (T\circledast_{\boldsymbol{\beta}_{\ell+1}} (\widehat{\mathbf{u}}_{[p_j+1]}^{(1)} \otimes \bigotimes_{i=2}^{\ell} \mathbf{u}_{[p_j+1]}^{(i)} \otimes \bigotimes_{i=\ell+3}^{k} \mathbf{u}_{[p_j]}^{(i)})) \circledast_2 \mathbf{u}_{[p_j]}^{(\ell+2)} \\ & = & C_{[p_j]}^{(\ell+1)} \circledast_2 \mathbf{u}_{[p_j]}^{(\ell+2)}, \quad \ell = 2\ldots, k-2. \end{array} \tag{3.17}$$

Similarly,

$$C_{[p_j]}^{(1)} \circledast_1 \widehat{\mathbf{u}}_{[p_j+1]}^{(1)} \quad = \quad \lambda_{[p_j+1]}^{(1)} \widehat{\mathbf{u}}_{[p_j+1]}^{(2)}$$

$$
\begin{aligned}
&= (T\circledast_{\boldsymbol{\beta}_1} (\bigotimes_{i=3}^{k} \mathbf{u}_{[p_j]}^{(i)}))\circledast_1 \widehat{\mathbf{u}}_{[p_j+1]}^{(1)} \\
&= (T\circledast_{\boldsymbol{\beta}_2} (\widehat{\mathbf{u}}_{[p_j+1]}^{(1)} \otimes \bigotimes_{i=4}^{k} \mathbf{u}_{[p_j]}^{(i)}))\circledast_2 \mathbf{u}_{[p_j]}^{(3)} = C_{[p_j]}^{(2)}\circledast_2 \mathbf{u}_{[p_j]}^{(3)}. \qquad (3.18)
\end{aligned}
$$

The special "twist" at Lines 16-17 with $\boldsymbol{\beta}_k = (1, k)$ in Algorithm 5 allows us to have the identities

$$
\begin{aligned}
C_{[p_j-1]}^{(k)}\circledast_2 \mathbf{u}_{[p_j]}^{(k)} &= \lambda_{[p_j]}^{(k)} \mathbf{u}_{[p_j]}^{(1)} \\
&= (T\circledast_{\boldsymbol{\beta}_k} (\bigotimes_{i=2}^{k-1} \mathbf{u}_{[p_j]}^{(i)}))\circledast_2 \mathbf{u}_{[p_j]}^{(k)} \\
&= (T\circledast_{\boldsymbol{\beta}_1} (\bigotimes_{i=3}^{k} \mathbf{u}_{[p_j]}^{(i)}))\circledast_2 \mathbf{u}_{[p_j]}^{(2)} = C_{[p_j]}^{(1)}\circledast_2 \mathbf{u}_{[p_j]}^{(2)}. \qquad (3.19)
\end{aligned}
$$

Finally, we also have

$$
\begin{aligned}
C_{[p_j]}^{(k-1)}\circledast_1 \mathbf{u}_{[p_j+1]}^{(k-1)} &= \lambda_{[p_j+1]}^{(k-1)} \widehat{\mathbf{u}}_{[p_j+1]}^{(k)} \\
&= (T\circledast_{\boldsymbol{\beta}_{k-1}} (\widehat{\mathbf{u}}_{[p_j+1]}^{(1)} \otimes \bigotimes_{i=2}^{k-2} \mathbf{u}_{[p_j+1]}^{(i)}))\circledast_1 \mathbf{u}_{[p_j+1]}^{(k-1)} \\
&= (T\circledast_{\boldsymbol{\beta}_k} (\bigotimes_{i=2}^{k-1} \mathbf{u}_{[p_j+1]}^{(i)}))\circledast_1 \widehat{\mathbf{u}}_{[p_j+1]}^{(1)} = C_{[p_j]}^{(k)}\circledast_1 \widehat{\mathbf{u}}_{[p_j+1]}^{(1)}. \qquad (3.20)
\end{aligned}
$$

Taking the limits, then it follows by construction and continuity that we have the relationships:

$$
\begin{aligned}
& \widetilde{\lambda}\mathbf{u}_{\natural}^{(1)} && = C^{(1)}\circledast_2 \mathbf{u}_{\natural}^{(2)}, \quad \text{(by (3.19))} \\
C^{(1)}\circledast_1 \widehat{\mathbf{u}}_{\sharp}^{(1)} = {} & \widetilde{\lambda}\widehat{\mathbf{u}}_{\sharp}^{(2)} && = C^{(2)}\circledast_2 \mathbf{u}_{\natural}^{(3)}, \quad \text{(by (3.18))} \\
C^{(\ell)}\circledast_1 \mathbf{u}_{\sharp}^{(\ell)} = {} & \widetilde{\lambda}\widehat{\mathbf{u}}_{\sharp}^{(\ell+1)} && = C^{(\ell+1)}\circledast_2 \mathbf{u}_{\natural}^{(\ell+2)}, \quad \ell = 2, \ldots, k-2, \quad \text{(by (3.17))} \\
C^{(k-1)} \circledast_1 \mathbf{u}_{\sharp}^{(k-1)} = {} & \widetilde{\lambda}\widehat{\mathbf{u}}_{\sharp}^{(k)} && = C^{(k)} \circledast_1 \widehat{\mathbf{u}}_{\sharp}^{(1)}, \quad \text{(by (3.20))} \\
C^{(k)} \circledast_1 \mathbf{u}_{\sharp}^{(1)} = {} & \widetilde{\lambda}\mathbf{u}_{\sharp}^{(k)}. && \qquad \text{(by Lines 16-17 in Algorithm 5)}
\end{aligned}
$$

By assumption, the dominant $\widetilde{\lambda}$ is simple and the corresponding singular vector is unique up to a sign change. However, because in Lines 6 to 8 of Algorithm 5 we have already required the first entry of the dominant singular vector to be positive, such a sign change does not exist. The best rank-1 approximation to the matrix $C^{(\ell)}$ therefore is unique. Recursively, the above relationships imply that the best rank-1 approximation to the matrix $C^{(\ell)}$ can be expressed in two ways:

$$\begin{cases} C^{(1)} & \approx & \widetilde{\lambda}\mathbf{u}_{\natural}^{(1)} \otimes \mathbf{u}_{\natural}^{(2)} & = & \widetilde{\lambda}\widehat{\mathbf{u}}_{\sharp}^{(1)} \otimes \widehat{\mathbf{u}}_{\sharp}^{(2)}, \\ C^{(\ell)} & \approx & \widetilde{\lambda}\widehat{\mathbf{u}}_{\sharp}^{(\ell)} \otimes \mathbf{u}_{\natural}^{(\ell+1)} & = & \widetilde{\lambda}\mathbf{u}_{\sharp}^{(\ell)} \otimes \widehat{\mathbf{u}}_{\sharp}^{(\ell+1)}, \quad \ell = 2, \ldots, k-1, \\ C^{(k)} & \approx & \widetilde{\lambda}\widehat{\mathbf{u}}_{\sharp}^{(1)} \otimes \widehat{\mathbf{u}}_{\sharp}^{(k)} & = & \widetilde{\lambda}\mathbf{u}_{\sharp}^{(1)} \otimes \mathbf{u}_{\sharp}^{(k)}. \end{cases} \tag{3.21}$$

By the uniqueness of dominant singular vectors for $C^{(\ell)}$ for each $\ell \in [\![k]\!]$, the assertion (3.16) follows from (3.21). □

The question is when the assumption imposed on T in Lemma 3.4.4 will hold. Specifically, let Ω denote the set of tensor $T \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_k}$ where there exists a convergent subsequence $\{\mathbf{u}_{[p_j]}^{(\ell)}\}$ such that the dominant singular value of the limit point $C^{(\ell)}$ of the corresponding $\{C_{[p_j]}^{(\ell)}\} \subset \mathbb{R}^{I_\ell \times I_{\ell+1}}$ is not simple. How large is the set Ω?

Consider the fact that symmetric matrices with multiple eigenvalues form an algebraic variety of codimension two [56]. For almost all matrices, therefore, the largest singular value is simple. For one particular limit point $C^{(\ell)}$ to have multiple dominant singular values, the subsequence $\{C_{[p_j]}^{(\ell)}\}$ that leads to it must approach arbitrarily close to that variety of matrices with multiple dominant singular values. But $\{C_{[p_j]}^{(\ell)}\}$ is defined in a specific algebraic way as in Line 4 of Algorithm 5. Backward tracing, the occurrence of $C^{(\ell)}$ with multiple dominant singular values depends on the set $\{\mathbf{u}_{[0]}^{(i)}, i \in [\![k]\!]\}$ of unit starting vectors, the particular subsequence $\{_{[p_j]}\}$ selected, and the underlying T. Any change of the starting vectors could alter the course of iteration. The choice of a different subsequence could lead to a different limit point. When both changes do not obliterate the appearance of dominant singular values, the tensor T itself must be something special. We thus conjecture that the set Ω should not be generic. Such a reasoning, of course, does not constitute a mathematical proof to support its genericity because we do not know of an analytic way to quantify a generic T. Thus, at the moment, we can only call it an assumption to be satisfied. Note that Lemma 3.4.4 is subsequence dependent. Its conclusion is with respect to only one particularly given convergent subsequence. The following condition is much stronger than what we need in Lemma 3.4.4.

Condition A. We say that a given order-k tensor T satisfies Condition A if for every convergent subsequences $\{\mathbf{u}_{[p_j]}^{(\ell)}\}$ generated by Algorithm 5 and the corresponding subsequence $\{C_{[p_j]}^{(\ell)}\}$, $\ell \in [\![k]\!]$, the dominant singular value $\widetilde{\lambda}$ of the limit point $C^{(\ell)} \in \mathbb{R}^{I_\ell \times I_{\ell+1}}$ defined in (3.14) is simple for all $\ell \in [\![k]\!]$.

We conjecture that Condition A holds for almost all order-k tensors. Even if not, keep in mind that it will be considered together with the Condition B which is generic and will be described below.

By the way the iteration is defined, and if Lemma 3.4.4 holds, any stationary point of Algorithm 5 necessarily satisfies the system of equations

$$T \circledast_\ell (\bigotimes_{i=1}^{\ell-1} \mathbf{u}^{(i)} \otimes \bigotimes_{i=\ell+1}^{k} \mathbf{u}^{(i)}) = \langle T, \bigotimes_{\ell=1}^{k} \mathbf{u}^{(\ell)} \rangle \mathbf{u}^{(\ell)}, \quad \ell \in [\![k]\!]. \tag{3.22}$$

The equation (3.22) is a polynomial system in the unknowns $(\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(k)}) \in \mathbb{R}^{I_1} \times \ldots \times \mathbb{R}^{I_k}$ with leading coefficients from entries of T. By the theory of parameter continuation [164, Theorem 7.1.1], we know that for almost all tensor $T \in \mathbb{C}^{I_1 \times \ldots \times I_k}$, except for an affine algebraic subset of codimension one in $\mathbb{C}^{I_1} \times \ldots \times \mathbb{C}^{I_k}$, the solutions to (3.22) are isolated. Together with the fact that $\mathbb{R}$ is dense in $\mathbb{C}$ under the Zariski topology, the real solutions are also isolated.

Condition B. We say that a given order-k tensor T satisfies Condition B if the real solutions to the corresponding polynomial system (3.22) are isolated.

Lemma 3.4.5. *For almost all tensors T, the accumulation points of the sequence $\{\mathbf{u}_t\}$ generated by Algorithm 5 and Algorithm 6 are geometrically isolated.*

Finally, we are ready to claim our major result which serves as the theoretical basis complementing the SVD-based Algorithm 5. We say that a tensor $T \in \mathbb{R}^{I_1 \times \ldots \times I_k}$ is generic if it satisfies both generic conditions A and B. The non-generic tensors must reside on some algebraic varieties and, hence, are of measure zero.

Theorem 3.4.1. *For almost all order-k tensors T satisfying Condition A and for arbitrary starting points, the vector sequence $\{(\mathbf{u}_{[p]}^{(1)}, \ldots, \mathbf{u}_{[p]}^{(k)})\}$ generated by Algorithm 5 converges to a local maximizer of the generalized Rayleigh quotient $\lambda(\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(k)})$ defined in (3.2).*

Proof. Let $\{\mathbf{u}_{[p_j]}^{(\ell)}\}$ be any simultaneously convergent subsequences for $\ell \in [\![k]\!]$. By Lemma 3.4.3, the subsequences $\{\mathbf{u}_{[p_j+1]}^{(\ell)}\}$ also converge. Indeed, by Lemma 3.4.4, both subsequences converge to the same limit point for all $\ell \in [\![k]\!]$. Thus $\|\mathbf{u}_{[p_j+1]}^{(\ell)} - \mathbf{u}_{[p_j]}^{(\ell)}\| \to 0$. On the other hand, by Lemma 3.4.5 we assume that the limit point is geometrically isolated. The convergence of the entire sequence $\{\mathbf{u}_{[p_j]}^{(\ell)}\}$ to the same limit point follows from Lemma 3.2.3. □

3.4.2 Convergence of Algorithm 6

Now we argue the convergence of Algorithm 6 where $\boldsymbol{\beta}$ is changed randomly. For clarity, enumerate the column vectors at the end of Line 13 by $\left\{\mathbf{u}_t^{(1)}, \ldots, \mathbf{u}_t^{(k)}\right\}$. By construction, for $t \geq 1$, only two of these vectors are updated by the dominant left and right singular vector of C_t while others remain the same. Now we establish the following result.

Theorem 3.4.2. *For almost all order-k tensors T and arbitrary starting points, the vector sequence $\{(\mathbf{u}_t^{(1)}, \ldots, \mathbf{u}_t^{(k)})\}$ generated by Algorithm 6 converges to a local maximizer of the generalized Rayleigh quotient defined in (3.2).*

Proof. Suppose that $\{\mathbf{u}_{t_i}^{(\ell)}\}$ is an arbitrary subsequence converging to $\overline{\mathbf{u}}^{(\ell)}$ simultaneously for $\ell \in [\![k]\!]$. Suppose also by Lemma 3.4.5 that the limit points $\overline{\mathbf{u}}^{(\ell)}$, $\ell \in [\![k]\!]$, are geometrically isolated. By construction, i.e., Line 7 in Algorithm 6, the subsequence $\{C_{t_i+1}^{(\ell)}\}$ of matrices converges. By continuity, the subsequence $\left\{\mathbf{u}_{t_i+1}^{(\ell)}\right\}$ must also converge to $\overline{\mathbf{u}}^{(\ell)}$. In particular, $\left\|\mathbf{u}_{t_i+1}^{(\ell)} - \mathbf{u}_{t_i}^{(\ell)}\right\| \to 0$. The condition in Lemma 3.2.3 therefore is satisfied. It follows that the whole sequence $\{\mathbf{u}_t^{(\ell)}\}$ converges to $\overline{\mathbf{u}}^{(\ell)}$. □

We remark that in our recent work for symmetric tensors [79], we have also proposed an SVD-based algorithm by using a mechanism of random update similar

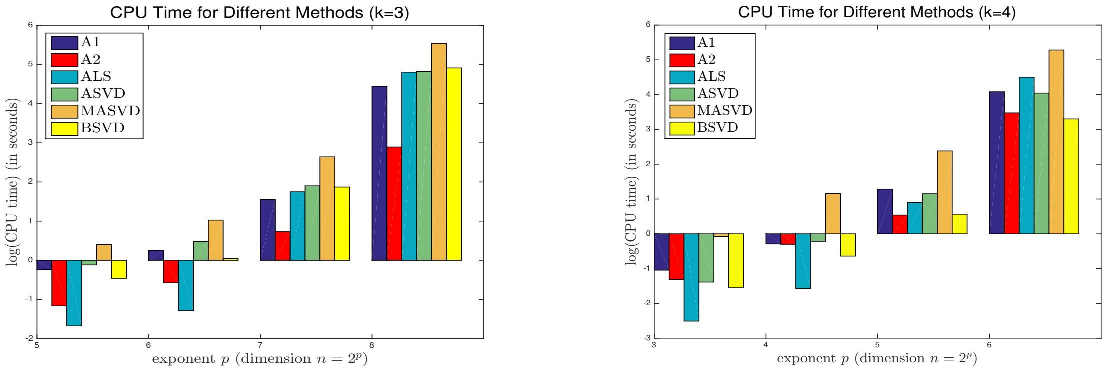

Figure 3.1: Comparison of CPU time among different methods.

to that adopted in Algorithm 6. The analysis there, in order to maintain symmetry, is much more involved than what we have shown above for non-symmetric tensors.

3.5 Numerical experiments

The idea of updating two factors simultaneously by taking advantage of the two-in-one global optimization property of SVD is appealing. Thus we investigate and offer a theoretical justification that the two variants of SVD-based methods described in this chapter indeed converge. Two questions naturally arise. First, is there a significant difference in performance among different SVD-based algorithms? Second, is the SVD-based algorithm always superior to the conventional ALS method? Although rigorous numerical testing is not the main objective of this chapter, we carry out some preliminary experiments with the hope of partially satisfying our own curiosity.

Experiment 1. To our knowledge, there are at least five variants of SVD-based algorithms. These are the ASVD [75], the MASVD [75], the block SVD (BSVD) [193], as well as Algorithm 5 and Algorithm 6 introduced in this manuscript. We are interesting in comparing the CPU time required for the iterates to meet the same stopping criteria – the iteration terminates automatically when the generalized Rayleigh quotients do not vary more than the tolerance 10^{-5} in three consecutive iterations. As a general reference point, also included is the performance of the conventional ALS method. To check the scalability, we consider the case where all factors are of same dimension and vary the size of problem as $n = 2^p$ where we choose $p = 5, \ldots, 8$ for $k = 3$ and $p = 3, \ldots, 6$ for $k = 4$. Each case of p is repeatedly tested 20 times with random starting unit vectors. We plot average running time in Figure 3.1. It should be pointed out that even from the same starting points, different methods may converge to different limit points because they involve different dynamics. Regardless, the comparison is based on the same starting points subject to the same stopping criteria for convergence.

For problems of modest sizes, e.g., $n = 2^6$ for $k = 3$ and $n = 2^4$ for $k = 4$, the cost of SVD computation outruns that of the high-order power method. Thus the ALS method uses less time at these modest dimensions which amounts to a full and dense tensor with approximately 2^{16} to 2^{18} entries. For odd order tensors, the block structure in the BSVD necessarily updates one vector via the ALS algorithm, which slows down its convergence. Thus we see that for order-3 tensors, Algorithm 6 outperforms the BSVD when $p > 6$. For order-4 tensors, both Algorithm 6 and the BSVD method in [193] update two distinct vectors simultaneously, thus are about equally fast. On the other hand, the cyclic progression of Algorithm 5 updates each factor twice but only the second time counts. So, it literately updates one factor a time. As such, it should always be less effective than Algorithm 6. The MASVD requires multiple ASVD calculation, so it is more expensive than ASVD. The ASVD checks through all possible permutations, so its performance is about the same as that of the Algorithm 5.

Experiment 2. In this experiment, we want to assess the quality of the iterates generated by the SVD-based method and the ALS method. To fix the idea, we consider a given order-4 tensor with $n = 2^6$. We use the ratio $\frac{\lambda_{SVD} - \lambda_{ALS}}{\|T\|}$ as the measurement of quality. The idea is that the larger the generalized Rayleigh quotient, the better the quality of the iteration. A positive ratio means that the SVD method is improving better than the ALS method, and vice versa. We perform

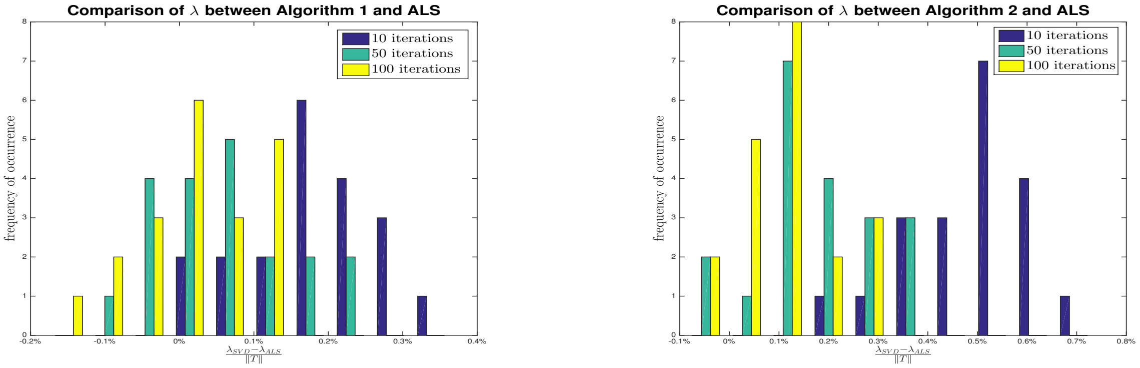

Figure 3.2: Comparison of quality between SVD and ALS.

the iteration 10, 50, and 100 times for each of the 20 randomly generated initial vectors for the same tensor and measure the ratios. These iteration steps are far away from convergence, but speak of an important trend. Plotted in Figure 3.2 are the histograms of ratios in 10 bins, when the comparison is made with respect to Algorithm 5 and Algorithm 6, respectively. Note that in the initial 10 iterations, the (dark blue) lobe of ratios leans toward the right of 0, indicating that the SVD method gives better improvement than the ALS method. However, when sufficiently many iterations are taken, the statistics in Figure 3.2 clearly shows that the (yellow) lobe shifts toward the left, indicating that the ALS method is gradually catching up the quality. In certain case, the ratio is negative, indicating that the ALS method might lead to a better local optimum eventually, although we have not seen the ultimate convergence yet. Comparing the two histograms in Figure 3.2, we also notice that Algorithm 6 generally keeps more positive ratios than Algorithm 5 does.

Chapter 4

Orthogonal Low Rank Approximation

4.1 Orthogonal low rank approximation

A tensor of the form $\bigotimes_{i=1}^{k} \mathbf{u}^{(i)} = \mathbf{u}^{(1)} \otimes \cdots \otimes \mathbf{u}^{(k)} := \left[u_{i_1}^{(1)} \cdots u_{i_k}^{(k)} \right]$ where elements are the products of entries from vectors $\mathbf{u}^{(i)} \in \mathbb{R}^{I_i}$, $i = 1, \ldots, k$ also referred as components is said to be of rank one [110].

For a given tensor $T \in \mathbb{R}^{I_1 \times \ldots \times I_k}$, low rank approximation studied in this chapter is to minimize

$$\left\| T - \sum_{r=1}^{R} \lambda_r \underbrace{\bigotimes_{i=1}^{k} \mathbf{u}_r^{(i)}}_{H_r} \right\|_F^2, \tag{4.1}$$

subject to the mutual orthogonality condition that

$$\langle H_{r_1}, H_{r_2} \rangle = \prod_{i=1}^{k} \left\langle \mathbf{u}_{r_1}^{(i)}, \mathbf{u}_{r_2}^{(i)} \right\rangle = \delta_{r_1 r_2}, \quad \text{for all} \quad 1 \leq r_1, r_2 \leq R, \tag{4.2}$$

where $\mathbf{u}_r^{(i)}$ are unit vectors for $i = 1, \ldots, k$, $r = 1, \ldots R$. To satisfy the constraints (4.2), one of the following orthogonality conditions has been imposed:

1. Complete orthogonality [40, 110, 114]: For all $i = 1, \ldots, k$, and $1 \leq r_1 \neq r_2 \leq R$, $\langle \mathbf{u}_{r_1}^{(i)}, \mathbf{u}_{r_1}^{(i)} \rangle = 1$, and $\langle \mathbf{u}_{r_1}^{(i)}, \mathbf{u}_{r_2}^{(i)} \rangle = 0$.

2. Orthogonality: For all $i = 1, \ldots, k$, and $1 \leq r \leq R$, $\langle \mathbf{u}_r^{(i)}, \mathbf{u}_r^{(i)} \rangle = 1$, and for

some $1 \leq i_1 < \ldots < i_\mu \leq k$,

$$\left\langle \mathbf{u}_{r_1}^{(i_1)}, \mathbf{u}_{r_2}^{(i_1)} \right\rangle = 0, \ldots, \left\langle \mathbf{u}_{r_1}^{(i_\mu)}, \mathbf{u}_{r_2}^{(i_\mu)} \right\rangle = 0, \quad \forall 1 \leq r_1 \neq r_2 \leq R.$$

3. Semi-orthogonality [40, 166, 190]: For all $i = 1, \ldots, k$, and $1 \leq r \leq R$, $\langle \mathbf{u}_r^{(i)}, \mathbf{u}_r^{(i)} \rangle = 1$ and there is one i such that

$$\langle \mathbf{u}_{r_1}^{(i)}, \mathbf{u}_{r_2}^{(i)} \rangle = 0, \quad \forall 1 \leq r_1 \neq r_2 \leq R.$$

As a result, the low rank approximation problem (4.1) with the constraints (4.2) can be classified as

1. Completely orthogonal low rank approximation:

$$\begin{cases} \min \left\| T - \sum_{r=1}^{R} \lambda_r \bigotimes_{i=1}^{k} \mathbf{u}_r^{(i)} \right\|_F^2, \\ \text{subject to the complete orthogonality constraint.} \end{cases} \tag{4.3}$$

2. Orthogonal low rank approximation:

$$\begin{cases} \min \left\| T - \sum_{r=1}^{R} \lambda_r \bigotimes_{i=1}^{k} \mathbf{u}_r^{(i)} \right\|_F^2, \\ \text{subject to orthogonality constraint.} \end{cases} \tag{4.4}$$

3. Semi-orthogonal low rank approximation:

$$\begin{cases} \min \left\| T - \sum_{r=1}^{R} \lambda_r \bigotimes_{i=1}^{k} \mathbf{u}_r^{(i)} \right\|_F^2, \\ \text{subject to the semi} - \text{orthogonality constraint.} \end{cases} \tag{4.5}$$

Completely orthogonal low rank approximation and semi-orthogonal low rank approximation of tensors have been studied, for example, in [40, 157, 190]. The "workhorse" algorithm for rank-1 approximation and semi-orthogonal low rank approximation of tensor has been alternating least squares (ALS) method which has been proved to be convergence globally for almost all tensors in [189, 190]. The completely orthogonal low rank approximation of tensors is similar to the truncated

SVD of matrices. Obviously, complete orthogonality implies semi-orthogonality, but the converse does not hold. Orthogonality is just a bridge between complete orthogonality and semi-orthogonality since orthogonality is exactly the complete orthogonality if $\mu = k$ and it reduces to semi-orthogonality if $\mu = 1$. The more restricted constraint by adding extra orthogonal factor matrix might be useful for other purpose.

Orthogonal low rank approximation of tensors has been highlighted in [190] and its ALS has been considered for this question. It has been pointed in [190] that "*the technique employed in the preceding section might not be immediately generalizable because we need to prove the convergence of both sequences* $\{\mathbf{v}_{r,[p]}^{(k-1)}\}$ *and* $\{\mathbf{v}_{r,[p]}^{(k)}\}$ *simultaneously. More study is needed*". Furthermore, it has also been addressed in [190] that "*the question of more than one semi-orthogonal factor matrix, except for the case of complete orthogonality, remains open*".

Since orthogonal low rank approximation is an open question addressed in [189] and it includes the completely orthogonal low rank approximation and semi-orthogonal low rank approximation as two special cases, we focus on this open question, i.e., we study orthogonal low rank approximation of tensors in this chapter.

Because of the constraints (4.2), the base tensors H_r $(r = 1, \ldots, R)$ are mutually orthonormal and the expression for the optimal scales λ_r in (4.1) can also be interpreted as the length of the projection of the "vector" T onto the "unit vector" H_r under the Frobenius inner product, and are necessarily given by

$$\lambda_r = \lambda_r \left(\mathbf{u}_r^{(1)} \ldots, \mathbf{u}_r^{(k)}\right) = \left\langle T, \bigotimes_{i=1}^{k} \mathbf{u}_r^{(i)} \right\rangle \quad r = 1, \ldots, R. \tag{4.6}$$

Thus, the orthogonal low rank approximation problem (4.4) can be reformulated as

- Orthogonal low rank approximation:

$$\begin{cases} \max \sum_{r=1}^{R} \lambda_r^2, \\ \text{subject to the orthogonality constraint.} \end{cases} \tag{4.7}$$

In this chapter, for orthogonal low rank approximation problem (4.4) (i.e., (4.7)), we assume without loss of generality that

$$\left\langle \mathbf{u}_{r_1}^{(k-\mu+1)}, \mathbf{u}_{r_2}^{(k-\mu+1)} \right\rangle = 0, \ldots, \left\langle \mathbf{u}_{r_1}^{(k)}, \mathbf{u}_{r_2}^{(k)} \right\rangle = 0, \quad \forall 1 \leq r_1 \neq r_2 \leq R. \tag{4.8}$$

4.2 Summary

In this chapter, we focus on the orthogonal low rank approximation of tensors. Inspired by the SVD-based algorithms for the rank-1 approximation [78, 79] and algorithm in [40], we develop an SVD-based algorithm for orthogonal low rank approximation of tensors, which updates two factors simultaneously and maintain the required orthogonality conditions by means of the polar decomposition. The convergence of our SVD-based algorithm for both objective function and iterates themselves is established. Moreover, numerical experiments have been presented to illustrate the performances of our SVD-based algorithm.

4.2.1 Outline of the chapter

The rest of this chapter is organized as follows: some preliminaries including the basic definitions and notions of tensors are provided in Section 4.3. Then our SVD-based algorithm is developed in Section 4.4, and its global convergence is analyzed in Section 4.5. Numerical experiments are presented in Section 4.6.

4.3 Basics

The following lemma is essentially the well known polar decomposition [89, 90, 102] which reveals the trace maximizing property that will play an important role for the development in the next section.

Lemma 4.3.1. *Let matrix $A \in \mathbb{R}^{m\times n}$ with $m \geq n$ have polar decomposition*

$$A = QS,$$

where $Q \in \mathbb{R}^{m\times n}$ is the column orthogonal polar factor and $S \in \mathbb{R}^{n\times n}$ is the symmetric positive semi-definite factor. Then

$$Q = \arg \max_{P\in\mathbb{R}^{m\times n},\ P^TP=I} \text{Trace}\left(P^T A\right).$$

Moreover, if A is of full column rank, then Q above is unique.

The next Eckart-Young Lemma provides the mechanism of our SVD-based algorithm in next section for updating two factors simultaneously.

Lemma 4.3.2. *[70] Given a matrix $A \in \mathbb{R}^{m\times n}$, then the global maximum of the generalized Rayleigh quotient*

$$\max_{\substack{\mathbf{x}\,\in\,\mathbb{R}^m,\,\|\mathbf{x}\|\,=\,1\\ \mathbf{y}\,\in\,\mathbb{R}^n,\,\|\mathbf{y}\|\,=\,1}} \langle \mathbf{x},\ A\mathbf{y}\rangle \tag{4.9}$$

is precisely the largest singular value σ_1 of A, where the global maximizer (x, y) consists of the corresponding left and right singular vectors of A, and thus the best rank-1 approximation to A is given by $\sigma_1 x^\top y$.

We close this section by one more lemma which will be used to prove the convergence of iterates.

Lemma 4.3.3. *[79, 140, 198] Let $\{a_n\}$ be a bounded sequence of real numbers. If the accumulation points of the sequence $\{a_n\}$ are isolated, and for every subsequence $\{a_{k_j}\}$ converging to a accumulation point a^* there is an infinite subsequence $\{a_{k_{j_i}}\}$ such that $|a_{k_{j_i}+1} - a_{k_{j_i}}| \to 0$, then the whole sequence $\{a_n\}$ converges to a^*.*

4.4 Algorithm for Orthogonal Low Rank Approximation

Followed by our previous work [78, 79] for the rank-1 tensor approximation using SVD and [40] for the completely orthogonal low rank tensor approximation, we apply the similar ideas to the orthogonal low rank approximation problem (4.7) and update two factors simultaneously. There is an additional challenge to maintain the required orthogonality condition in (4.7), we handle this difficulty by means of the polar decomposition [89].

For any $\mathbf{u}_r^{(1)}, \ldots, \mathbf{u}_r^{(\ell)}, \mathbf{u}_r^{(\ell+1)}, \ldots, \mathbf{u}_r^{(k)},\ r = 1, \ldots, R$, it is important to know how we can update to obtain "better" ones.

(i) For any $1 \le \ell \le k - \mu - 1$ and $r = 1, 2, \ldots, R$, let $\beta_\ell = (\ell, \ell+1)$ and

$$C_r^{(\ell)} = T \circledast_{\beta_\ell} \left(\bigotimes_{i=1}^{\ell-1} \mathbf{u}_r^{(i)} \otimes \bigotimes_{i=\ell+2}^{k} \mathbf{u}_r^{(i)} \right).$$

Denote the dominate left and right singular vectors of $C_r^{(\ell)}$ corresponding to its largest singular value by $\tilde{\mathbf{u}}_r^{(\ell)}$ and $\tilde{\mathbf{u}}_r^{(\ell+1)}$, respectively. Then by Lemma 4.3.2, we have

$$\begin{aligned} \lambda_r &= \left\langle T,\ \bigotimes_{\ell=1}^{k} \mathbf{u}_r^{(\ell)} \right\rangle = \left\langle \mathbf{u}_r^{(\ell)},\ C_r^{(\ell)} \mathbf{u}_r^{(\ell+1)} \right\rangle \\ &\le \max_{\mathbf{x}^T\mathbf{x}=1,\ \mathbf{y}^T\mathbf{y}=1} \left\langle \mathbf{x},\ C_r^{(\ell)} \mathbf{y} \right\rangle = \left\langle \tilde{\mathbf{u}}_r^{(\ell)},\ C_r^{(\ell)} \tilde{\mathbf{u}}_r^{(\ell+1)} \right\rangle \\ &= \tilde{\lambda}_r. \end{aligned}$$

Obviously, we can update $\mathbf{u}_r^{(\ell)}$ by $\tilde{\mathbf{u}}_r^{(\ell)}$ and $\mathbf{u}_r^{(\ell+1)}$ by $\tilde{\mathbf{u}}_r^{(\ell+1)}$.

(ii) The Lagrangian for the optimization problem (4.7) (i.e., (4.4)) is

$$\mathcal{L} := \sum_{r=1}^{R} \lambda_r^2 - \sum_{\ell=1}^{k} \sum_{r=1}^{R} \rho_r^{(\ell)} \left(\left\langle \mathbf{u}_r^{(\ell)},\ \mathbf{u}_r^{(\ell)} \right\rangle - 1 \right) - \sum_{1 \le r_1 < r_2 \le R}\ \sum_{\ell=k-\mu+1}^{k} \alpha_{r_1 r_2}^{(\ell)} \left\langle \mathbf{u}_{r_1}^{(\ell)},\ \mathbf{u}_{r_2}^{(\ell)} \right\rangle,$$

where λ_r is given by (4.6) and $\rho_r^{(\ell)}, \alpha_{r_1 r_2}^{(\ell)}$ are Lagrange multipliers. According to [130], the first order optimality condition for a stationary point is to satisfy

$$\lambda_r T \circledast_\ell \left(\bigotimes_{i=1}^{\ell-1} \mathbf{u}_r^{(i)} \otimes \bigotimes_{i=\ell+1}^{k} \mathbf{u}_r^{(i)} \right) = \rho_r^{(\ell)} \mathbf{u}_r^{(\ell)}, \quad \ell = 1, \ldots, k-\mu, \quad r = 1, \ldots, R. \tag{4.10}$$

and

$$\lambda_r T \circledast_\ell \left(\bigotimes_{i=1}^{\ell-1} \mathbf{u}_r^{(i)} \otimes \bigotimes_{i=\ell+1}^{k} \mathbf{u}_r^{(i)} \right) = \rho_r^{(\ell)} \mathbf{u}_r^{(\ell)} + \sum_{r_1<r} \frac{\alpha_{r_1 r}^{(\ell)}}{2} \mathbf{u}_{r_1}^{(\ell)} + \sum_{r<r_2} \frac{\alpha_{r r_2}^{(\ell)}}{2} \mathbf{u}_{r_2}^{(\ell)}, \quad (4.11)$$
$$\ell = k-\mu+1,\ldots,k, r = 1,\ldots,R.$$

It follows from the orthogonality condition that

$$\rho_r^{(\ell)} = \lambda_r^2, \qquad \ell = 1,\ldots,k, \quad r = 1,\ldots,R,$$

and furthermore,

$$V^{(\ell)}\Lambda^{(\ell)} = U^{(\ell)}S^{(\ell)}, \ S^{(\ell)} \text{ is symmetric}, \quad \ell = k-\mu+1,\ldots,k, \qquad (4.12)$$

where

$$\mathbf{v}_r^{(\ell)} = T \circledast_\ell \left(\bigotimes_{i=1}^{\ell-1} \mathbf{u}_r^{(i)} \otimes \bigotimes_{i=\ell+1}^{k} \mathbf{u}_r^{(i)} \right), \quad \ell = k-\mu+1,\ldots,k, \quad r = 1,\ldots,R,$$

and

$$\begin{aligned} V^{(\ell)} &= \left[\mathbf{v}_1^{(\ell)}, \cdots, \mathbf{v}_R^{(\ell)} \right], \\ U^{(\ell)} &= \left[\mathbf{u}_1^{(\ell)}, \cdots, \mathbf{u}_R^{(\ell)} \right], \\ \Lambda^{(\ell)} &= \begin{bmatrix} \lambda_1^{(\ell)} & & \\ & \ddots & \\ & & \lambda_R^{(\ell)} \end{bmatrix}. \end{aligned}$$

It is from Lemma 4.3.1 and equation (4.12) that we can update $U^{(\ell)}$ by the orthogonal polar factor of the matrix $V^{(\ell)}\Lambda^{(\ell)}$: Let the polar decomposition of $V^{(\ell)}\Lambda^{(\ell)}$ be

$$V^{(\ell)}\Lambda^{(\ell)} = \tilde{U}^{(\ell)}\tilde{S}^{(\ell)}, \quad \ell = k-\mu+1,\ldots,k,$$

where $\tilde{U}^{(\ell)}$ is column orthogonal and $\tilde{S}^{(\ell)}$ is symmetric and positive semi-definite.

Denote

$$\tilde{\lambda}_r^{(\ell)} = \left\langle \mathbf{v}_r^{(\ell)},\ \tilde{\mathbf{u}}_r^{(\ell)} \right\rangle, \quad \ell = k-\mu+1,\ldots,k, \quad r = 1,\ldots,R.$$

We have

$$\sum_{r=1}^{R}(\lambda_r^{(\ell)})^2 = \text{Trace}\left((U^{(\ell)})^T V^{(\ell)}\Lambda^{(\ell)}\right) \leq \text{Trace}\left((\tilde{U}^{(\ell)})^T V^{(\ell)}\Lambda^{(\ell)}\right) = \sum_{r=1}^{R}\tilde{\lambda}_r^{(\ell)}\lambda_r^{(\ell)},$$
$$\ell = k-\mu+1,\ldots,k.$$

Consequently, we have by using the Cauchy-Schwarz inequality that

$$\sum_{r=1}^{R}(\lambda_r^{(\ell)})^2 \leq \sum_{r=1}^{R}(\tilde{\lambda}_r^{(\ell)})^2, \quad \ell = k-\mu+1,\ldots,k, \tag{4.13}$$

and the equality holds if and only if

$$\lambda_r^{(\ell)} = \tilde{\lambda}_r^{(\ell)}, \quad \ell = k-\mu+1,\ldots,k, \quad r = 1,\ldots,R.$$

Hence, we update $U^{(\ell)}$ by $\tilde{U}^{(\ell)}$ for $\ell = k-\mu+1,\ldots,k$.

To update two factors simultaneously as possible, we have to consider if $k-\mu$ is even or odd:

- For the case that $k-\mu$ is even, we can update $\mathbf{u}_r^{(\ell)}$ and $\mathbf{u}_r^{(\ell+1)}$ simultaneously by SVDs for $\ell = 1,3,\ldots,k-\mu-1$, $r = 1,\ldots,R$. Then we can update $U^{(\ell)}$ with $\ell = k-\mu+1$, …, k by polar decompositions given in (ii) above to ensure the orthogonality condition.

- For the case $k-\mu$ is odd, we can update $\mathbf{u}_r^{(\ell)}$ and $\mathbf{u}_r^{(\ell+1)}$ simultaneously by SVDs for $\ell = 1,3,\ldots,k-\mu-2$, $r = 1,\ldots,R$. Next update the new "$\mathbf{u}_r^{(k-\mu-1)}$" and $\mathbf{u}_r^{(k-\mu)}$ simultaneously by SVD (so, $\mathbf{u}_r^{(k-\mu-1)}$ will be updated one more time), $r = 1,\ldots,R$. Then we can update $U^{(\ell)}$ with $\ell = k-\mu+1$, …, k by polar decompositions given in (ii) above to ensure the orthogonality condition.

The idea above leads to our SVD-based algorithm named **Algorithm 7** for orthogonal low rank approximation of tensors.

Algorithm 7 (Orthogonal low rank approximation for tensors.)

Input: An order-k tensor $T \in \mathbb{R}^{I_1 \times \dots \times I_k}$ and starting unit vectors $\mathbf{u}^{(\ell)}_{r.[0]} \in \mathbb{R}^{I_\ell}, \ell = 1, \dots, k, r = 1, \dots, R$ and $\mathbf{u}^{(\ell)}_{i,[0]} \perp \mathbf{u}^{(\ell)}_{j,[0]}$ for $\ell = k - \mu + 1, \dots, k$, $1 \leq i \neq j \leq R$

Output: An orthogonal rank-R approximation

```
$T = \frac{1}{\|T\|_F} T$ {Normalize $T$}
$\tau := k - \mu - 1$
if $k - \mu$ is odd then
    $\tau := k - \mu - 2$
end if
for $p = 0, 1, \dots,$ do
    for $\ell = 1, 3, \dots, \tau$ do
        $\beta_\ell = (\ell, \ell + 1)$
        for $r = 1, 2, \dots, R,$ do
            $C^{(\ell)}_{r,[p+1]} = T \circledast_{\beta_\ell} \left( \bigotimes_{i=1}^{\ell-1} \mathbf{u}^{(i)}_{r,[p+1]} \otimes \bigotimes_{i=\ell+2}^{k} \mathbf{u}^{(i)}_{r,[p]} \right)$ {A matrix of size $I_\ell \times I_{\ell+1}$}
            $[\mathbf{u}, s, \mathbf{v}] = \mathsf{svds}(C^{(\ell)}_{r,[p+1]}, 1)$ {Dominant singular value triplet via Matlab routine svds;assume uniqueness}
            if $\mathbf{u}_1 < 0$ then
                $\mathbf{u} = -\mathbf{u}, \mathbf{v} = -\mathbf{v}$
            end if
            $\mathbf{u}^{(\ell)}_{r,[p+1]} := \mathbf{u}$
            $\mathbf{u}^{(\ell+1)}_{r,[p+1]} := \mathbf{v}$ {if $k - \mu$ is odd, use $\hat{\mathbf{u}}^{(k-\mu-1)}_{r,[p+1]} := \mathbf{v}$}
            $\lambda^{(\ell)}_{r,[p+1]} := s, \quad \lambda^{(\ell+1)}_{r,[p+1]} := s$ {if $k - \mu$ is odd, use $\hat{\lambda}^{(k-\mu-1)}_{r,[p+1]} := s$}
        end for
    end for
    if $\tau = k - \mu - 2$ then
        $\beta_{k-\mu-1} = (k - \mu - 1, \ k - \mu)$
        for $r = 1, 2, \dots, R,$ do
            $C^{(k-\mu-1)}_{r,[p+1]} = T \circledast_{\beta_{k-\mu-1}} \left( \bigotimes_{i=1}^{k-\mu-2} \mathbf{u}^{(i)}_{r,[p+1]} \otimes \bigotimes_{i=k-\mu+1}^{k} \mathbf{u}^{(i)}_{r,[p]} \right)$ {A matrix of size
            $I_{k-\mu-1} \times I_{k-\mu}\}$
            $[\mathbf{u}, s, \mathbf{v}] = \mathsf{svds}(C^{(k-\mu-1)}_{r,[p+1]}, 1)$ {Dominant singular value triplet via Matlab routine svds;assume uniqueness}
            if $\mathbf{u}_1 < 0$ then
                $\mathbf{u} = -\mathbf{u}, \mathbf{v} = -\mathbf{v}$
            end if
            $\mathbf{u}^{(k-\mu-1)}_{r,[p+1]} := \mathbf{u}, \quad \mathbf{u}^{(k-\mu)}_{r,[p+1]} := \mathbf{v}$
            $\lambda^{(k-\mu-1)}_{r,[p+1]} := s, \ \lambda^{(k-\mu)}_{r,[p+1]} := s$
        end for
    end if
    for $\ell = k - \mu + 1, \ldots, k$ do
        for $r = 1, 2, \ldots, R,$ do
            $\mathbf{v}^{(\ell)}_{r,[p+1]} = T \circledast_\ell \left( \bigotimes_{i=1}^{\ell-1} \mathbf{u}^{(i)}_{r,[p+1]} \otimes \bigotimes_{i=\ell+1}^{k} \mathbf{u}^{(i)}_{r,[p]} \right)$ {define columns of $V^{(\ell)}_{[p+1]}$}
            $\hat{\lambda}^{(\ell)}_{r,[p+1]} := \langle \mathbf{v}^{(\ell)}_{r,[p+1]}, \mathbf{u}^{(\ell)}_{r,[p]} \rangle$ {define diagonals of $\Lambda^{(\ell)}_{[p+1]}$}
        end for
        $[U^{(\ell)}_{[p+1]}, S^{(\ell)}_{[p+1]}] = \mathsf{poldec}(V^{(\ell)}_{[p+1]} \Lambda^{(\ell)}_{[p+1]})$
        for $r = 1, 2, \ldots, R,$ do
            $\mathbf{u}^{(\ell)}_{r,[p+1]} := U^{(\ell)}_{[p+1]}(:, r)$
            $\lambda^{(\ell)}_{r,[p+1]} := S^{(\ell)}_{[p+1]}(r, r) (= \langle \mathbf{v}^{(\ell)}_{r,[p+1]}, \mathbf{u}^{(\ell)}_{r,[p+1]} \rangle)$
        end for
    end for
end for
```

4.5 Convergence

In this section we prove the convergence of objective function (4.4) first and then the convergence of iterates.

4.5.1 Convergence of objective function

As discussed in (i) of Section 3, Algorithm 7 ensures an inherent ascending property on $\lambda_{r,[p]} := \lambda_r \left(\mathbf{u}_{r,[p]}^{(1)}, \ldots, \mathbf{u}_{r,[p]}^{(k)}\right)$ in the sense that

$$\lambda_{r,[p]} \leq \lambda_{r,[p+1]}^{(1)} = \lambda_{r,[p+1]}^{(2)} \leq \lambda_{r,[p+1]}^{(3)} = \lambda_{r,[p+1]}^{(4)} \leq \ldots \leq \lambda_{r,[p+1]}^{(k-\mu-1)} = \lambda_{r,[p+1]}^{(k-\mu)}, \quad r = 1, \ldots, R.$$

Therefore, we have

$$\sum_{r=1}^{R} (\lambda_{r,[p]})^2 \leq \sum_{r=1}^{R} (\lambda_{r,[p+1]}^{(1)})^2 \leq \ldots \leq \sum_{r=1}^{R} (\lambda_{r,[p+1]}^{(k-\mu)})^2.$$

Then according to the discussion in (ii) of Section 3, we also have

$$\sum_{r=1}^{R} (\lambda_{r,[p+1]}^{(k-\mu)})^2 \leq \sum_{r=1}^{R} \lambda_{r,[p+1]}^{(k-\mu)} \lambda_{r,[p+1]}^{(k-\mu+1)} \leq \cdots \leq \sum_{r=1}^{R} \lambda_{r,[p+1]}^{(k-1)} \lambda_{r,[p+1]}^{(k)} \leq \sum_{r=1}^{R} (\lambda_{r,[p+1]}^{(k)})^2 = \sum_{r=1}^{R} (\lambda_{r,[p+1]})^2.$$

Therefore, the sequence $\{\sum_{r=1}^{R} (\lambda_{r,[p]})^2\}$ is increasing. Since $\sum_{r=1}^{R} (\lambda_{r,[p]})^2 \leq \|T\|_F^2$ for any $p = 1, 2, \cdots$, the following convergence result is ready.

Theorem 4.5.1. *Let the sequence $\{\lambda_{1,[p]}, \ldots, \lambda_{R,[p]}\}$ be generated in Algorithm 7. Then the objective value $\sum_{r=1}^{R} (\lambda_{r,[p]})^2$ converges.*

4.5.2 Convergence of iterates

Before proving the global convergence of the iterates $\{\mathbf{u}_{r,[p]}^{(\ell)}\}$, we show that generically accumulation points of Algorithm 7 are geometrically isolated. Denote the accumulation point generated by Algorithm 7 as

$$\{(\mathbf{u}_1^{(1)}, \ldots, \mathbf{u}_1^{(k)}), \ldots, (\mathbf{u}_R^{(1)}, \ldots, \mathbf{u}_R^{(k)})\}. \tag{4.14}$$

Then by equations (4.10) and (4.11), any accumulation point of Algorithm 7 necessarily satisfies the system of nonlinear equations, let $\beta_\ell = (\ell, \ell+1)$,

$$\begin{cases} T \circledast_{\beta_\ell} \left(\bigotimes_{i=1}^{\ell-1} \mathbf{u}_r^{(i)} \otimes \bigotimes_{i=\ell+2}^{k} \mathbf{u}_r^{(i)} \right) \mathbf{u}_r^{(\ell+1)} = \left\langle T,\ \bigotimes_{i=1}^{k} \mathbf{u}_r^{(i)} \right\rangle \mathbf{u}_r^{(\ell)}, \\ \qquad\qquad\qquad\qquad \ell = 1, \ldots, k-\mu-1, \quad r = 1, \ldots, R, \\ T \circledast_{\ell} \left(\bigotimes_{i=1}^{\ell-1} \mathbf{u}_r^{(i)} \otimes \bigotimes_{i=\ell+1}^{k} \mathbf{u}_r^{(i)} \right) = \sum_{t=1}^{R} \left\langle T, \bigotimes_{i=1}^{\ell-1} \mathbf{u}_r^{(i)} \otimes \mathbf{u}_t^{(\ell)} \otimes \bigotimes_{i=\ell+1}^{k} \mathbf{u}_r^{(i)} \right\rangle \mathbf{u}_t^{(\ell)}, \\ \qquad\qquad\qquad\qquad \ell = k-\mu+1, \ldots, k, \quad r = 1, \ldots, R. \end{cases} \tag{4.15}$$

This is exactly the same as the accumulation point of ALS method proposed in [190] if $\mu = 1$. Furthermore, this system is the same as the one characterizing the first order optimality condition for a stationary point of the optimization problem (4.7) (i.e., (4.4)). As a result, the following result in [190] also holds for Algorithm 7.

Lemma 4.5.1. *[190] For almost all tensors $T \in \mathbb{R}^{I_1\times\cdots\times I_k}$, the accumulation points of any sequence generated by Algorithm 7 and the stationary points of the optimization problem (4.7) (i.e., (4.4)) are necessarily isolated.*

Assumption A. We say that a given tensor $T \in \mathbb{R}^{I_1\times\cdots\times I_k}$ satisfies Assumption A if for every convergent subsequence $\left\{\mathbf{u}_{r,[p_j]}^{(\ell)}\right\}$ generated by Algorithm 7, the dominant singular value of the limiting point $C_r^{(\ell)}$ of the corresponding subsequence $\left\{C_{r,[p_j]}^{(\ell)}\right\}$ are simple for all $\ell = 1,\ldots,k-\mu$, $r = 1,\ldots,R$. Moreover, the limiting point $V^{(\ell)}\Lambda^{(\ell)}$ of the matrix $V_{[p_j]}^{(\ell)}\Lambda_{[p_j]}^{(\ell)}$ for $\ell = k-\mu+1,\ldots,k$ are of full column rank.

Remark: It follows from Lemma 4.5.1, a stronger assumption which is independent of Algorithm 7 is as follows:

Assumption $\tilde{\mathrm{A}}$. We say that a given tensor $T \in \mathbb{R}^{I_1\times\cdots\times I_k}$ satisfies Assumption $\tilde{\mathrm{A}}$ if for every stationary points $\left\{\mathbf{u}_r^{(\ell)}\right\}$ of the optimization problem (4.7), the dominant singular value of corresponding $C_r^{(\ell)}$ are simple for all $\ell = 1,\ldots,k-\mu$, $r = 1,\ldots,R$. Moreover, the corresponding $V^{(\ell)}\Lambda^{(\ell)}$ for $\ell = k-\mu+1,\ldots,k$ are of full column rank.

Assumption A is to ensure the uniqueness of singular value decompositions and polar decompositions in Algorithm 7. Since vectors $\mathbf{u}_{r,[p]}^{(\ell)}$ defined in Algorithm 7 are unit vectors, the sequence $\left\{\mathbf{u}_{r,[p]}^{(\ell)}\right\}$ must have a convergent subsequence. Selecting the common subset $\{r, [p_j]\}$ of nonnegative integers so that $\left\{\mathbf{u}_{r,[p_j]}^{(\ell)}\right\}$ converges simultaneously for all $\ell = 1,\ldots,k$, $r = 1,\ldots,R$.

Lemma 4.5.2. *For all $\ell = 1,\ldots,k$, $r = 1,\ldots,R$, if subsequences $\left\{\mathbf{u}_{r,[p_j]}^{(\ell)}\right\}$ generated by Algorithm 7 converge simultaneously, then subsequences $\left\{\mathbf{u}_{r,[p_j+1]}^{(\ell)}\right\}$ also converge simultaneously. Furthermore, under Assumption A, $\left\{\mathbf{u}_{r,[p_j]}^{(\ell)}\right\}$ and $\left\{\mathbf{u}_{r,[p_j+1]}^{(\ell)}\right\}$ converge to the same limiting point for $\ell = 1,\ldots,k$, $r = 1,\ldots,R$.*

Proof. The simultaneous convergence of $\left\{\mathbf{u}_{r,[p_j]}^{(\ell)}\right\}$ for $\ell = 3, \ldots k$, $r = 1, \ldots, R$ implies that the subsequence $\left\{C_{r,[p_j+1]}^{(1)}\right\}$ converges. By the continuity inherited in the SVD, $\left\{\mathbf{u}_{r,[p_j+1]}^{(1)}\right\}$ and $\left\{\mathbf{u}_{r,[p_j+1]}^{(2)}\right\}$ converge since we have aligned the direction of dominate left and right singular vector in Algorithm 7. This results in that $\left\{C_{r,[p_j+1]}^{(3)}\right\}$ converges and so do $\left\{\mathbf{u}_{r,[p_j+1]}^{(3)}\right\}$ and $\left\{\mathbf{u}_{r,[p_j+1]}^{(4)}\right\}$. Then the simultaneous convergence of $\left\{\mathbf{u}_{r,[p_j]}^{(\ell+2)}\right\}, \ldots, \left\{\mathbf{u}_{r,[p_j]}^{(k)}\right\}$, and $\left\{\mathbf{u}_{r,[p_j+1]}^{(1)}\right\}, \ldots, \left\{\mathbf{u}_{r,[p_j+1]}^{(\ell-1)}\right\}$ ensures that $\left\{C_{r,[p_j+1]}^{(\ell)}\right\}$ converges and consequently $\left\{\mathbf{u}_{r,[p_j+1]}^{(\ell)}\right\}$ and $\left\{\mathbf{u}_{r,[p_j+1]}^{(\ell+1)}\right\}$ converge for $\ell = 1, 3, \ldots, k-\mu-1$, $r = 1, \ldots, R$. One thing should be mentioned that there are additional convergent $\left\{\hat{\mathbf{u}}_{r,[p_j+1]}^{(k-\mu-1)}\right\}$ if $k-\mu$ is odd. Furthermore, $V_{[p_j+1]}^{(\ell)}\Lambda_{[p_j+1]}^{(\ell)}$ converges by definition and thus by continuity of the polar decomposition, $\left\{\mathbf{u}_{r,[p_j+1]}^{(\ell)}\right\}$ converges for $\ell = k-\mu+1, \ldots, k$, $r = 1, \ldots, R$.

Let the limiting points of $\left\{\mathbf{u}_{r,[p_j]}^{(\ell)}\right\}$, $\left\{\mathbf{u}_{r,[p_j+1]}^{(\ell)}\right\}$ and $\left\{C_{r,[p_j+1]}^{(\ell)}\right\}$ be $\mathbf{u}_r^{(\ell)}$, $\tilde{\mathbf{u}}_r^{(\ell)}$ and $\tilde{C}_r^{(\ell)}$, respectively. In addition, use $\hat{\mathbf{u}}_r^{(k-\mu-1)}$ to denote the limiting point of $\left\{\hat{\mathbf{u}}_{r,[p_j+1]}^{(k-\mu-1)}\right\}$ if $k-\mu$ is odd. Now we prove that $\mathbf{u}_r^{(\ell)} = \tilde{\mathbf{u}}_r^{(\ell)}$ for $\ell = 1, \ldots, k$, $r = 1, \ldots, R$.

First we have for $r = 1, \ldots, R$,

$$\begin{aligned}
& \lambda_{r,[p_j]} = \left\langle \mathbf{u}_{r,[p_j]}^{(1)},\ C_{r,[p_j+1]}^{(1)}\mathbf{u}_{r,[p_j]}^{(2)} \right\rangle \\
\leq\ & \lambda_{r,[p_j+1]}^{(1)} = \langle \mathbf{u}_{r,[p_j+1]}^{(1)},\ C_{r,[p_j+1]}^{(1)}\mathbf{u}_{r,[p_j+1]}^{(2)}\rangle = \langle \mathbf{u}_{r,[p_j]}^{(3)},\ C_{r,[p_j+1]}^{(3)}\mathbf{u}_{r,[p_j]}^{(4)}\rangle = \lambda_{r,[p_j+1]}^{(2)} \\
\leq\ & \lambda_{r,[p_j+1]}^{(3)} = \langle \mathbf{u}_{r,[p_j+1]}^{(3)},\ C_{r,[p_j+1]}^{(3)}\mathbf{u}_{r,[p_j+1]}^{(4)}\rangle = \langle \mathbf{u}_{r,[p_j]}^{(5)},\ C_{r,[p_j+1]}^{(5)}\mathbf{u}_{r,[p_j]}^{(6)}\rangle = \lambda_{r,[p_j+1]}^{(4)} \\
\leq\ & \cdots \\
\leq\ & \lambda_{r,[p_j+1]}^{(\ell)} = \langle \mathbf{u}_{r,[p_j+1]}^{(\ell)},\ C_{r,[p_j+1]}^{(\ell)}\mathbf{u}_{r,[p_j+1]}^{(\ell+1)}\rangle = \langle \mathbf{u}_{r,[p_j]}^{(\ell+2)},\ C_{r,[p_j+1]}^{(\ell+2)}\mathbf{u}_{r,[p_j]}^{(\ell+3)}\rangle = \lambda_{r,[p_j+1]}^{(\ell+1)} \quad (\ell = 1, 3, 5, \ldots) \\
\leq\ & \cdots \\
\leq\ & \begin{cases}
\lambda_{r,[p_j+1]}^{(k-\mu-5)} = \langle \mathbf{u}_{r,[p_j+1]}^{(k-\mu-5)},\ C_{r,[p_j+1]}^{(k-\mu-5)}\mathbf{u}_{r,[p_j+1]}^{(k-\mu-4)}\rangle = \langle \mathbf{u}_{r,[p_j]}^{(k-\mu-3)},\ C_{r,[p_j+1]}^{(k-\mu-3)}\mathbf{u}_{r,[p_j]}^{(k-\mu-2)}\rangle \\
\qquad = \lambda_{r,[p_j+1]}^{(k-\mu-4)} \qquad \text{if } k-\mu \text{ is even} \\
\lambda_{r,[p_j+1]}^{(k-\mu-4)} = \langle \mathbf{u}_{r,[p_j+1]}^{(k-\mu-4)},\ C_{r,[p_j+1]}^{(k-\mu-4)}\mathbf{u}_{r,[p_j+1]}^{(k-\mu-3)}\rangle = \langle \mathbf{u}_{r,[p_j]}^{(k-\mu-2)},\ C_{r,[p_j+1]}^{(k-\mu-2)}\mathbf{u}_{r,[p_j]}^{(k-\mu-1)}\rangle \\
\qquad = \lambda_{r,[p_j+1]}^{(k-\mu-3)} \qquad \text{if } k-\mu \text{ is odd}
\end{cases}
\end{aligned}$$

$$
\begin{aligned}
\leq\ & \begin{cases}
\lambda^{(k-\mu-3)}_{r,[p_j+1]} = \langle \mathbf{u}^{(k-\mu-3)}_{r,[p_j+1]},\ C^{(k-\mu-3)}_{r,[p_j+1]}\mathbf{u}^{(k-\mu-2)}_{r,[p_j+1]}\rangle = \langle \mathbf{u}^{(k-\mu-1)}_{r,[p_j]},\ C^{(k-\mu-1)}_{r,[p_j+1]}\mathbf{u}^{(k-\mu)}_{r,[p_j]}\rangle \\
\qquad = \lambda^{(k-\mu-2)}_{r,[p_j+1]} \qquad \text{if } k-\mu \text{ is even} \\
\lambda^{(k-\mu-2)}_{r,[p_j+1]} = \langle \mathbf{u}^{(k-\mu-2)}_{r,[p_j+1]},\ C^{(k-\mu-2)}_{r,[p_j+1]}\hat{\mathbf{u}}^{(k-\mu-1)}_{r,[p_j+1]}\rangle = \langle \hat{\mathbf{u}}^{(k-\mu-1)}_{r,[p_j+1]},\ C^{(k-\mu-1)}_{r,[p_j+1]}\mathbf{u}^{(k-\mu)}_{r,[p_j]}\rangle \\
\qquad = \hat{\lambda}^{(k-\mu-1)}_{r,[p_j+1]} \qquad \text{if } k-\mu \text{ is odd}
\end{cases} \\
\leq\ & \lambda^{(k-\mu-1)}_{r,[p_j+1]} = \langle \mathbf{u}^{(k-\mu-1)}_{r,[p_j+1]},\ C^{(k-\mu-1)}_{r,[p_j+1]}\mathbf{u}^{(k-\mu)}_{r,[p_j+1]}\rangle = \lambda^{(k-\mu)}_{r,[p_j+1]},
\end{aligned}
$$

then we have by taking the limits that

$$
\begin{aligned}
& \lambda_r = \langle \mathbf{u}^{(1)}_r,\ \tilde{C}^{(1)}_r\mathbf{u}^{(2)}_r\rangle \\
\leq\ & \tilde{\lambda}^{(1)}_r = \langle \tilde{\mathbf{u}}^{(1)}_r,\ \tilde{C}^{(1)}_r\tilde{\mathbf{u}}^{(2)}_r\rangle = \langle \mathbf{u}^{(3)}_r,\ \tilde{C}^{(3)}_r\mathbf{u}^{(4)}_r\rangle = \tilde{\lambda}^{(2)}_r \\
\leq\ & \tilde{\lambda}^{(3)}_r = \langle \tilde{\mathbf{u}}^{(3)}_r,\ \tilde{C}^{(3)}_r\tilde{\mathbf{u}}^{(4)}_r\rangle = \langle \mathbf{u}^{(5)}_r,\ \tilde{C}^{(5)}_r\mathbf{u}^{(6)}_r\rangle = \tilde{\lambda}^{(4)}_r \\
\leq\ & \cdots \\
\leq\ & \tilde{\lambda}^{(\ell)}_r = \langle \tilde{\mathbf{u}}^{(\ell)}_r,\ \tilde{C}^{(\ell)}_r\tilde{\mathbf{u}}^{(\ell+1)}_r\rangle = \langle \mathbf{u}^{(\ell+2)}_r,\ \tilde{C}^{(\ell+2)}_r\mathbf{u}^{(\ell+3)}_r\rangle = \tilde{\lambda}^{(\ell+1)}_r \quad (\ell = 1, 3, 5, \ldots) \\
\leq\ & \cdots \\
\leq\ & \begin{cases}
\tilde{\lambda}^{(k-\mu-5)}_r = \langle \tilde{\mathbf{u}}^{(k-\mu-5)}_r,\ \tilde{C}^{(k-\mu-5)}_r\tilde{\mathbf{u}}^{(k-\mu-4)}_r\rangle = \langle \mathbf{u}^{(k-\mu-3)}_r,\ \tilde{C}^{(k-\mu-3)}_r\mathbf{u}^{(k-\mu-2)}_r\rangle \\
\qquad = \tilde{\lambda}^{(k-\mu-4)}_r \qquad \text{if } k-\mu \text{ is even} \\
\tilde{\lambda}^{(k-\mu-4)}_r = \langle \tilde{\mathbf{u}}^{(k-\mu-4)}_r,\ \tilde{C}^{(k-\mu-4)}_r\tilde{\mathbf{u}}^{(k-\mu-3)}_r\rangle = \langle \mathbf{u}^{(k-\mu-2)}_r,\ \tilde{C}^{(k-\mu-2)}_r\mathbf{u}^{(k-\mu-1)}_r\rangle \\
\qquad = \tilde{\lambda}^{(k-\mu-3)}_r \qquad \text{if } k-\mu \text{ is odd}
\end{cases} \\
\leq\ & \begin{cases}
\tilde{\lambda}^{(k-\mu-3)}_r = \langle \tilde{\mathbf{u}}^{(k-\mu-3)}_r,\ \tilde{C}^{(k-\mu-3)}_r\tilde{\mathbf{u}}^{(k-\mu-2)}_r\rangle = \langle \mathbf{u}^{(k-\mu-1)}_r,\ \tilde{C}^{(k-\mu-1)}_r\mathbf{u}^{(k-\mu)}_r\rangle \\
\qquad = \tilde{\lambda}^{(k-\mu-2)}_r \qquad \text{if } k-\mu \text{ is even} \\
\tilde{\lambda}^{(k-\mu-2)}_r = \langle \tilde{\mathbf{u}}^{(k-\mu-2)}_r,\ \tilde{C}^{(k-\mu-2)}_r\hat{\mathbf{u}}^{(k-\mu-1)}_r\rangle = \langle \hat{\mathbf{u}}^{(k-\mu-1)}_r,\ \tilde{C}^{(k-\mu-1)}_r\mathbf{u}^{(k-\mu)}_r\rangle \\
\qquad = \hat{\lambda}^{(k-\mu-1)}_r \qquad \text{if } k-\mu \text{ is odd}
\end{cases} \\
\leq\ & \tilde{\lambda}^{(k-\mu-1)}_r = \langle \tilde{\mathbf{u}}^{(k-\mu-1)}_r,\ \tilde{C}^{(k-\mu-1)}_r\tilde{\mathbf{u}}^{(k-\mu)}_r\rangle = \tilde{\lambda}^{(k-\mu)}_r.
\end{aligned}
$$

Combined with discussion in (ii) of Section 4.3,

$$
\begin{aligned}
& \sum_{r=1}^{R}(\lambda_{r,[p_j]})^2 \leq \sum_{r=1}^{R}(\lambda^{(1)}_{r,[p_j+1]})^2 = \sum_{r=1}^{R}(\lambda^{(2)}_{r,[p_j+1]})^2 \\
\leq\ & \cdots \\
\leq\ & \sum_{r=1}^{R}(\lambda^{(\ell)}_{r,[p_j+1]})^2 = \sum_{r=1}^{R}(\lambda^{(\ell+1)}_{r,[p_j+1]})^2 \quad (\ell = 1, 3, 5, \ldots)
\end{aligned}
$$

$$
\begin{aligned}
&\leq \;\ldots \\
&\leq \begin{cases} \sum_{r=1}^{R}(\lambda_{r,[p_j+1]}^{(k-\mu-3)})^2 = \sum_{r=1}^{R}(\lambda_{r,[p_j+1]}^{(k-\mu-2)})^2 & \text{if } k-\mu \text{ is even} \\ \sum_{r=1}^{R}(\lambda_{r,[p_j+1]}^{(k-\mu-2)})^2 = \sum_{r=1}^{R}(\hat{\lambda}_{r,[p_j+1]}^{(k-\mu-1)})^2 & \text{if } k-\mu \text{ is odd} \end{cases} \\
&\leq \sum_{r=1}^{R}(\lambda_{r,[p_j+1]}^{(k-\mu-1)})^2 = \sum_{r=1}^{R}(\lambda_{r,[p_j+1]}^{(k-\mu)})^2 \leq \sum_{r=1}^{R}\lambda_{r,[p_j+1]}^{(k-\mu)}\lambda_{r,[p_j+1]}^{(k-\mu+1)} \\
&\leq \sum_{r=1}^{R}(\lambda_{r,[p_j+1]}^{(k-\mu+1)})^2 \leq \cdots \leq \sum_{r=1}^{R}(\lambda_{r,[p_j+1]}^{(k)})^2 = \sum_{r=1}^{R}(\lambda_{r,[p_j+1]})^2.
\end{aligned}
$$

So, it follows that

$$
\begin{aligned}
&\sum_{r=1}^{R}(\lambda_r)^2 \leq \sum_{r=1}^{R}(\tilde{\lambda}_r^{(1)})^2 = \sum_{r=1}^{R}(\tilde{\lambda}_r^{(2)})^2 \\
&\leq \;\ldots \\
&\leq \sum_{r=1}^{R}(\tilde{\lambda}_r^{(\ell)})^2 = \sum_{r=1}^{R}(\tilde{\lambda}_r^{(\ell+1)})^2 \quad (\ell = 1, 3, 5, \ldots) \\
&\leq \;\ldots \\
&\leq \begin{cases} \sum_{r=1}^{R}(\tilde{\lambda}_r^{(k-\mu-3)})^2 = \sum_{r=1}^{R}(\tilde{\lambda}_r^{(k-\mu-2)})^2 & \text{if } k-\mu \text{ is even} \\ \sum_{r=1}^{R}(\tilde{\lambda}_r^{(k-\mu-2)})^2 = \sum_{r=1}^{R}(\hat{\lambda}_r^{(k-\mu-1)})^2 & \text{if } k-\mu \text{ is odd} \end{cases} \\
&\leq \sum_{r=1}^{R}(\tilde{\lambda}_r^{(k-\mu-1)})^2 = \sum_{r=1}^{R}(\tilde{\lambda}_r^{(k-\mu)})^2 \leq \sum_{r=1}^{R}\tilde{\lambda}_r^{(k-\mu)}\tilde{\lambda}_r^{(k-\mu+1)} \\
&\leq \sum_{r=1}^{R}(\tilde{\lambda}_r^{(k-\mu+1)})^2 \leq \cdots \leq \sum_{r=1}^{R}(\tilde{\lambda}_r^{(k)})^2 = \sum_{r=1}^{R}(\tilde{\lambda}_r)^2.
\end{aligned}
$$

In addition, we also know by Theorem 4.5.1,

$$\sum_{r=1}^{R}(\lambda_r)^2 = \sum_{r=1}^{R}(\tilde{\lambda}_r)^2.$$

Hence, we obtain for $r = 1, \ldots, R$,

$$
\begin{cases} \lambda_r = \tilde{\lambda}_r^{(1)} = \cdots = \tilde{\lambda}_r^{(k-\mu-1)} = \tilde{\lambda}_r^{(k-\mu)}, & \text{if } k-\mu \text{ is even} \\ \lambda_r = \tilde{\lambda}_r^{(1)} = \cdots = \tilde{\lambda}_r^{(k-\mu-1)} = \hat{\lambda}_r^{(k-\mu-1)} = \tilde{\lambda}_r^{(k-\mu)}, & \text{if } k-\mu \text{ is odd} \end{cases}
$$

which, in return, gives

$$\tilde{\lambda}_r^{(1)} = \langle \tilde{\mathbf{u}}_r^{(1)},\ \tilde{C}_r^{(1)}\tilde{\mathbf{u}}_r^{(2)}\rangle = \langle \mathbf{u}_r^{(1)},\ \tilde{C}_r^{(1)}\mathbf{u}_r^{(2)}\rangle = \lambda_r, \tag{4.16}$$

$$\tilde{\lambda}_r^{(3)} = \langle \tilde{\mathbf{u}}_r^{(3)},\ \tilde{C}_r^{(3)} \tilde{\mathbf{u}}_r^{(4)} \rangle = \langle \mathbf{u}_r^{(3)},\ \tilde{C}_r^{(3)} \mathbf{u}_r^{(4)} \rangle = \tilde{\lambda}_r^{(2)}, \tag{4.17}$$

$$\vdots$$

$$\tilde{\lambda}_r^{(\ell)} = \langle \tilde{\mathbf{u}}_r^{(\ell)},\ \tilde{C}_r^{(\ell)} \tilde{\mathbf{u}}_r^{(\ell+1)} \rangle = \langle \mathbf{u}_r^{(\ell)},\ \tilde{C}_r^{(\ell)} \mathbf{u}_r^{(\ell+1)} \rangle = \tilde{\lambda}_r^{(\ell-1)} \quad (\ell = 1, 3, 5, \ldots) \tag{4.18}$$

$$\vdots$$

$$\begin{cases} \tilde{\lambda}_r^{(k-\mu-3)} = \langle \tilde{\mathbf{u}}_r^{(k-\mu-3)},\ \tilde{C}_r^{(k-\mu-3)} \tilde{\mathbf{u}}_r^{(k-\mu-2)} \rangle = \langle \mathbf{u}_r^{(k-\mu-3)},\ \tilde{C}_r^{(k-\mu-3)} \mathbf{u}_r^{(k-\mu-2)} \rangle \\ \qquad = \tilde{\lambda}_r^{(k-\mu-4)}, \qquad \text{if } k-\mu \text{ is even,} \\ \tilde{\lambda}_r^{(k-\mu-2)} = \langle \tilde{\mathbf{u}}_r^{(k-\mu-2)}, \tilde{C}_r^{(k-\mu-2)} \hat{\mathbf{u}}_r^{(k-\mu-1)} \rangle = \langle \mathbf{u}_r^{(k-\mu-2)},\ \tilde{C}_r^{(k-\mu-2)} \mathbf{u}_r^{(k-\mu-1)} \rangle \\ \qquad = \tilde{\lambda}_r^{(k-\mu-3)}, \qquad \text{if } k-\mu \text{ is odd,} \end{cases} \tag{4.19}$$

$$\begin{cases} \tilde{\lambda}_r^{(k-\mu-1)} = \langle \tilde{\mathbf{u}}_r^{(k-\mu-1)},\ \tilde{C}_r^{(k-\mu-1)} \tilde{\mathbf{u}}_r^{(k-\mu)} \rangle = \langle \mathbf{u}_r^{(k-\mu-1)},\ \tilde{C}_r^{(k-\mu-1)} \mathbf{u}_r^{(k-\mu)} \rangle \\ \qquad = \tilde{\lambda}_r^{(k-\mu-2)}, \qquad \text{if } k-\mu \text{ is even,} \\ \tilde{\lambda}_r^{(k-\mu-1)} = \langle \tilde{\mathbf{u}}_r^{(k-\mu-1)},\ \tilde{C}_r^{(k-\mu-1)} \tilde{\mathbf{u}}_r^{(k-\mu)} \rangle = \langle \hat{\mathbf{u}}_r^{(k-\mu-1)},\ \tilde{C}_r^{(k-\mu-1)} \mathbf{u}_r^{(k-\mu)} \rangle \\ \qquad = \hat{\lambda}_r^{(k-\mu-1)}, \qquad \text{if } k-\mu \text{ is odd.} \end{cases} \tag{4.20}$$

Since $\lambda_{r,[p_j+1]}^{(1)}$, $\lambda_{r,[p_j+1]}^{(3)}$, …, $\lambda_{r,[p_j+1]}^{(k-\mu-1)}$ are the largest singular values of $C_{r,[p_j+1]}^{(1)}$, $C_{r,[p_j+1]}^{(3)}$, …, $\lambda_{r,[p_j+1]}^{(k-\mu-1)}$ respectively, thus, $\tilde{\lambda}_r^{(1)}$, $\tilde{\lambda}_r^{(3)}$, …, $\tilde{\lambda}_r^{(k-\mu-1)}$ are the largest singular values of $\tilde{C}_r^{(1)}$, $\tilde{C}_r^{(3)}$, …, $\tilde{C}_r^{(k-\mu-1)}$ respectively.

Furthermore, under Assumption A, the left and right singular vectors of $\tilde{C}_r^{(1)}$, …, $\tilde{C}_r^{(k-\mu-1)}$ are unique. Therefore, we have for $r = 1, \ldots, R$,

$$\tilde{\mathbf{u}}_r^{(1)} = \mathbf{u}_r^{(1)},\ \tilde{\mathbf{u}}_r^{(2)} = \mathbf{u}_r^{(2)}, \quad \text{(by (4.16))}$$

$$\tilde{\mathbf{u}}_r^{(3)} = \mathbf{u}_r^{(3)},\ \tilde{\mathbf{u}}_r^{(4)} = \mathbf{u}_r^{(4)}, \quad \text{(by (4.17))}$$

$$\vdots$$

$$\tilde{\mathbf{u}}_r^{(\ell)} = \mathbf{u}_r^{(\ell)},\ \tilde{\mathbf{u}}_r^{(\ell+1)} = \mathbf{u}_r^{(\ell+1)}\ (\ell = 1, 3, 5, \ldots) \quad \text{(by (4.18))}$$

$$\vdots$$

$$\begin{cases} \tilde{\mathbf{u}}_r^{(k-\mu-3)} = \mathbf{u}_r^{(k-\mu-3)}, \tilde{\mathbf{u}}_r^{(k-\mu-2)} = \mathbf{u}_r^{(k-\mu-2)}, \quad \text{if } k-\mu \text{ is even,} \\ \tilde{\mathbf{u}}_r^{(k-\mu-2)} = \mathbf{u}_r^{(k-\mu-2)}, \hat{\mathbf{u}}_r^{(k-\mu-1)} = \mathbf{u}_r^{(k-\mu-1)}, \quad \text{if } k-\mu \text{ is odd,} \end{cases} \quad \text{(by (4.19))}$$

$$\begin{cases} \tilde{\mathbf{u}}_r^{(k-\mu-1)} = \mathbf{u}_r^{(k-\mu-1)}, \tilde{\mathbf{u}}_r^{(k-\mu)} = \mathbf{u}_r^{(k-\mu)}, \quad \text{if } k-\mu \text{ is even,} \\ \tilde{\mathbf{u}}_r^{(k-\mu-1)} = \hat{\mathbf{u}}_r^{(k-\mu-1)}, \tilde{\mathbf{u}}_r^{(k-\mu)} = \mathbf{u}_r^{(k-\mu)}, \quad \text{if } k-\mu \text{ is odd,} \end{cases} \quad \text{(by (4.20))}$$

that is $\mathbf{u}_r^{(\ell)} = \tilde{\mathbf{u}}_r^{(\ell)}$ hold for $\ell = 1, \ldots, k - \mu$, $r = 1, \ldots, R$. Finally, since

$$\tilde{V}^{(\ell)} = \lim V_{[p_j+1]}^{(\ell)} = \lim V_{[p_j]}^{(\ell)} = V^{(\ell)},\ \tilde{\Lambda}^{(\ell)} = \lim \Lambda_{[p_j+1]}^{(\ell)} = \lim \Lambda_{[p_j]}^{(\ell)} = \Lambda^{(\ell)},$$

we have with Assumption A that

$$\tilde{U}^{(\ell)} = \lim U_{[p_j+1]}^{(\ell)} = \lim U_{[p_j]}^{(\ell)} = U^{(\ell)}, \quad \forall \ell = k - \mu + 1, \ldots, k.$$

□

Now we are ready to prove the convergence of the whole sequence $\left\{\mathbf{u}_{r,[p]}^{(\ell)}\right\}$ for $\ell = 1, \ldots, k$, $r = 1, \ldots, R$.

Theorem 4.5.2. *For almost all tensors T satisfying Assumption A, the sequence $\left\{\mathbf{u}_{r,[p]}^{(\ell)}\right\}$ generated in Algorithm 7 converges for $\ell = 1, \ldots, k$, $r = 1, \ldots, R$.*

Proof. Suppose that $\left\{\mathbf{u}_{r,[p_j]}^{(\ell)}\right\}$ is any subsequence converging to a limiting point $\mathbf{u}_r^{(\ell)}$ for $\ell = 1, \ldots, k$, $r = 1, \ldots, R$. By Lemma 4.5.1, $\mathbf{u}_r^{(\ell)}$ is isolated, and by Lemma 4.5.2, the subsequence $\left\{\mathbf{u}_{r,[p_j+1]}^{(\ell)}\right\}$ also converges to $\mathbf{u}_r^{(\ell)}$ and $\left\|\mathbf{u}_{r,[p_j+1]}^{(\ell)} - \mathbf{u}_{r,[p_j]}^{(\ell)}\right\| \to 0$ for $\ell = 1, \ldots, k$, $r = 1, \ldots, R$. It follows from Lemma 4.3.3 that the whole sequence $\left\{\mathbf{u}_{r,[p_j+1]}^{(\ell)}\right\}$ converges to $\mathbf{u}_r^{(\ell)}$ for $\ell = 1, \ldots, k$, $r = 1, \ldots, R$. □

4.6 Numerical experiments

In this section, we present numerical experiments to illustrate the convergence of Algorithm 7 by measuring

- objective value $\sum_{r=1}^{R} \lambda_r^2$;
- iterate error $\sum_{\ell=1}^{k} \sum_{r=1}^{R} \|\mathbf{u}_{r,[p+1]}^{(\ell)} - \mathbf{u}_{r,[p]}^{(\ell)}\|_2^2$.

We experiment with Random tensor, Stochastic tensor, Cauchy tensor, Hilbert tensor, and Toeplitz tensor, with the same size $R^{20\times16\times10\times32}$ and $\mu = 2$, $R = 5$:

- Random tensor [40]: randomly generate.

- Stochastic tensor [129]: $\tau_{i_1,i_2,i_3,i_4} = \begin{cases} c & i_1 \neq i_2, i_2 \neq i_3, i_3 \neq i_4 \\ 0 & i_1 = i_2, i_2 \neq i_3, i_3 \neq i_4 \\ 1/20 & \text{otherwise} \end{cases}$, where c is randomly in $(0, 1)$ by the homogenous distribution such as $\sum_{i_1 \in [\![20]\!]} \tau_{i_1,i_2,i_3,i_4} = 1$ with $i_j \neq i_{j+1}, j = 1, 2, 3$.

- Cauchy tensor [39]: $\tau_{i_1,i_2,i_3,i_4} = \frac{1}{c(i_1)+c(i_2)+c(i_3)+c(i_4)}$, where c is a random vector with size 32.

- Hilbert tensor [165]: $\tau_{i_1,i_2,i_3,i_4} = \frac{1}{i_1+i_2+i_3+i_4-3}$.

- Toeplitz tensor [41]: $\tau_{i_1+j,i_2+j,i_3+j,i_4+j} = \tau_{i_1,i_2,i_3,i_4}$ for $j \in [\![min(20 - i_1, 16 - i_2, 10 - i_3, 32 - i_4)]\!]$.

The initial vectors are chosen by four different ways:

- 'Random Initial'–unit vectors $\mathbf{u}_r^{(\ell)}$ for $\ell = 1, \ldots, k$ and $r = 1, \ldots, R$ are generated randomly to satisfy orthogonality constrain with $\mu = 2$.

- 'Identity Initial'–each $[\mathbf{u}_1^{(\ell)}, \ldots, \mathbf{u}_R^{(\ell)}]$ for $\ell = 1, \ldots, k$ are taken as the first R columns of identity matrices.

- 'Orthogonal Initial'–each $[\mathbf{u}_1^{(\ell)}, \ldots, \mathbf{u}_R^{(\ell)}]$ for $\ell = 1, \ldots, k$ are taken as the first R columns of random orthonormal matrices.

- 'Singular Value Initial'–the major left singular vectors of the unfoldings of the tensors are used as initials.

The numerical results for the first 150 iterations are shown in Figure 4.1–Figure 4.5.

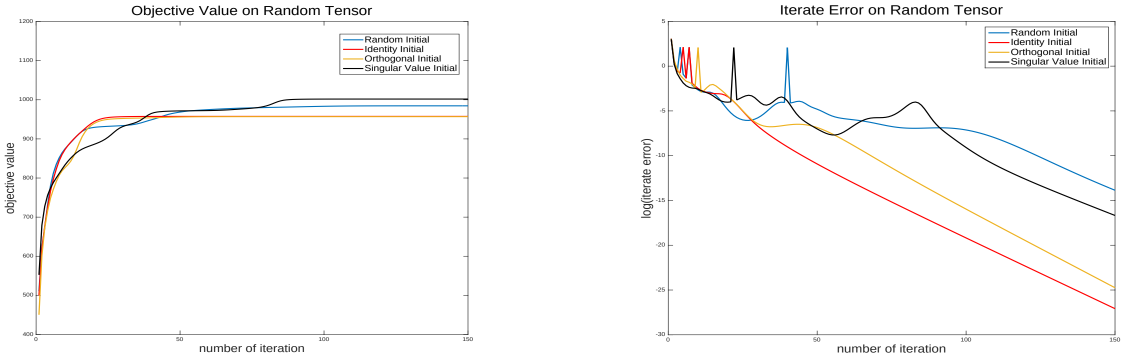

Figure 4.1: Comparison on Random Tensor.

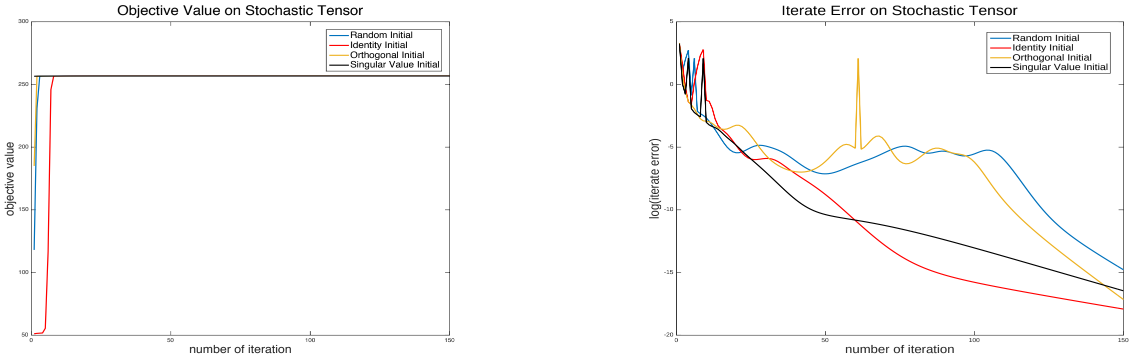

Figure 4.2: Comparison on Stochastic Tensor.

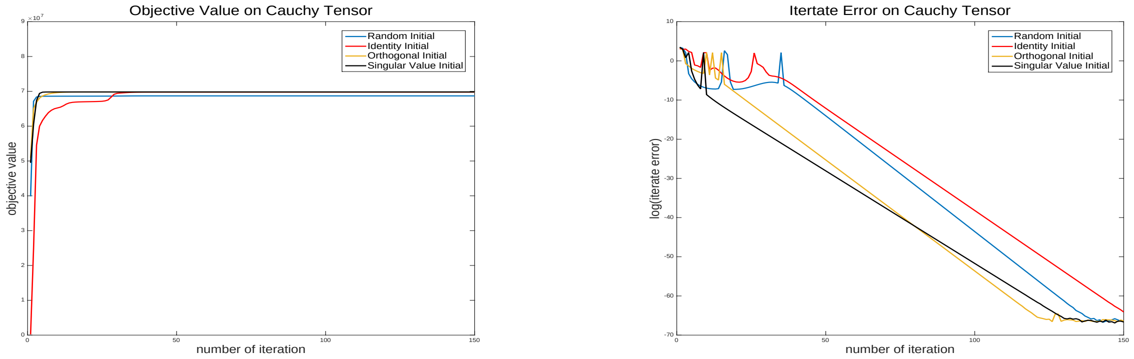

Figure 4.3: Comparison on Cauchy Tensor.

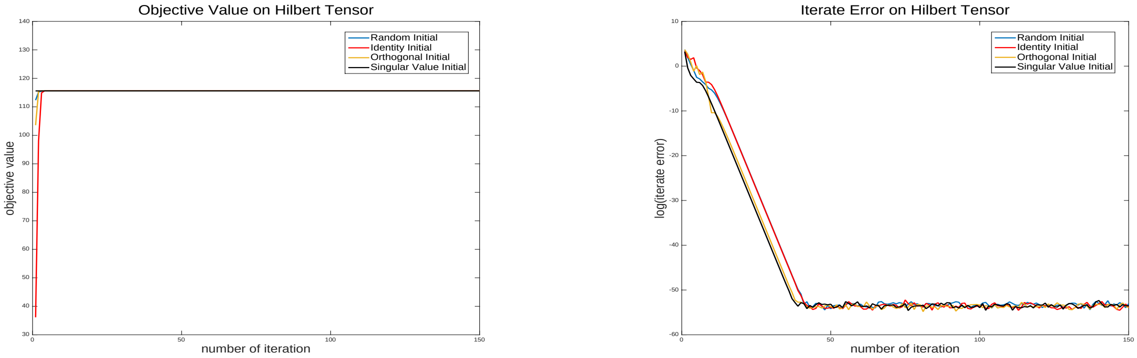

Figure 4.4: Comparison on Hilbert Tensor.

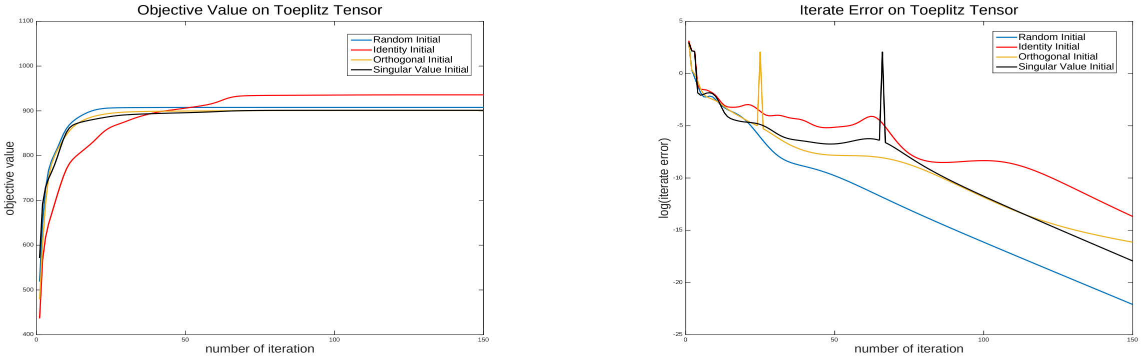

Figure 4.5: Comparison on Toeplitz Tensor.

Figure 4.1–Figure 4.5 lead to the following observations:

- Objective value satisfies the monotone increasing property for each iteration as proved in the previous section;

- For different initial vectors, the approximated objective values may be different for the same test tensor, that is, iterates may converge to different limit points. Hence, the computed result is only optimal in a local neighborhood for each initial vector. As addressed in [40], it is interesting to study for what tensors or what initial guesses Algorithm 7 converges to the global optimum.

- Iterates converge, but they are not monotone in each step. Note that iterate errors increase suddenly, for example, at step 61 with "Orthogonal Initial" on Stochastic tensor, step 25 with "Orthogonal Initial" and step 66 with "Singular Value Initial" on Toeplitz tensor. The reason might be that the largest singular value of $C^{(\ell)}_{r,[p]}$ involved in Algorithm 7 is not simple at that certain iteration steps.

- Iterates converge but slower than that of objective values. Thus, it is important to study strategies to speed-up the convergence of iterates.

- When it comes to the qualities of the final approximation, among 4 different initial vectors, none does offer obvious advantage. It is challenging to study for what initial guesses Algorithm 7 converges faster in terms of both objective values and iterate errors.

Chapter 5

General Convergence of ADM

5.1 Introduction

Many algorithms can be cast in the abstract form

$$\left\{\begin{array}{rcl}\mathbf{x}_{k+1} & = & f(\mathbf{y}_k),\\ \mathbf{y}_{k+1} & = & g(\mathbf{x}_{k+1}),\end{array}\right. \quad k = 0, 1, \ldots, \tag{5.1}$$

where $f : U \to V$ and $g : V \to U$, referred to henceforth as the generating functions, are maps representing some black-box evaluations or some intermediate numerical procedures. The variables $\mathbf{x}$ and $\mathbf{y}$ can be vectors, matrices, or even functions. The choice of U, V depends on the desired properties of the variables $\mathbf{x}$ and $\mathbf{y}$, which can be, for instance, nonnegative, orthogonal, or stochastic. In this chpater, we focus only on finite dimensional variables, so the feasible sets U, V are subsets in some Euclidean spaces with suitable dimensions and constraints. We shall give several interesting but nontrivial examples in the later part of this discussion to demonstrate this point. For more complicated problems involving n variables $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}$, a similar alternating iteration can be written in this form

$$\begin{cases} \mathbf{x}_{k+1}^{(1)} &= f^{(1)}(\mathbf{x}_k^{(2)}, \mathbf{x}_k^{(3)}, \dots, \mathbf{x}_k^{(n)}), \\ \mathbf{x}_{k+1}^{(2)} &= f^{(2)}(\mathbf{x}_{k+1}^{(1)}, \mathbf{x}_k^{(3)}, \dots, \mathbf{x}_k^{(n)}), \\ &\vdots \\ \mathbf{x}_{k+1}^{(n)} &= f^{(n)}(\mathbf{x}_{k+1}^{(1)}, \mathbf{x}_{k+1}^{(2)}, \dots, \mathbf{x}_{k+1}^{(n-1)}), \end{cases} \quad k = 0, 1, \dots. \tag{5.2}$$

Perhaps the simplest algorithm in the form of (5.2) is the Gauss-Seidel iterative scheme used for solving a linear system where all maps $f^{(\ell)}$ are linear and $\mathbf{x}^{(\ell)}$ are scalars. Another example is the alternating least squares (ALS) method used for low rank tensor approximations of a given order-n tensor [52,139,179,189,197], where all variables are expected to be of unit length. We shall concentrate on the analysis for (5.1) in this chapter. The generalization to (5.2) can be accomplished in a similar way.

Obviously, the sequence $\{\mathbf{y}_k\}$ generated by (5.1) can be obtained from the fixed-point iteration

$$\mathbf{y}_{k+1} = g(f(\mathbf{y}_k)), \quad k = 0, 1, \dots. \tag{5.3}$$

If the composite $F := g \circ f$, referred to henceforth as the transition function (of one sweep for $\mathbf{y}_k$), is a contraction map, then the Banach fixed-point theorem asserts that the iterates from (5.3) converge to a unique fixed point. This is the most impeccable conclusion, but often proving that $g \circ f$ is a contraction map is difficult or impossible. Likewise, if $g \circ f$ is continuous and maps a convex compact set into itself, then the Brouwer fixed-point theorem asserts that there is a fixed-point $\mathbf{y}^*$ such that $g \circ f(\mathbf{y}^*) = \mathbf{y}^*$. In general, however, not much is known about the limiting behavior of the sequence $\{\mathbf{y}_k\}$ itself. For many of the algorithms discussed in the literature and even used in practice, we find that lacking a rigorous convergence analysis for the iterates $\{\mathbf{x}_k\}$ and $\{\mathbf{y}_k\}$ themselves is a serious and widespread shortfall [21,139].

5.1.1 Summary

The main contribution of this chapter is a general framework for characterizing the limiting behavior of (5.1) under much easy-to-check criteria. We apply the framework to a variety of alternating direction methods and, in particular, the alternating least squares algorithms to demonstrate how the theory facilitates convergence analysis, some of which are difficult to come by otherwise.

5.1.2 Outline of the chapter

This chapter is organized as follows. In Section 5.2, we build our framework by progressively adding in conditions. The theory works in its most basic form, but more conditions make it easier to draw conclusions. As a demonstration, we apply the theory in Section 5.3 to a variety of classical results in the literature. In this context, the proof of convergence is not new, but it shows the versatility of our framework. In Section 5.4, we use our theory to argue the convergence of algorithms for the Tucker nearest problem and the structured Kronecker approximation problem.

5.2 Basic theory

We begin our theory with the most basic form, namely, checking the difference between every convergent subsequence and its immediate next iterate. The detailed prove of following lemma can be found in Lemma 2.2.6.

Lemma 5.2.1. *Assume that a^* is an isolated accumulation point of a sequence $\{a_k\}$ such that for every subsequence $\{a_{k_j}\}$ converging to a^*, there is an infinite subsequence $\{a_{k_{j_i}}\}$ such that $|a_{k_{j_i}+1} - a_{k_{j_i}}| \to 0$. Then the whole sequence $\{a_k\}$ converges to a^*.*

To apply Lemma 5.2.1 to algorithms such as (5.1), we follow the steps that

a. Check to see that an accumulation point of a convergent subsequence $\{a_{k_j}\}$ is isolated. (See the remarks following Corollary 5.2.1 and Lemma 5.2.2.)

b. Search for a subsequence $\{a_{k_{j_i}}\}$ such that after applying the transition map, say F, the difference $|F(a_{k_{j_i}}) - a_{k_{j_i}}|$ diminishes to zero.

For specific applications, see our recent work on the convergence of the ALS algorithm and the SVD-based algorithm for the best rank-1 tensor approximations in [79, 189].

By imposing the continuity on the generating function and the finiteness on isolated accumulation points, the following lemma asserts a specific limiting behavior of the resulting iterates.

Theorem 5.2.1. *Let $F : U \longrightarrow U$ be a continuous map over a closed subset $U \subset \mathbb{R}^n$. Suppose that the sequence $\{\mathbf{z}_k\}$ generated by iterative scheme $\mathbf{z}_{k+1} = F(\mathbf{z}_k)$ is well defined, bounded, and has finitely many isolated accumulation points. Then*

1. *Either the sequence $\{\mathbf{z}_k\}$ converges, or*

2. *There are disjoint neighborhoods of the accumulation points such that, for k large enough, the consecutive elements $\mathbf{z}_k, \mathbf{z}_{k+1}, \ldots$ visit each neighborhood in a cyclic order.*

Proof. Let $\left\{\mathbf{z}_{k_i}\right\}$ denote an arbitrary convergent subsequence of $\{\mathbf{z}_k\}$. By continuity, the subsequence $\left\{\mathbf{z}_{k_i+1}\right\}$ also converges. Repeating this process, we denote the limiting behavior when $i \to \infty$ as

$$\begin{array}{rcl} \mathbf{z}_{k_i} & \longrightarrow & \mathbf{z}_0^* \\ \mathbf{z}_{k_i+1} & \longrightarrow & \mathbf{z}_1^* = F(\mathbf{z}_0^*) \\ \mathbf{z}_{k_i+2} & \longrightarrow & \mathbf{z}_2^* = F(\mathbf{z}_1^*) \\ \vdots & & \vdots \end{array} \tag{5.4}$$

The sequence $\{\mathbf{z}_0^*, \mathbf{z}_1^*, \ldots\}$ is part of the accumulation points of $\{\mathbf{z}_k\}$ and thus must be finite. Let $s \geq 0$ be the smallest integers such that $\mathbf{z}_{s+p}^* = \mathbf{z}_s^*$ for some positive integer p. Then by continuity, we have $\mathbf{z}^*_{s+p+1} = \mathbf{z}^*_{s+1}$, and so on. In this way, elements in $\{\mathbf{z}^*_0, \dots \mathbf{z}^*_{s+p-1}\}$ are distinct and are the only accumulation points in the process of (5.4).

As these points are isolated, there exists $\epsilon > 0$ such that the spheres $N_\epsilon(\mathbf{z}^*_q)$ centered at $\mathbf{z}^*_q$ with radius ϵ, $q = 0, 1, \dots, s+p-1$, are disjoint from each other. For each fixed integer t, all but finitely many points from this sequence $\{\mathbf{z}_{k_i+t}\}$ belong to $N_\epsilon(\mathbf{z}^*_q)$ with

$$q := \begin{cases} t, & \text{if } 0 \le t \le s, \\ s + ((t-s) \bmod p), & \text{if } s < t. \end{cases}$$

On the other hand, for a fixed $\mathbf{z}_{k_i}$ with sufficiently large i, write $\mathbf{z}_{k_j} = \mathbf{z}_{k_i+t_j}$ with $t_j := k_j - k_i$ for all $j > i$. Since $\mathbf{z}_{k_j} \in N_\epsilon(\mathbf{z}^*_0)$ when j is sufficiently large, we conclude that the two conditions

$$\begin{cases} s &= 0, \\ (k_j - k_i) \bmod p &= 0, \quad \text{for all sufficiently large } i, j \end{cases} \tag{5.5}$$

must hold simultaneously.

Suppose $\{\mathbf{z}_{\ell_j}\}$ is an arbitrary convergent subsequence of $\{\mathbf{z}_k\}$. For each ℓ_j, let k_{i_j} be one of the indices $\{k_i\}$ that is smaller than ℓ_j. Then $\mathbf{z}_{\ell_j} = \mathbf{z}_{k_{i_j}+(\ell_j - k_{i_j})}$ and hence all but finitely many elements in $\{\mathbf{z}_{\ell_j}\}$ must belong to one of these balls $N_\epsilon(\mathbf{z}^*_q)$. In this way, we have proved that all convergent subsequences of $\{\mathbf{z}_k\}$ satisfy (5.5).

If $p = 1$, then $q = 0$, the sequence $\{\mathbf{z}_k\}$ converges to $\mathbf{z}^*_0$. If $p > 1$, then $\{\mathbf{z}_k\}$ does not converge, but its elements for sufficiently large k must be distributed in such a way as residing alternately among $N_\epsilon(\mathbf{z}^*_q)$ in the order $q = 0, \dots, p-1$. □

It is informative to remark further on the three conditions required by Theorem 5.2.1 as follows:

a. *The sequence* $\{\mathbf{z}_k\}$ *being bounded.* This usually poses no additional burden because it is the prerequisite for convergence.

b. *The generating function F being continuous.* If F is given in analytic form, then its continuity can easily be checked. However, if F is given as a computational procedure, then cautions should be taken to ensure the continuity. For example, if $F(Y)$ refers to the orthogonal matrix $U(Y)$ in the singular value decomposition of the matrix $Y = U\Sigma V^{\top}$, then in theory U can be made to be continuously dependent on Y [29, 192]. But if U is obtained by a certain SVD algorithm, then the signs of columns of $U(Y_1)$ may differ from those of $U(Y_2)$ even if Y_2 is close to Y_1, leading to discontinuous jumps in the numerical outcomes. An easy fix in the procedure is due.

c. *The accumulation points being finite and geometrically isolated.* This is the most demanding task. Even so, there are multiple avenues to tackle this task. For example, in many algorithm formulations the model (5.1) is actually a polynomial system in the variables $\mathbf{x}$ and $\mathbf{y}$. The notion of algebraic geometry might be used as a tool for arguing the finite cardinality and isolation of solutions.

The following lemma from the theory of parameter continuation [164, Theorem 7.1.1] is often useful for checking the last condition above.

Lemma 5.2.2. *Let $P(\mathbf{z};\mathbf{q})$ be a system of n polynomials in variables $\mathbf{z} \in \mathbb{C}^n$ and parameters $\mathbf{q} \in \mathbb{C}^m$. Let $\mathcal{N}(\mathbf{q})$ denote the number of geometrically isolated solutions to $P(\mathbf{z};\mathbf{q}) = 0$ over the algebraically closed complex space. Then,*

1. *$\mathcal{N}(\mathbf{q})$ is finite, and it is the same, say $\mathcal{N}$, for almost all $\mathbf{q} \in \mathbb{C}^m$;*

2. *For all $\mathbf{q} \in \mathbb{C}^m$, $\mathcal{N}(\mathbf{q}) \leq \mathcal{N}$;*

3. *The subset of $\mathbb{C}^m$ where $\mathcal{N}(\mathbf{q}) = \mathcal{N}$ is a Zariski open set. That is, the exceptional subset of $\mathbf{q} \in \mathbb{C}^m$ where $\mathcal{N}(\mathbf{q}) < \mathcal{N}$ is an affine algebraic set contained within an algebraic set of codimension one.*

Since $\mathbb{R}^n$ (indeed, the closure of any infinite subset) is Zariski dense in $\mathbb{C}^n$, the above statements hold for almost all parameters $\mathbf{q} \in \mathbb{R}^m$, except that the number of real-valued isolated solutions varies as a function of $\mathbf{q}$ and is no longer a constant. For our applications, we only need the fact that the real roots of a polynomial system are finite and geometrically isolated for generic $\mathbf{q}$.

The argument in Theorem 5.2.1 can be generalized to multi-level iterative schemes such as (5.1). Suppose that both functions f and g are continuous and that the sequences $\{\mathbf{x}_k\}$ and $\{\mathbf{y}_k\}$ generated are bounded and have finitely many isolated accumulation points, respectively. Then any convergent subsequence $\{\mathbf{y}_{k_i}\}$ will lead to a process

$$\begin{array}{rcl} \mathbf{y}_{k_i} & \longrightarrow & \mathbf{y}_0^* \\ \mathbf{x}_{k_i+1} & \longrightarrow & \mathbf{x}_1^* = f(\mathbf{y}_0^*) \\ \mathbf{y}_{k_i+1} & \longrightarrow & \mathbf{y}_1^* = g(\mathbf{x}_1^*) \\ \mathbf{x}_{k_i+2} & \longrightarrow & \mathbf{x}_2^* = f(\mathbf{y}_1^*) \\ \vdots & & \vdots \end{array} \tag{5.6}$$

From this point on, an argument can be made to draw the same conclusion as in Theorem 5.2.1 for both $\{\mathbf{x}_k\}$ and $\{\mathbf{y}_k\}$ simultaneously. In this way, we may also interpret Theorem 5.2.1 as if $F = g \circ f$ applied to $\mathbf{y}$ for (5.1) and similarly for the general scheme (5.2).

An obvious condition for convergence is the exclusion of any possible cyclic behavior. This often can be accomplished if we know additional information such as some monotonicity associated with the iteration.

Corollary 5.2.1. *Suppose that the iteration (5.2) represents an alternating optimization mechanism for an objective function $h(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)})$. Under the same conditions of Theorem 5.2.1 where F denotes the transition function representing one complete sweep of the alternating procedure, the objective function h assumes the same value at all accumulation points.*

Proof. By Theorem 5.2.1, we only need to consider the case when the sequence $\{\mathbf{z}_k\}$ has cyclic behavior. Without loss of generality, it suffices to consider the scheme (5.1) which involves only two variables $\mathbf{z} = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)})$. To fix the idea, we assume that the alternating optimization is doing minimization. By the way the sequence

$\{(\mathbf{x}_k^{(1)}, \mathbf{x}_k^{(2)})\}$ is generated, we should have the relationship

$$h(\mathbf{x}_{k+1}^{(1)}, \mathbf{x}_{k+1}^{(2)}) \leq h(\mathbf{x}_{k+1}^{(1)}, \mathbf{x}_k^{(2)}) \leq h(\mathbf{x}_k^{(1)}, \mathbf{x}_k^{(2)}).$$

Abbreviate $(\mathbf{x}_k^{(1)}, \mathbf{x}_k^{(2)})$ to $\mathbf{z}_k$. The sequence $\{h(\mathbf{z}_k)\}$ is monotone and must converge. If there are more than one isolated accumulation points, let $\mathbf{z}_0^*$ and $\mathbf{z}_1^*$ denote any two such points. The iterates $\{\mathbf{z}_k\}$ must visit arbitrarily diminishing vicinity of each accumulation point infinitely many times. Suppose $h(\mathbf{z}_0^*) < h(\mathbf{z}_1^*)$. Then there exists a neighborhood $N_\epsilon(\mathbf{z}_0^*)$ of $\mathbf{z}_0^*$ such that the iterates cannot possibly "return" to revisit the higher level $\mathbf{z}_1^*$ again once it has visited $N_\epsilon(\mathbf{z}_0^*)$ because of the non-ascending property mentioned earlier. Similarly, it cannot happen that $h(\mathbf{z}_0^*) > h(\mathbf{z}_1^*)$. Therefore, the objective function must assume the same value at all accumulation points. □

Motivated by Corollary 5.2.1, we now impose some mild conditions of smoothness on the part of the optimization mechanism. The following observation is handy for applications.

Theorem 5.2.2. *Suppose that an alternating optimization method can be cast in form of (5.2). Write* $\mathbf{z} = (\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)})$ *where* $\mathbf{x}^{(\ell)} \in U^{(\ell)}$ *and* $U^{(\ell)} \subset \mathbb{R}^{I_\ell}$. *Assume that*

1. *The conditions in Theorem 5.2.1 are satisfied where* $F(\mathbf{z})$ *denotes the transition function of one complete sweep of the alternating optimization,* $\mathbf{z}_{k+1} = F(\mathbf{z}_k)$.

2. *Each generating function* $f^{(\ell)}$ *represents the optimization mechanism in the* ℓ*-th direction, is continuously differentiable, and returns the unique global*[1] *minimizer* $\mathbf{x}_{k+1}^{(\ell)}$ *of the restricted objective function*

$$h_\ell(\mathbf{w}) := h(\mathbf{x}_{k+1}^{(1)}, \ldots, \mathbf{x}_{k+1}^{(\ell-1)}, \mathbf{w}, \mathbf{x}_k^{(\ell+1)}, \ldots, \mathbf{x}_k^{(n)}).$$

[1] What is really needed in the proof is the continuous differentiability of the transition function F. The uniqueness is to ascertain that $f^{(\ell)}$ unambiguously defines $\mathbf{x}_{k+1}^{(\ell)}$. So long as this map $f^{(\ell)}$ is well defined, the requirement of being a global minimizer is not essential.

3. *The objective function $h(\mathbf{z})$ is second order continuously differentiable.*

4. *One of the accumulation points $\mathbf{z}_0^*$ of $\{\mathbf{z}_k\}$ is a local minimizer of $h(\mathbf{z})$ at which the Hessian $\nabla^2 h(\mathbf{z}_0^*)$ is symmetric and positive definite.*

Then the sequence $\{\mathbf{z}_k\}$ converges.

Proof. Let $\{\mathbf{z}_{k_i}\}$ be an arbitrary convergent subsequence with limit point $\mathbf{z}_0^*$. We claim that the spectral radius $\rho(F'(\mathbf{z}_0^*))$ is strictly less than 1. If this claim is true, then there exists a neighborhood $N_\epsilon(\mathbf{z}_0^*)$ in which F is a contraction. That is, for any convergent subsequence $\{\mathbf{z}_{k_j}\} \subset N_\epsilon(\mathbf{z}_0^*)$, the subsequence $\{F(\mathbf{z}_{k_j})\}$ is also contained in $N_\epsilon(\mathbf{z}_0^*)$. Since $\{\mathbf{z}_{k_j+1}\}$ must also converge by the continuity of F, it converges to $\mathbf{z}_0^*$. By Lemma 5.2.1, we know the sequence $\{\mathbf{z}_k\}$ converges.

It only remains to prove that $\rho(F'(\mathbf{z}_0^*)) < 1$. It suffices to consider the case (5.1) only. The proof can be extended to the general case (5.2). The following argument is modified from the ideas in [22, Lemma 2]. Define $H : U^{(1)} \times U^{(2)} \times U^{(1)} \times U^{(2)} \to U^{(1)} \times U^{(2)}$ by

$$H(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}; \mathbf{y}^{(1)}, \mathbf{y}^{(2)}) := \begin{bmatrix} \frac{\partial h}{\partial \mathbf{x}^{(1)}}(\mathbf{x}^{(1)}, \mathbf{y}^{(2)}) \\ \\ \frac{\partial h}{\partial \mathbf{x}^{(2)}}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) \end{bmatrix}, \tag{5.7}$$

where the right hand side denotes the partial gradient of H with respect to the variables $(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})$, but evaluated at different points. Define also $G : U^{(1)} \times U^{(2)} \to U^{(1)} \times U^{(2)}$ by

$$G(\mathbf{y}^{(1)}, \mathbf{y}^{(2)}) := H(F(\mathbf{y}^{(1)}, \mathbf{y}^{(2)}); \mathbf{y}^{(1)}, \mathbf{y}^{(2)}). \tag{5.8}$$

Given any $(\mathbf{y}_k^{(1)}, \mathbf{y}_k^{(2)})$ near $\mathbf{z}_0^*$, observe that

$$G(\mathbf{y}_k^{(1)}, \mathbf{y}_k^{(2)}) = H(\mathbf{y}_{k+1}^{(1)}, \mathbf{y}_{k+1}^{(2)}, \mathbf{y}_k^{(1)}, \mathbf{y}_k^{(2)}) = \begin{bmatrix} \frac{\partial h}{\partial \mathbf{y}^{(1)}}(\mathbf{y}_{k+1}^{(1)}, \mathbf{y}_k^{(2)}) \\ \\ \frac{\partial h}{\partial \mathbf{y}^{(2)}}(\mathbf{y}_{k+1}^{(1)}, \mathbf{y}_{k+1}^{(2)}) \end{bmatrix} = 0,$$

because $\mathbf{y}_{k+1}^{(1)}$ and $\mathbf{y}_{k+1}^{(2)}$ are the respective global minimizers of the restrictive objective functions h_1 and h_2. We see that $G \equiv 0$ in a neighborhood $N_\epsilon(\mathbf{z}_0^*)$. From (5.8), the evaluation of the Jacobian of G at $\mathbf{z}_0^*$ yields

$$\left(\frac{\partial H}{\partial(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})} \frac{\partial F}{\partial(\mathbf{y}^{(1)}, \mathbf{y}^{(2)})} + \frac{\partial H}{\partial(\mathbf{y}^{(1)}, \mathbf{y}^{(2)})}\right)\Big|_{\mathbf{z}_0^*} = 0, \tag{5.9}$$

where by (5.7) we have

$$\begin{aligned}\frac{\partial H}{\partial(\mathbf{x}^{(1)},\mathbf{x}^{(2)})} &= \begin{bmatrix} \frac{\partial}{\partial \mathbf{x}^{(1)}}(\frac{\partial h}{\partial \mathbf{x}^{(1)}}) & 0 \\ \frac{\partial}{\partial \mathbf{x}^{(1)}}(\frac{\partial h}{\partial \mathbf{x}^{(2)}}) & \frac{\partial}{\partial \mathbf{x}^{(2)}}(\frac{\partial h}{\partial \mathbf{x}^{(2)}}) \end{bmatrix}, \\ \frac{\partial H}{\partial(\mathbf{y}^{(1)},\mathbf{y}^{(2)})} &= \begin{bmatrix} 0 & \frac{\partial}{\partial \mathbf{x}^{(2)}}(\frac{\partial h}{\partial \mathbf{x}^{(1)}}) \\ 0 & 0 \end{bmatrix}.\end{aligned}$$

Note that the above two matrices make up

$$\nabla^2 h(\mathbf{z}_0^*) = (\frac{\partial H}{\partial(\mathbf{x}^{(1)},\mathbf{x}^{(2)})} + \frac{\partial H}{\partial(\mathbf{y}^{(1)},\mathbf{y}^{(2)})})\Big|_{\mathbf{z}_0^*},$$

which is assumed to be symmetric and positive definite. It follows from (5.9) that

$$F'(\mathbf{z}_0^*) = -(\frac{\partial H}{\partial(\mathbf{x}^{(1)},\mathbf{x}^{(2)})}\Big|_{\mathbf{z}_0^*})^{-1}(\frac{\partial H}{\partial(\mathbf{y}^{(1)},\mathbf{y}^{(2)})}\Big|_{\mathbf{z}_0^*}) \tag{5.10}$$

is well defined. Furthermore, we see in (5.10) that $F'(\mathbf{z}_0^*)$ is of the form $-(D-L)^{-1}U$ which is precisely the iteration matrix if the (block) Gauss-Seidel scheme is applied to solving a linear equation where the coefficient matrix A is split as $A = D-L-U$ [150, Theorem 7.1.9]. Since the Gauss-Seidel method converges when A is symmetric and positive definite, we know that $\rho(F'(\mathbf{z}_0^*)) < 1$. □

Alternating optimization, or more generally alternating direction, is not usually the best approach for solving the underlying problem. However, swapping one complicated problem of many variables with a sequence of simpler problems each of which handles and adjusts one subset of variables a time can sometimes be implemented more easily and offer computational convenience. The above theory outlines a basic convergence analysis framework for these types of alternating direction iterations. In the remaining portion of this chapter, we discuss some interesting applications. Some of the convergence results are new.

5.3 Applications to some known cases

The convergence behavior of examples in this section is well understood in the literature. Certainly we are not trying to reinvent the wheels. Rather, we use these

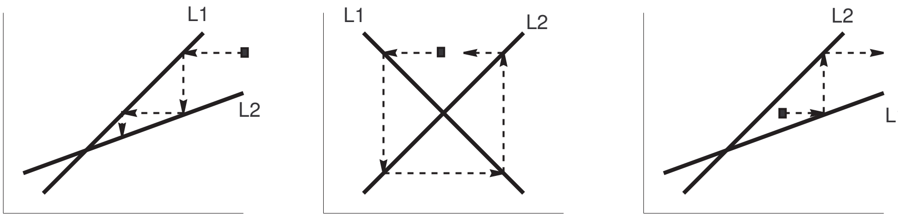

Figure 5.1: Converging, cyclic, and diverging behavior of Gauss-Seidel iterations in $\mathbb{R}^2$.

known facts to demonstrate the subtleties in dealing with convergence when some of the conditions mentioned in the preceding section are not met. On the other hand, we also demonstrate that our framework offers an alternative and unified proof of convergence which is much simpler than some of those already done in the literature.

5.3.1 The Gauss-Seidel method for solving a system of linear equations

The classical Gauss-Seidel iteration scheme is of the form (5.2). It is well known that the method applied to the linear system $A\mathbf{x} = \mathbf{b}$ with non-zero elements on the diagonals does not always produce a convergent result. Convergence is guaranteed only in a few cases such as the matrix A being diagonally dominant or being symmetric and positive definite. In the event that the Gauss-Seidel method fails to converge for a given A, what has happened is that either the iterates become unbounded or the iterates go cyclically, as has been characterized in Theorem 5.2.1. The directions of variables are alternated by satisfying one linear equation a time. See Figure 5.1 for a graphical interpretation of the Gauss-Seidel method applied to a 2-dimensional problem. The scheme itself does not contain any type of optimization in its iteration.

5.3.2 The power method for finding the dominant eigenvector

Given a matrix $A \in \mathbb{R}^{n\times n}$ and an initial unit vector $\mathbf{y}_0 \in \mathbb{R}^n$, the power method

$$\begin{cases} \mathbf{x}_{k+1} = A\mathbf{y}_k, \\ \mathbf{y}_{k+1} = \frac{\mathbf{x}_{k+1}}{\|\mathbf{x}_{k+1}\|_\infty}, \end{cases} \tag{5.11}$$

is in the form of (5.1). The sequences $\{\mathbf{x}_k\}$ and $\{\mathbf{y}_k\}$ are clearly bounded. The functions $f^{(\ell)}$, $\ell = 1, 2$, on the right side of (5.11) are clearly continuous. Excluding the extraneous zero solution after scaling the second equation by a multiplier $\|\mathbf{x}_{k+1}\|_\infty$, the entire system can be regarded as a polynomial system depending on the parameter A. By Lemma 5.2.2, we know that for almost all matrices $A \in \mathbb{R}^{n\times n}$, the stationary points are finite and isolated. By Theorem 5.2.1, we conclude that the iterates generated by the power method converge for a generic A. In numerical linear algebra, we know even more specifics when the method fails to converge, e.g., when A has multiple dominant eigenvalues, in which case the matrix A has a multi-dimensional eigenspace and the system (5.11) has non-isolated stationary points. See also Section 5.3.5 for more detailed discussion from the perspective of the high-order power method.

5.3.3 The alternating least squares method for computing the QR decomposition

There are efficient algorithms for computing the fundamentally important QR decomposition of a given matrix $A \in \mathbb{R}^{m\times n}$. Surely it is of little value to try to find this decomposition by the alternating least squares approach

$$\begin{cases} R_{k+1} := \underset{R = \text{upper triangular}}{\arg\min} \|A - Q_k R\|_F, \\ Q_{k+1} := \underset{Q^\top Q = I_n}{\arg\min} \|A - QR_{k+1}\|_F. \end{cases} \tag{5.12}$$

Despite its inefficiency, however, the scheme (5.12) is theoretically doable. Indeed, it can be argued that R_{k+1} is the upper triangular part of the matrix $Q_k^\top A$ and Q_{k+1} is exactly the orthogonal portion in the polar decomposition of $AR_{k+1}^\top$ (which is more expensive than the QR decomposition itself). By construction, the objective values $\|A - Q_k R_k\|_F$ descend and converge, but possibly to a nonzero value. Clearly, the sequences $\{Q_k\}$ and $\{R_k\}$ are bounded and the abstract functions defining them are continuous. The stationary points must satisfy the optimality conditions

$$\left\{\begin{array}{rcl} R & = & \mathsf{triu}(Q^\top A), \\ Q^\top A R^\top & = & R A^\top Q, \end{array}\right.$$

which is a linear polynomial system in Q with A as the parameter and, by Lemma 5.2.2, has finitely many isolated solutions for generic $A \in \mathbb{R}^{m\times n}$. The conditions in Theorem 5.2.2 are satisfied, so the iterates $\{Q_k\}$ and $\{R_k\}$ do converge, even though not necessarily they converge to the QR decomposition of A.

5.3.4 The alternating projection method for finding structured low rank matrices

Let $\mathcal{R}(r)$ denote the set of all rank r matrices and Ω the set of matrices with a prescribed structure, say, Toeplitz or Hankel matrices. Then the desired set of structured rank r matrices can be regarded as the intersection of these two sets. To find a structured low rank matrix, if exist, the idea of alternating projections between these two sets can be employed [31,45,46]. The process is to satisfy the rank constraint and the structural constraint alternately while the distance in between is being reduced. The geometry of lift and project is depicted in Figure 5.2. The procedures outlined in Algorithm 8 obviously fits the basic model (5.1) where both actions of lifting and the projection are continuous. Since $\mathcal{R}(r)$ is not convex, the iteration might stagnate back and forth between $\mathcal{R}(r)$ and Ω. In that case, an intersection has not been found, but still the iterates converge to a locally nearest location between the two geometric entities by our theory.

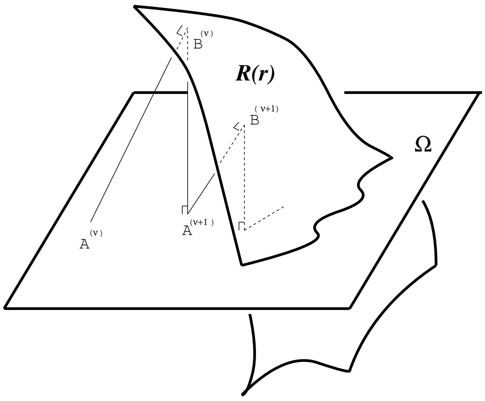

Figure 5.2: Alternating projections between lower rank matrices and structured matrices

Algorithm 8 (Lift-and-project algorithm.)

Input: Given an arbitrary $A^{(0)} = A \in \Omega$

Output: A pair of matrices that locally minimizes the distance between $\mathcal{R}(r)$ and Ω

```
repeat
    lift: Compute the rank $r$ matrix $B^{(\nu)}$ in $\mathcal{R}(r)$ that is nearest to $A^{(\nu)}$.
    project: Compute the projection $A^{(\nu+1)}$ of $B^{(\nu)}$ onto the subspace $\Omega$.
until the sequence $\{A^{(\nu)}\}$ meets stopping criteria
```

5.3.5 Best rank-one tensor approximation

We discuss the rank-one approximation first. The Tucker nearest problem will be discussed in Section 5.4.1.

The most popular approach for the best rank-one approximation is the notion of alternating least squares method. The procedures are described in Algorithm 9, where the subscript $\cdot_{[p]}$ indicates the quantities resulting from the p-th iteration, $\widehat{\mathbf{u}}^{(\ell)}$ means to exclude this vector from the list, and

$$T \circledast_\ell S := [\langle \tau_{:,\nu_\ell,:}, S \rangle] \in \mathbb{R}^{I_\ell}, \quad \nu_\ell = 1, \ldots, I_\ell, \tag{5.13}$$

with $\tau_{:,\nu_\ell,:}$ denoting the ν_ℓ-th "slice" of the tensor T in the ℓ-th direction and $\langle \cdot, \cdot \rangle$ the Frobenius inner product generalized to multi-dimensional arrays.

While the limiting behavior of the objective values $\{\lambda^{(\ell)}_{[p]}\}$ is easy to understand,

Algorithm 9 (High-order power method.)

Input: A generic order-k tensor T and k unit vectors $\mathbf{u}^{(1)}_{[0]} \in \mathbb{R}^{I_1}, \ldots, \mathbf{u}^{(k)}_{[0]} \in \mathbb{R}^{I_k}$,

Output: A local best rank-1 approximation to T

```
1: for $p = 0, 1, \cdots,$ do
2:     for $\ell = 1, 2, \cdots, k$ do
3:         $\mathbf{u}^{(\ell)}_{[p+1]} = T\circledast_\ell (\mathbf{u}^{(1)}_{[p+1]}\otimes\ldots\otimes\mathbf{u}^{(\ell-1)}_{[p+1]}\otimes\widehat{\mathbf{u}}^{(\ell)}\otimes\mathbf{u}^{(\ell+1)}_{[p]}\ldots\otimes\mathbf{u}^{(k)}_{[p]})$
4:         $\lambda^{(\ell)}_{[p+1]} := \|\mathbf{u}^{(\ell)}_{[p+1]}\|_2$
5:         $\mathbf{u}^{(\ell)}_{[p+1]} := \frac{\mathbf{u}^{(\ell)}_{[p+1]}}{\lambda^{(\ell)}_{[p+1]}}$
6:     end for
7: end for
```

it has taken tremendous effort to prove the convergence of the iterates $\{\mathbf{u}^{(\ell)}_{[p]}\}$ themselves [179, 189]. We now apply our theory to Algorithm 9 to demonstrate how the convergence can be argued in a quick and convenient way.

First, the definition of $\mathbf{u}^{(\ell)}_{[p+1]}$ in Line 3 followed by Line 5 gives rise to precisely the unique global maximizer of the function

$$\lambda^{(\ell)}_{[p+1]}(\mathbf{w}) := \langle T, \mathbf{u}^{(1)}_{[p+1]}\otimes\ldots\otimes\mathbf{u}^{(\ell-1)}_{[p+1]}\otimes\mathbf{w}\otimes\mathbf{u}^{(\ell+1)}_{[p]}\ldots\otimes\mathbf{u}^{(k)}_{[p]}\rangle$$

which is the restriction of the objective function

$$\lambda(\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(k)}) = \langle T, \mathbf{u}^{(1)}\otimes\ldots\otimes\mathbf{u}^{(k)}\rangle \tag{5.14}$$

to the ℓ-th direction subject to the constraint of unit length. As a polynomial in variables $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(k)}$, the smoothness of λ and the associated $\lambda^{(\ell)}_{[p+1]}$ is guaranteed. The first order optimality condition for a stationary point of (5.14) is to satisfy the system of $\sum_{\ell=1}^{k} I_\ell$ polynomials [130, 189]

$$T\circledast_\ell (\mathbf{u}^{(1)}\otimes\ldots\otimes\widehat{\mathbf{u}}^{(\ell)}\otimes\ldots\otimes\mathbf{u}^{(k)}) = \langle T, \mathbf{u}^{(1)}\otimes\ldots\otimes\mathbf{u}^{(k)}\rangle\mathbf{u}^{(\ell)}, \quad \ell = 1, \ldots, k, \tag{5.15}$$

which, by Lemma 5.2.2, contains only geometrically isolated solutions for a generic tensor T. Conditions in Theorem 5.2.1 are satisfied generically. It is easy to see that the sequence $\{\lambda(\mathbf{u}^{(1)}_{[p]}, \ldots, \mathbf{u}^{(k)}_{[p]})\}$ is monotone non-decreasing. Assuming the generic condition that the Hessian of λ at such a local maximizer is negative definite, then the convergence of the iterates $\{(\mathbf{u}_{[p]}^{(1)}, \ldots, \mathbf{u}_{[p]}^{(k)})\}$ is ensured by Theorem 5.2.2.

5.4 Applications to some new problems

In this section, we apply our theory to two important yet challenging problems in the field — the Tucker nearest problem and the structured Kronecker approximation problem. While numerical algorithms have been proposed and used in practice, we find little discussion of convergence analysis in the literature. This is probably due to the fact that the algorithms usually involve complex algebraic manipulations. Nonetheless, our framework requires fairly mild conditions on these manipulations. We can explain the convergence.

For the ease of later reference, we restate the notion of orthogonality and polar decomposition defined in Lemma 4.3.1 which will appear in both problems. Let $\mathscr{S}(p, q)$ denote the Stiefel manifold of matrices in $\mathbb{R}^{p\times q}$ with orthonormal columns and $\mathbb{I}_q$ the identity matrix in $\mathbb{R}^{q\times q}$. The following lemma is essentially the well known polar decomposition [89, 90, 102], yet its view as the normal bundle of an element Q on $\mathscr{S}(p, q)$ is useful for the subsequent discussion [47].

Lemma 5.4.1. *Given a matrix $Q \in \mathscr{S}(p, q)$, then a matrix $Z \in \mathbb{R}^{p\times q}$ whose orthogonal projection to $\mathscr{S}(p, q)$ is precisely Q if and only if $Q^\top Z$ is symmetric.*

Proof. Let $Q_\perp$ denote the matrix in $\mathscr{S}(p, p-q)$ so that the augmented matrix $[Q, Q_\perp]$ is orthogonal. It is easy to see that the tangent space $\mathscr{T}_Q\mathscr{S}(p, q)$ at $Q \in \mathscr{S}(p, q)$ is made of matrices in the form

$$H = QK + Q_\perp Q_\perp^\top W,$$

where $K \in \mathbb{R}^{q\times q}$ is skew-symmetric and $W \in \mathbb{R}^{p\times q}$ is arbitrary. For the vector $Z - Q$ to be perpendicular to the surface $\mathscr{S}(p, q)$, it must be such that

$$\mathbf{Proj}_{\mathscr{T}_Q\mathscr{S}(p,q)}(Z - Q) = Q\frac{Q^\top(Z - Q) - (Z - Q)^\top Q}{2} + Q_\perp Q_\perp^\top(Z - Q) = 0. \quad (5.16)$$

Note that the two terms in the middle equation of (5.16) are mutually orthogonal. Therefore, each term must be zero by itself. Upon simplification, we see that $Z - Q$ is perpendicular to $\mathscr{S}(p,q)$ if and only if

$$\begin{cases} Q^\top Z &= Z^\top Q, \\ Q_\perp^\top Z &= 0. \end{cases} \tag{5.17}$$

Given Q, (5.17) is a homogeneous linear system of $pq - \frac{q(q+1)}{2}$ independent equations in pq unknowns of Z. So the solutions form a subspace of dimension $\frac{q(q+1)}{2}$. Indeed, if we write the columns of $Z \in \mathbb{R}^{p\times q}$ in terms of the orthonormal basis

$$Z = QS + Q_\perp T,$$

where $S \in \mathbb{R}^{q\times q}$ and $T \in \mathbb{R}^{(p-q)\times q}$, then Z is a solution to (5.17) if and only if $T = Q_\perp^\top Z = 0$ and $S = Q^\top Z$ is symmetric. □

In the above lemma, we look up from a given $Q \in \mathscr{S}(p,q)$ for its normal bundle in $\mathbb{R}^{p\times q}$. Now we look down from a given $Z \in \mathbb{R}^{p\times q}$ for its projection onto $\mathscr{S}(p,q)$.

Corollary 5.4.1. *Given an arbitrary $Z \in \mathbb{R}^{p\times q}$, suppose that $Z = UP$ is the polar decomposition of Z where $U \in \mathscr{S}(p,q)$ and $P \in \mathbb{R}^{q\times q}$ is symmetric and positive semi-definite. Then U is the projection of Z onto $\mathscr{S}(p,q)$ and is the nearest matrix in $\mathscr{S}(p,q)$ to Z.*

In the polar decomposition, we stress that the symmetric matrix $P = U^\top Z$ is always unique, but U is unique only if Z is of full column rank.

5.4.1 Tucker nearest problem

Given the rank parameter $\mathbf{r} = (r_1, \ldots, r_k)$, an order-k tensor in the form

$$A = \sum_{j_1=1}^{r_1} \cdots \sum_{j_k=1}^{r_k} \beta_{j_1,\ldots,j_k} \mathbf{v}_{j_1}^{(1)} \otimes \ldots \otimes \mathbf{v}_{j_k}^{(k)} \in \mathbb{R}^{I_1\times\ldots\times I_k} \tag{5.18}$$

with orthonormal vectors $\mathbf{v}_{j_\ell}^{(\ell)} \in \mathbb{R}^{I_\ell}$ is said to be in the Tucker format with core tensor

$$\boldsymbol{\beta} := [\beta_{j_1,\ldots,j_k}] \in \mathbb{R}^{r_1 \times \ldots \times r_k}. \tag{5.19}$$

If we assemble the orthonormal vectors into factor matrices by denoting

$$V^{(\ell)} := [\mathbf{v}_1^{(\ell)}, \ldots, \mathbf{v}_{r_\ell}^{(\ell)}] \in \mathbb{R}^{I_\ell \times r_\ell}, \quad \ell = 1, \ldots, k, \tag{5.20}$$

then $V^{(\ell)} \in \mathscr{S}(I_\ell, r_\ell)$ and the tensor A in (5.18) can be written as

$$A = \boldsymbol{\beta} \times_1 V^{(1)} \times_2 V^{(2)} \times_3 \ldots \times_k V^{(k)}, \tag{5.21}$$

where $\times_d$ denotes the mode-d product[2] [114]. Given an order-k tensor $T \in \mathbb{R}^{I_1 \times \ldots \times I_k}$, the Tucker nearest problem is to find a tenor in the Tucker form (5.21) with a fixed rank parameter $\mathbf{r}$ such that

$$\widetilde{h}(\boldsymbol{\beta}, V^{(1)}, \ldots, V^{(k)}) := \|\boldsymbol{\beta} \times_1 V^{(1)} \times_2 V^{(2)} \times_3 \ldots \times_k V^{(k)} - T\|_F \tag{5.22}$$

is minimized.

For an order-k tensor $T \in \mathbb{R}^{I_1 \times \ldots \times I_k}$, let $\mathbf{vec}(T)$ denote the linear array where the entry $\tau_{i_1,\ldots,i_k}$ of T is saved at the location

$$i_1 + \sum_{s=2}^{k} (i_s - 1) \prod_{t=1}^{s-1} I_t \tag{5.23}$$

of the array. Then it can be verified that (5.21) is equivalent to [15, Formula (12)]

$$\mathbf{vec}(A) = (V^{(k)} \otimes \ldots \otimes V^{(1)}) \mathbf{vec}(\boldsymbol{\beta}), \tag{5.24}$$

where $\otimes$ stands for the Kronecker product. We make it clear that the notation $\otimes$ defined between vectors is the outer product [91, 123] while $\otimes$ stands for the Kronecker product between matrices in this thesis. The expression above sheds an

[2] Given an order-k tensor $T \in \mathbb{R}^{I_1 \times \ldots \ldots \times I_d \times \ldots \times I_k}$ and a matrix $M \in \mathbb{R}^{m \times I_d}$, the mode-$d$ product $\Theta = T \times_d M$ is defined to be the tensor in $\mathbb{R}^{I_1 \times \ldots \times I_{d-1} \times m \times I_{d+1} \times \ldots \times I_k}$ with element $f_{i_1,\ldots,i_{d-1},t,i_{d+1},\ldots i_k} := \sum_{s=1}^{I_d} m_{t,s} \tau_{i_1,\ldots,i_{d-1},s,i_{d+1},\ldots i_k}$.

important insight — entries in $\mathbf{vec}(\boldsymbol{\beta})$ are the coordinates of $\mathbf{vec}(A)$ in terms of the orthonormal columns of $V^{(k)} \otimes \ldots \otimes V^{(1)}$, i.e.,

$$\mathbf{vec}(\boldsymbol{\beta}) = (V^{(k)} \otimes \ldots \otimes V^{(1)})^\top \mathbf{vec}(A). \tag{5.25}$$

Therefore, given fixed matrices $V^{(\ell)} \in \mathscr{S}(I_\ell, r_\ell)$, $\ell = 1, \ldots, k$, the minimizer $\boldsymbol{\beta}$ in (5.22) is given by the projection of $\mathbf{vec}(T)$ onto the column space of $V^{(k)} \otimes \ldots \otimes V^{(1)}$, or equivalently,

$$\boldsymbol{\beta} := T \times_1 V^{(1)^\top} \times_2 V^{(2)^\top} \times_3 \ldots \times_k V^{(k)^\top} \in \mathbb{R}^{r_1 \times \ldots \times r_k}. \tag{5.26}$$

In this way, the Tucker nearest problem is equivalent to the problem of maximizing the Frobenius norm of the tensor

$$\pi(V^{(1)}, \ldots, V^{(k)}) := T \times_1 V^{(1)^\top} \times_2 V^{(2)^\top} \times_3 \ldots \times_k V^{(k)^\top}, \tag{5.27}$$

subject to the constraint that $V^{(\ell)} \in \mathscr{S}(I_\ell, r_\ell)$, $\ell = 1, \ldots, k$.

The relationship (5.26) can further be expressed in terms of the mode-d unfolding [15, Formula (11)]

$$\boldsymbol{\beta}_{(d)} = V^{(d)^\top} \underbrace{T_{(d)}(V^{(k)} \otimes \ldots \otimes V^{(d+1)} \otimes V^{(d-1)} \otimes \ldots \otimes V^{(1)})}_{\Upsilon_{(d)}}, \quad d = 1, \ldots k, \tag{5.28}$$

where the mode-d unfolding $T_{(d)}$ is simply a rearrangement of T into a matrix of size $I_d \times \prod_{\ell \neq d} I_\ell$ by assigning the element $(T_{(d)})_{i_d, j} := \tau_{i_1, \ldots, i_k}$ with $j = 1 + \sum_{s=1, s\neq d}^{k} (i_s - 1) \prod_{t=1}^{s-1} I_t$. Likewise, $\boldsymbol{\beta}_{(d)}$ is an unfolding of size $r_d \times \prod_{\ell \neq d} r_\ell$. Taking advantage of the form (5.28) by modifying one factor matrix $V^{(\ell)}$ a time via the singular value decomposition, Algorithm 10 therefore has been proposed in the field as a way for tackling the Tucker nearest problem. By construction, we also know that

$$\begin{aligned} \lambda_{[p]} := \|\pi(V_{[p]}^{(1)}, \ldots, V_{[p]}^{(k)})\|_F &\leq \lambda_{[p+1]}^{(1)} \leq \lambda_{[p+1]}^{(2)} \leq \cdots \leq \lambda_{[p+1]}^{(k)} \\ &= \|\pi(V_{[p+1]}^{(1)}, \ldots, V_{[p+1]}^{(k)})\|_F, \end{aligned} \tag{5.29}$$

so the convergence of scalars $\{\lambda_{[p]}\}$ is clear. Thus far, however, we have not seen any proof of convergence for the iterates $\{(V_{[p]}^{(1)}, \ldots, V_{[p]}^{(k)})\}$ in the literature. Using our framework, we can establish the convergence as follows.

Without loss of generality, consider the objective function to be maximized as

$$h(V^{(1)},\dots,V^{(k)}) = \frac{1}{2}\|\pi(V^{(1)},\dots,V^{(k)})\|_F^2 = \frac{1}{2}\langle {V^{(d)}}^\top \Upsilon_{(d)}, {V^{(d)}}^\top \Upsilon_{(d)}\rangle \tag{5.30}$$

which, as indicated in (5.28), has the same value $\frac{\|\beta\|_F^2}{2}$ for all $d = 1,\dots,k$. Clearly, h is secondly order continuous differentiable. The definition of $V_{[p+1]}^{(\ell)}$ at Line 5 is the unique global maximizer of the restricted function

$$h_\ell(W) := \frac{1}{2}\|\pi(V_{[p+1]}^{(1)},\dots V_{[p+1]}^{(\ell-1)}, W, V_{[p]}^{(\ell+1)},\dots,V_{[p]}^{(k)})\|_F^2, \tag{5.31}$$

subject to the constraint that $W \in \mathscr{S}(I_\ell, r_\ell)$, so Algorithm 10 is an ALS algorithm.

To apply our framework, we need to check out two additional conditions. First, the partial gradient of h with respect to a general $V^{(d)}$ is given by

$$\nabla^{(d)} h(V^{(d)}) := \frac{\partial h}{\partial V^{(d)}} = \Upsilon_{(d)}\Upsilon_{(d)}^\top V^{(d)}, \quad d = 1,\dots,k. \tag{5.32}$$

At a stationary point, the projection of $\nabla^{(d)}h(V^{(d)})$ onto the tangent space of $\mathscr{S}(I_d, r_d)$ is zero, implying that

$$\Upsilon_{(d)}\Upsilon_{(d)}^\top V^{(d)} = V^{(d)}{V^{(d)}}^\top \Upsilon_{(d)}\Upsilon_{(d)}^\top V^{(d)}, \quad d = 1,\dots,k. \tag{5.33}$$

In other words, the stationary points of the objective function (5.30) are solutions to a system of $\sum_{d=1}^k I_d r_d$ polynomials (5.33) that is parameterized by T. By Lemma 5.2.2, we conclude that for almost all tensors, the accumulation points of Algorithm 10 are finite and geometrically isolated.

Second, each of the constraint $V^{(\ell)} \in \mathscr{S}(I_\ell, r_\ell)$, $\ell = 1,\dots,k$, is a compact set. The local maximizer for h does exist. The Hessian of h in (5.30), which depends on T, at its local maximizer is necessarily negative semi-definite. Furthermore, positive definite matrices form an open set whose boundaries consist of positive semi-definite matrices which resides on a submanifold of codimension 1. A small perturbation can easily disrupt the semi-definiteness. We may therefore assume that for almost all tensors, the Hessian of h at one of the stationary point is symmetric and positive definite.

By now, all conditions in Theorem 5.2.2 are satisfied. To our knowledge, the following result is new.

Algorithm 10 (HOSVD method for Tucker nearest problem.)

Input: A generic order-k tensor T, a fixed rank parameter $\mathbf{r}$, and k initial matrix $V_{[0]}^{(\ell)} \in \mathbb{R}^{I_\ell \times r_\ell}$ with orthonormal columns,

Output: A local best Tucker approximation to T

```
for $p = 0, 1, \cdots,$ do
    for $\ell = 1, 2, \cdots, k$ do
        $B_{[p+1]}^{(\ell)} := T_{(\ell)}(V_{[p]}^{(k)} \otimes \ldots \otimes V_{[p]}^{(\ell+1)} \otimes V_{[p+1]}^{(\ell-1)} \otimes \ldots \otimes V_{[p+1]}^{(1)})$    {Of size $I_\ell \times \prod_{j=1, j\neq \ell}^{k} r_j$.}
        $[U, S, \tilde{}\,] = \mathsf{svds}(B_{[p+1]}^{(\ell)}, r_\ell)$ {Compute the largest $r_\ell$ singular values and left singular vectors.}
        $V_{[p+1]}^{(\ell)} := U$
        $\lambda_{[p+1]}^{(\ell)} = \|S\|_F$
    end for
end for
```

Theorem 5.4.1. *For almost all order-k tensor T, the iterates $\{(V_{[p]}^{(1)}, \ldots, V_{[p]}^{(k)})\}$ generated by Algorithm 10 converge to a local solution of the Tucker nearest problem.*

5.4.2 Structured Kronecker approximation

Given $A \in \mathbb{R}^{m\times n}$ with $m = m_1 m_2$ and $n = n_1 n_2$ and a small enough but fixed integer r, the Kronecker approximation problem concerns finding matrices $B_i \in \mathbb{R}^{m_1\times n_1}$ and $C_i \in \mathbb{R}^{m_2 \times n_2}$ such that the objective function

$$\phi_A(B_1, \ldots, B_r, C_1, \ldots, C_r) = \|A - \sum_{i=1}^{r} B_i \otimes C_i\|_F^2, \tag{5.34}$$

is minimized [146]. The problem is equivalent to a rank-r approximation problem [181]

$$\|A - \sum_{i=1}^{r} B_i \otimes C_i\|_F = \|\mathscr{R}(A) - \sum_{i=1}^{r} \mathbf{vec}(B_i)\mathbf{vec}(C_i)^\top\|_F, \tag{5.35}$$

where $\mathscr{R}(A) \in \mathbb{R}^{m_1 n_1 \times m_2 n_2}$ is a rearrangement of A as

$$\mathscr{R}(A) := \begin{bmatrix} \mathbf{vec}(A_{1,1})^\top \\ \mathbf{vec}(A_{2,1})^\top \\ \vdots \\ \mathbf{vec}(A_{m_1,n_1})^\top \end{bmatrix},$$

if A is partitioned as a $m_1 \times n_1$ block matrix with blocks $A_{ij} \in \mathbb{R}^{m_2 \times n_2}$,

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1,n_1} \\ A_{21} & A_{22} & \cdots & A_{2,n_1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m_1,1} & A_{m_1,2} & \cdots & A_{m_1,n_1} \end{bmatrix}.$$

The Kronecker approximation problem (5.34) therefore can be solved effectively by using the truncated singular value decomposition.

It is important to note that the Kronecker product often inherits structures from its factors. For example, the following properties are listed in [180].

$$\text{If } B \text{ and } C \text{ are} \left\{ \begin{array}{l} \text{nonsingular} \\ \text{lower(upper) triangular} \\ \text{banded} \\ \text{symmetric} \\ \text{positive definite} \\ \text{stochastic} \\ \text{Toeplitz} \\ \text{permutations} \\ \text{orthogonal} \end{array} \right\}, \text{then } B \otimes C \text{ is} \left\{ \begin{array}{l} \text{nonsingular} \\ \text{lower(upper) triangular} \\ \text{banded} \\ \text{symmetric} \\ \text{positive definite} \\ \text{stochastic} \\ \text{Toeplitz} \\ \text{permutations} \\ \text{orthogonal} \end{array} \right\}.$$

Also, with respect to factorizations, the LU-with-partial-pivoting, Cholesky, and QR factorizations of $B \otimes C$ merely require the corresponding factorizations of B and C. An interesting question about the converse then arises, which we refer to as the structured Kronecker approximation problem. Let $\Omega_B \subset \mathbb{R}^{m_1 \times n_1}$ and $\Omega_C \subset \mathbb{R}^{m_2 \times n_2}$

denote the subsets of desired structures of factors, respectively. How should the approximation (5.34) be accomplished if it is expected that $B_i \in \Omega_B$ and $C_i \in \Omega_C$, even if the given A is not structured?

In what follows, we consider only the case $r = 1$. Generalizations to general r is possible but with tedious manipulations. See, for example, the work in [100] for the block Toeplitz structure. Once the procedure such that the generating function is specified, we think that it is possible that our framework is still applicable.

For the case $r = 1$, the following result naturally defines an alternating procedure. In [181, Theorem 4.1], the result can be interpreted as the power method applied to $\mathscr{R}(A)$ for finding the left and right singular vectors associated with its largest singular value.

Lemma 5.4.2. *Let $A \in \mathbb{R}^{m\times n}$ with $m = m_1m_2$ and $n = n_1n_2$ be given.*

1. *Suppose $C \in \mathbb{R}^{m_2\times n_2}$ is fixed, then the matrix $B \in \mathbb{R}^{m_1\times n_1}$ defined by*

$$b_{ij} := \frac{\langle A_{ij}, C\rangle}{\langle C, C\rangle}, \quad 1 \le i \le m_1, \quad 1 \le j \le n_1, \tag{5.36}$$

minimizes $\|A - B \otimes C\|_F$.

2. *Suppose $B \in \mathbb{R}^{m_1\times n_1}$ is fixed, then the matrix $C \in \mathbb{R}^{m_2\times n_2}$ defined by*

$$c_{ij} := \frac{\langle \widetilde{A}_{ij}, B\rangle}{\langle B, B\rangle}, \quad 1 \le i \le m_2, \quad 1 \le j \le n_2, \tag{5.37}$$

where $\widetilde{A}_{ij} = A(i : m_2 : m, j : n_2 : n) \in \mathbb{R}^{m_1\times n_1}$, minimizes $\|A - B \otimes C\|_F$.

The above lemma can then be exploited to answer a few structured approximation problems, provided that A is similarly structured. We mention, for example, the cases that

$$\text{If } A \text{ and } B \text{ are} \left\{ \begin{array}{l} \text{nonnegative} \\ \text{symmetric} \\ \text{positive definite} \end{array} \right\},$$

Algorithm 11 (ALS method for structured Kronecker approximation.)

Input: A generic matrix $A \in \mathbb{R}^{m_1 m_2 \times n_1 n_2}$, two specific structures Ω_B and Ω_C, an initial matrix $C_0 \in \Omega_C$,

Output: A local best structured Kronecker approximation to A.

```
1: for $k = 0, 1, \cdots,$ do
2:     $B_{k+1} = \arg\min_{B \in \Omega_B} \|A - B \otimes C_k\|_F$
3:     $C_{k+1} = \arg\min_{C \in \Omega_C} \|A - B_{k+1} \otimes C\|_F$
4: end for
```

$$\text{then the minimizer } C \text{ of } \|A - B \otimes C\|_F \text{ is } \left\{ \begin{array}{l} \text{nonnegative} \\ \text{symmetric} \\ \text{positive definite} \end{array} \right\}.$$

For other structures, including the case that the given A does not have any structure at all, the formulas in Lemma 5.4.2 does not preserve the structures in general. Some other numerical procedures are needed.

The prototypical ALS procedure proposed in Algorithm 11 is a plausible procedure to tackle the structured Kronecker approximation problem, provided the structured least squares subproblems at Lines 2 and 3 can be resolved. Even so, the nonlinear nature of the Kronecker product would make a formal proof of convergence of the iterates for the general case challenging. Our contribution is that, if the procedures can be checked to satisfy the conditions demanded in Theorem 5.2.2, then our framework kicks in and the method will converge.

To demonstrate our point, we concentrate on two special structures — orthogonal factors and stochastic factors — in the subsequent discussion. We propose algorithmic details for computing the structured least squares solutions and carry out the crucial task of checking that the conditions in Theorem 5.2.2 are met. At the end, we are able to draw the conclusion of convergence.

Orthogonal factors. To fix the idea, we restate our problem: Given $A \in \mathbb{R}^{m_1 m_2 \times n_1 n_2}$, where $m_1 \geq n_1$ and $m_2 \geq n_2$, find $Q_1 \in \mathscr{S}(m_1, n_1)$ and $Q_2 \in \mathscr{S}(m_2, n_2)$

so that the objective function

$$g(Q_1, Q_2) := \frac{1}{2}\|A - Q_1 \otimes Q_2\|_F^2 \tag{5.38}$$

is minimized. We shall consider the constraint as the manifold $\mathscr{S}(m_1, n_1) \times \mathscr{S}(m_2, n_2)$ with the product topology.

To find the critical point for the constrained optimization of (5.38), we compute the projected gradient of $g(Q_1, Q_2)$. We begin with the action of the Fréchet derivative of $g(Q_1, Q_2)$ at a general point $(H_1, H_2) \in \mathbb{R}^{m_1 \times n_1} \times \mathbb{R}^{m_2 \times n_2}$. Under the product topology, we may consider the partial derivatives separately. Thus, the action of the partial derivative of g with respect to Q_1 on H_1 is given by

$$\begin{aligned}\frac{\partial g}{\partial Q_1}.H_1 &= \langle -H_1 \otimes Q_2, A - Q_1 \otimes Q_2\rangle \\ &= -\langle \mathbf{vec}(H_1), \mathscr{R}(A - Q_1 \otimes Q_2)\mathbf{vec}(Q_2)\rangle \\ &= -\langle H_1, \mathscr{A} \circledast_{(m_1,n_1)} Q_2 - n_2 Q_1\rangle,\end{aligned}$$

where the block matrix A is considered as an order-4 tensor $\mathscr{A} \in \mathbb{R}^{m_1 \times n_1 \times m_2 \times n_2}$ and, similar to the operation (5.13),

$$\mathscr{A} \circledast_{(m_1,n_1)} Q_2 := \big[\langle A_{ij}, Q_2\rangle\big] \in \mathbb{R}^{m_1 \times n_1}.$$

Similarly,

$$\frac{\partial g}{\partial Q_2}.H_2 = -\langle H_2, \mathscr{A} \circledast_{(m_2,n_2)} Q_1 - n_1 Q_2\rangle$$

with

$$\mathscr{A} \circledast_{(m_2,n_2)} Q_1 := \left[\langle \widetilde{A}_{ij}, Q_1\rangle\right] \in \mathbb{R}^{m_2 \times n_2}.$$

By the Riesz representation theorem, the partial gradients of $g(Q_1, Q_2)$ can be interpreted as

$$\left\{\begin{aligned}\tfrac{\partial g}{\partial Q_1} &= n_2 Q_1 - \mathscr{A} \circledast_{(m_1,n_1)} Q_2, \\ \tfrac{\partial g}{\partial Q_2} &= n_1 Q_2 - \mathscr{A} \circledast_{(m_2,n_2)} Q_1.\end{aligned}\right. \tag{5.39}$$

We now project the partial gradients onto the tangent spaces of the respective Stiefel spaces. Applying (5.16) to both partial gradients, we obtain

$$\left\{\begin{array}{cl} & \mathbf{Proj}_{\mathscr{T}_{Q_1}\mathscr{S}(m_1,n_1)}\frac{\partial g}{\partial Q_1} \\ = & Q_1\frac{(\mathscr{A}\circledast_{(m_1,n_1)}Q_2)^\top Q_1 - Q_1^\top(\mathscr{A}\circledast_{(m_1,n_1)}Q_2)}{2} - (\mathbb{I}_{m_1} - Q_1Q_1^\top)\mathscr{A}\circledast_{(m_1,n_1)} Q_2, \\ & \\ & \mathbf{Proj}_{\mathscr{T}_{Q_2}\mathscr{S}(m_2,n_2)}\frac{\partial g}{\partial Q_2} \\ = & Q_2\frac{(\mathscr{A}\circledast_{(m_2,n_2)}Q_1)^\top Q_2 - Q_2^\top(\mathscr{A}\circledast_{(m_2,n_2)}Q_1)}{2} - (\mathbb{I}_{m_2} - Q_2Q_2^\top)\mathscr{A}\circledast_{(m_2,n_2)} Q_1. \end{array}\right.$$

We now are ready to characterize the first order optimality condition for the orthogonal Kronecker approximation problem (5.38).

Lemma 5.4.3. *For (Q_1, Q_2) to be a critical point for (5.38), it must be such that*

1. *Q_1 is the orthogonal portion in the polar decomposition of $\mathscr{A}\circledast_{(m_1,n_1)} Q_2$, and*

2. *Q_2 is the orthogonal portion in the polar decomposition of $\mathscr{A}\circledast_{(m_2,n_2)} Q_1$*

simultaneously.

Proof. The first order optimality condition is that the projected gradients should be zero. The conclusion follows from the argument used in proving Corollary 5.4.1. □

Based on this characterization, we are now able to define the two steps at Lines 2 and 3 in Algorithm 11 more specifically as in Algorithm 12 for the orthogonal Kronecker approximation. Furthermore, using our framework, we are able to argue for the convergence of the algorithm under the following assumptions.

Theorem 5.4.2. *Assume that*

1. *The given matrix A is such that the Hessian of the corresponding objective function g defined in (5.38) is positive definite at one of its local minimizers; and*

2. *The initial matrix $Q_2^{(0)} \in \mathscr{S}(m_2, n_2)$ is such that the subsequent matrices $\{\mathscr{A}\circledast_{(m_1,n_1)} Q_2^{(p)}\}$ and $\{\mathscr{A}\circledast_{(m_2,n_2)} Q_1^{(p+1)}\}$ defined in Algorithm 12 are of full column rank in $\mathbb{R}^{m_1\times n_1}$ and $\mathbb{R}^{m_2\times n_2}$, respectively.*

Then the sequence $\{(Q_1^{(p)}, Q_2^{(p)})\}$ *generated by Algorithm 12 converges to a local solution to the orthogonal Kronecker approximation problem.*

Proof. To apply our framework for convergence, the conditions needed by Theorem 5.2.2 should be satisfied by Algorithm 12. We check out two particular conditions, while others are either obvious or assumed.

Observe first that the definitions at Lines 2 and 3 actually represent an ALS optimization mechanism because

$$g(Q_1, Q_2) = \|\mathscr{R}(A) - \mathbf{vec}(Q_1)\mathbf{vec}(Q_2)^\top\|_F^2 = \|\mathscr{A} \circledast_{(m_1,n_1)} Q_2 - Q_1\|_F^2$$

and, by Corollary 5.4.1, the nearest $Q \in \mathscr{S}(p,q)$ to a fixed point $Z \in \mathbb{R}^{p\times q}$ comes from the polar decomposition of Z. The polar decomposition is unique for a full rank matrix and is continuous in its parameters.

Observe next that the accumulation points of the iteration must satisfy the system of polynomials [130, 189]

$$\left\{\begin{array}{rcl} Q_1^\top(\mathscr{A} \circledast_{(m_1,n_1)} Q_2) &=& (\mathscr{A} \circledast_{(m_1,n_1)} Q_2)^\top Q_1, \\ Q_2^\top(\mathscr{A} \circledast_{(m_2,n_2)} Q_1) &=& (\mathscr{A} \circledast_{(m_2,n_2)} Q_1)^\top Q_2, \\ \mathscr{A} \circledast_{(m_1,n_1)} Q_2 &=& Q_1 Q_1^\top(\mathscr{A} \circledast_{(m_1,n_1)} Q_2), \\ \mathscr{A} \circledast_{(m_2,n_2)} Q_1 &=& Q_2 Q_2^\top(\mathscr{A} \circledast_{(m_2,n_2)} Q_1). \end{array}\right. \tag{5.40}$$

which, by Lemma 5.2.2, contains only geometrically isolated solutions for almost all data matrix A. The iterates $\left\{(Q_1^{(p)}, Q_2^{(p)})\right\}$ are obviously bounded as they are from the Stiefel manifolds. Conditions in Theorem 5.2.2 are satisfied. □

We remark that the first assumption in Theorem 5.4.2 holds for generic A. We conjecture that the second assumption is also true for generic A and $Q_2^{(0)}$ because, otherwise, rank deficient matrices are the union of low dimensional manifolds and are susceptible to perturbations. At present we do not have a formal proof of the genericity, so we state them as assumptions.

Stochastic factors. Again, we first restate the problem: Let $\mathcal{M}(q)$ denote the convex and compact subset of all column stochastic matrices in $\mathbb{R}^{q\times q}$. Given

Algorithm 12 (Polar method for orthogonal Kronecker approximation.)

Input: A generic matrix $A \in \mathbb{R}^{m_1 m_2 \times n_1 n_2}$, and an initial matrix $Q_2^{(0)} \in \mathscr{S}(m_2, n_2)$,

Output: A local best orthogonal Kronecker approximation to A

```
for $p = 0, 1, \cdots,$ do
    $[Q_1^{(p+1)}, P_1^{(p+1)}] = \mathsf{poldec}(\mathscr{A} \circledast_{(m_1,n_1)} Q_2^{(p)})$        {using polar decomposition.}
    $[Q_2^{(p+1)} P_2^{(p+1)}] = \mathsf{poldec}(\mathscr{A} \circledast_{(m_2,n_2)} Q_1^{(p+1)})$        {using polar decomposition.}
end for
```

$A \in \mathbb{R}^{n_1 n_2 \times n_1 n_2}$, the stochastic Kronecker approximation concerns finding the factors $B \in \mathscr{M}(n_1)$ and $C \in \mathscr{M}(n_2)$ so that the objective function

$$\psi(B, C) := \frac{1}{2}\|A - B \otimes C\|_F^2 \tag{5.41}$$

is minimized.

It is worth mentioning that the problem has an interesting interpretation. The entry of $B \otimes C$ has the form $b_{ij} c_{st}$. Thus, the approximation amounts to aggregating the $n_1 n_2$ states into n_1 groups $G_1, \ldots, G_{n_1}$, each of size n_2, such that the transition probability among states within each group is the same. Thus, b_{ij} stands for the probability of transition from group G_j to state G_i while c_{st} stands for the probability of transition from state t to state s within any group.

Each of the two structured least squares subproblems in Algorithm 11 can easily be formulated to take into the stochastic structure. For instance, the subproblem

$$\min_{\mathbf{1}_{n_1}^\top B = \mathbf{1}_{n_1}^\top, B \geq 0} \|A - B \otimes C\|_F^2, \tag{5.42}$$

where $C \in \mathscr{M}(n_2)$ is fixed and $\mathbf{1}_{n_1} \in \mathbb{R}^{n_1}$ is the column vector of all ones, is a classical constrained linear least squares problem which can be solved via existent optimization software package [125]. Furthermore, the problem (5.42) is a convex programming problem. If we assume the generic condition that the data are such that the objective function is strictly convex, then the solution to (5.42) is unique. Replacing the constraints in Algorithm 11 by $\mathscr{M}(n_1)$ and $\mathscr{M}(n_2)$, and equipped with the ability to solve each subproblem of the restricted objective functions, our concern is whether the iteration will converge.

To apply our theory, we need to check in particular the finiteness and isolation of stationary points. The procedure should be quite routine now, except that the feasible sets now have boundaries, i.e., some of the entries of B or C are zero. The projection at the boundaries is equivalent to the KKT conditions. For simplicity, we shall omit the details. We only demonstrate the projected gradient for the problem (5.42) at an interior point. The partial gradient of ψ with respect to B is

$$\frac{\partial \psi}{\partial B} = B\|C\|_F^2 - (A \circledast_{(n_1,n_1)} C) \in \mathbb{R}^{n_1 \times n_1}. \tag{5.43}$$

The tangent space of $\mathscr{M}(n_1)$ is made of matrices whose column sum is zero. The projection of any $Z \in \mathbb{R}^{n_1 \times n_1}$ onto the tangent space of $\mathscr{M}(n_1)$ is trivially given by

$$\mathbf{Proj}_{\mathscr{T}_B(\mathscr{M}(n_1))}(Z) = Z - \mathbf{1}_{n_1} \left[\frac{\sum_{i=1}^{n_1} z_{i,1}}{n_1}, \ldots, \frac{\sum_{i=1}^{n_1} z_{i,n_1}}{n_1} \right].$$

So the projected gradient can be calculated. Likewise, the projected gradient of ψ with respect to C can be calculated. In all, setting the projected gradient of $\psi(B, C)$ to zero is equivalent to a system of polynomials which, by Lemma 5.2.2, contains finitely many geometrically isolated solutions for a generic A. Without filling in more details, we have sketched a proof by using our theory that the matrices generated by the ALS iteration for the stochastic Kronecker approximation problem converge almost surely.

Chapter 6

Conclusion

In this thesis, we mainly focus on finding the best low rank CP approximation for rank-1 and rank-R ($R > 1$) respectively. Our algorithms are all based on the well developed notion of singular value decomposition (SVD) which update two factors simultaneously. Some backgrounds and applications of tensor decompositions and approximations are introduced in the first chapter, especially on CP model. Moreover, some existing algorithms and convergence analysis for tensor low rank approximations are also mentioned in Chapter 1.

Applying SVD offers a simpler alternative argument that generically the best rank-1 approximation to given symmetric tensor is symmetric. As a by-product, three SVD-based algorithms are proposed in Chapter 2 for computing the symmetric best rank-1 approximation, which should perform superior to the classical ALS methods. The main contribution is on the proof of convergence of both the objective values and the iterates generated by these methods.

In Chapter 3, we consider the best rank-1 approximation of a generic tensor. In contrast to the conventional ALS method that updates one factor a time for the rank-1 tensor approximation, the SVD-based method updates two factors simultaneously. We prove that the iteration by such a mechanism does converge for almost all tensors under Condition A. It is conjectured that tensors satisfying Condition A are generic, but an analytic proof is yet to be further investigated. For large scale problems, numerical experiments suggest that the SVD-based methods do have the advantage of saving the computational time. On the other hand, partly due to the nonlinearity of the objective function, the SVD-based methods do not necessarily provide a better approximation in the long run.

In Chapter 4, an SVD-based algorithm has been presented for the orthogonal low rank approximation problem (4.7) of tensors, which includes the completely orthogonal low rank approximation [40] and semi-orthogonal low rank approximation [190] as two special cases. The convergence of the proposed algorithm has been analyzed. Numerical examples have been provided to illustrate the convergence behavior of our algorithm.

A general theory has been established in Chapter 5 as a useful tool for arguing that an alternating optimization method will converge under mild conditions. The conditions are the continuity of the algorithm, the differentiability of the objective function, the boundedness, finiteness, and geometrical isolation of the accumulation points. An array of problems arising from different backgrounds are demonstrated to be under this framework and satisfy these conditions. In particular, algorithms designed for the Tucker nearest problem and the structured Kronecker approximation problems are shown to converge, which is perhaps new in the literature. The theory might serve as an algorithmic foundation for many other methods having the characteristics of iteration by alternating variables.

Besides the above contents, we are also interested in many other tensor problems such as tensor train, segment CP and Tucker approximation, quantum entanglement and so on.

Chapter 7

List of Author's Publications

The work in this thesis has appeared in the following papers written or co-written by the author.

[1] Y. Guan, M. T. Chu, and D. Chu, *SVD-based algorithms for the best rank-1 approximation of a symmetric tensor*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 1095–1115.

[2] Y. Guan, M. T. Chu, and D. Chu, *Convergence analysis of an SVD-based algorithm for the best rank-1 tensor approximation*, Linear Algebra Appl., 555 (2018), pp. 53–69.

[3] Yu Guan, Nan Jiang, Bo Dong and Moody T. Chu, *Convergence Analysis of Alternating Direction Methods: A General Framework and Its Applications to Tensor Approximations*, submitted.

[4] Y. Guan and D. Chu, *Numerical Computation for Orthogonal Low Rank Approximation of Tensors*, submitted.

Bibliography

[1] E. Acar, C. A. Bingol, H. Bingol, R. Bro, and B. Yener, *Multiway analysis of epilepsy tensors*, Bioinformatics, 23 (2007), pp. i10–i18.

[2] ——, *Seizure recognition on epilepsy feature tensor*, in in EMBS 2007:Proceedings of the 29th Annual International Conference on Engineering in Medicine and Biology Society, IEEE, 2007, pp. 4273–4276.

[3] E. Acar, S. A. Çamtepe, M. S. Krishnamoorthy, and B. Yener, *Modeling and multiway analysis of chatroom tensors*, in Proceedings of the 2005 IEEE international conference on Intelligence and Security Informatics, ISI'05, Berlin, Heidelberg, 2005, Springer-Verlag, pp. 256–268.

[4] E. Acar, S. A. Çamtepe, and B. Yener, *Collective sampling and analysis of high order tensors for chatroom communications*, in Proceedings of the 4th IEEE international conference on Intelligence and Security Informatics, ISI'06, Berlin, Heidelberg, 2006, Springer-Verlag, pp. 213–224.

[5] E. Acar and B. Yener, *Unsupervised multiway data analysis: A literature survey*, IEEE Trans. on Knowl. and Data Eng., 21 (2009), pp. 6–20.

[6] L. Albera, A. Ferréol, P. Comon, and P. Chevalier, *Blind identification of overcomplete mixtures of sources (BIOME)*, Linear Algebra Appl., 391 (2004), pp. 3–30.

[7] C. M. Andersen and R. Bro, *Practical aspects of parafac modeling of fluorescence excitation-emission data*, Journal of Chemometrics, 17 (2003), pp. 200–215.

[8] C. A. Andersson and R. Bro, *The n-way toolbox for matlab*, Chemometrics and Intelligent Laboratory Systems, 52 (2000), pp. 1–4.

[9] C. J. Appellof and E. R. Davidson, *Strategies for analyzing data from video fluorometric monitoring of liquid chromatographic effluents*, Analytical Chemistry, 53 (1981), pp. 2053–2056.

[10] H. Attouch, J. Bolte, and B. F. Svaiter, *Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods*, Math. Program., 137 (2013), pp. 91–129.

[11] B. Bader and T. Kolda, *Matlab tensor toolbox version 2.2, january*, URL http://csmr. ca. sandia. gov/tgkolda/TensorToolbox, (2007).

[12] B. W. Bader, M. W. Berry, and M. Browne, *Discussion tracking in enron email using parafac*, in Survey of Text Mining II, Springer London, 2008, pp. 147–163.

[13] B. W. Bader, R. A. Harshman, and T. G. Kolda, *Temporal analysis of semantic graphs using asalsan*, in Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on, oct. 2007, pp. 33–42.

[14] B. W. Bader and T. G. Kolda, *Algorithm 862: MATLAB tensor classes for fast algorithm prototyping*, ACM Trans. Math. Software, 32 (2006), pp. 635–653.

[15] ——, *Efficient matlab computations with sparse and factored tensors*, SIAM J. Sci. Comput., 30 (2007), pp. 205–231.

[16] S. Banach, *Über homogene polynome in (l^2)*, Studia Mathematica, 7 (1938), pp. 36–44.

[17] K. Batselier, H. Liu, and N. Wong, *A constructive algorithm for decomposing a tensor into a finite sum of orthonormal rank-1 terms*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 1315–1337.

[18] C. F. Beckmann and S. M. Smith, *Tensorial extensions of independent component analysis for multisubject fmri analysis*, Neuroimage, 25 (2005), pp. 294–311.

[19] G. Beylkin and M. J. Mohlenkamp, *Numerical operator calculus in higher dimensions*, Proceedings of the National Academy of Sciences, 99 (2002), pp. 10246–10251.

[20] G. Beylkin and M. J. Mohlenkamp, *Algorithms for numerical analysis in high dimensions*, SIAM J. Sci. Comput., 26 (2005), pp. 2133–2159.

[21] J. C. Bezdek and R. J. Hathaway, *Some notes on alternating optimization*, in Advances in Soft Computing — AFSS 2002: 2002 AFSS International Conference on Fuzzy Systems Calcutta, India,, N. R. Pal and M. Sugeno, eds., Springer, Berlin, Heidelberg, 2002, pp. 288–300.

[22] ——, *Convergence of alternating optimization*, Neural Parallel Sci. Comput., 11 (2003), pp. 351–368.

[23] D. Bini, M. Capovani, F. Romani, and G. Lotti, *$O(n^{2.7799})$ complexity for $n \times n$ approximate matrix multiplication*, Information Processing Letters, 8 (1979), pp. 234–235.

[24] J. Brachat, P. Comon, B. Mourrain, and E. Tsigaridas, *Symmetric tensor decomposition*, Linear Algebra Appl., 433 (2010), pp. 1851–1872.

[25] R. Bro, *Parafac. tutorial and applications*, Chemometrics and Intelligent Laboratory Systems, 38 (1997), pp. 149–171.

[26] R. Bro, *Multi-way analysis in the food industry. Models, algorithms, and applications.*, PhD thesis, University of Amsterdam, Amsterdam, The Netherlands, 1998.

[27] R. Bro, *Review on multiway analysis in chemistry—2002-2005*, Critical Reviews in Analytical Chemistry, 36 (2006), pp. 279–293.

[28] R. Bro and C. A. Andersson, *Improving the speed of multiway algorithms: Part ii: Compression*, Chemometrics and intelligent laboratory systems, 42 (1998), pp. 105–113.

[29] A. Bunse-Gerstner, R. Byers, V. Mehrmann, and N. K. Nichols, *Numerical computation of an analytic singular value decomposition of a matrix valued function*, Numer. Math., 60 (1991), pp. 1–40.

[30] A. C.A. and B. R., *Improving the speed of multi-way algorithms: - part i. tucker3*, Chemometrics and Intelligent Laboratory Systems, 42 (1998), pp. 93–103.

[31] J. A. Cadzow, *Signal enhancement: A composite property mapping algorithm*, IEEE Trans. on Acoust., Speech, Signal Processing, 36 (1988), pp. 39–62.

[32] J. D. Carroll and J.-J. Chang, *Analysis of individual differences in multidimensional scaling via an n-way generalization of Eckart-Young decomposition*, Psychometrika, 35 (1970), pp. 283–319.

[33] J. D. Carroll, S. Pruzansky, and J. B. Kruskal, *Candelinc: A general approach to multidimensional analysis of many-way arrays with linear constraints on parameters*, Psychometrika, 45 (1980), pp. 3–24.

[34] R. B. Cattell, *Parallel proportional profiles and other principles for determining the choice of factors by rotation*, Psychometrika, 9 (1944), pp. 267–283.

[35] ——, *The three basic factor-analytic research designs-their interrelations and derivatives*, Psychological bulletin, 49 (1952), p. 499.

[36] S. Celar, Z. Stojkic, Z. Seremet, Z. Marusic, and D. Zelenika, *Classification of test documents based on handwritten student id's characteristics*, Procedia Engineering, 100 (2015), pp. 782–790. 25th DAAAM International Symposium on Intelligent Manufacturing and Automation, 2014.

[37] B. Chen, S. He, Z. Li, and S. Zhang, *Maximum block improvement and polynomial optimization*, SIAM J. Optim., 22 (2012), pp. 87–107.

[38] B. Chen, A. P. Petropulu, and L. De Lathauwer, *Blind identification of convolutive mimo systems with 3 sources and 2 sensors*, EURASIP J. Appl. Signal Process., 2002 (2002), pp. 487–496.

[39] H. Chen and L. Qi, *Positive definiteness and semi-definiteness of even order symmetric cauchy tensors*, MANAGEMENT, 11 (2015), pp. 1263–1274.

[40] J. Chen and Y. Saad, *On the tensor SVD and the optimal low rank orthogonal approximation of tensors*, SIAM J. Matrix Anal. Appl., 30 (2008/09), pp. 1709–1734.

[41] Z. Chen and L. Qi, *Circulant tensors with applications to spectral hypergraph theory and stochastic process*, MANAGEMENT, 12 (2016), pp. 1227–1247.

[42] L. Cheng, Y.-C. Wu, and H. V. Poor, *Probabilistic tensor canonical polyadic decomposition with orthogonal factors*, IEEE Trans. Signal Process., 65 (2017), pp. 663–676.

[43] P. A. Chew, B. W. Bader, T. G. Kolda, and A. Abdelali, *Cross-language information retrieval using parafac2*, in Proceedings of the 13th ACM

SIGKDD international conference on Knowledge discovery and data mining, KDD '07, New York, NY, USA, 2007, ACM, pp. 143–152.

[44] R. Chill, *On the Łojasiewicz-Simon gradient inequality*, J. Funct. Anal., 201 (2003), pp. 572–601.

[45] M. T. Chu, R. E. Funderlic, and R. J. Plemmons, *Structured low rank approximation*, Linear Algebra Appl., 366 (2003), pp. 157–172. Special issue on structured matrices: analysis, algorithms and applications (Cortona, 2000).

[46] M. T. Chu, M. M. Lin, and L. Wang, *A study of singular spectrum analysis with global optimization techniques*, J. Global Optim., 60 (2014), pp. 551–574.

[47] M. T. Chu and N. T. Trendafilov, *The orthogonally constrained regression revisited*, J. Comput. Graph. Statist., 10 (2001), pp. 746–771.

[48] P. Comon, *Tensor decompositions: state of the art and applications*, in Mathematics in signal processing, V (Coventry, 2000), vol. 71 of Inst. Math. Appl. Conf. Ser. New Ser., Oxford Univ. Press, Oxford, 2002, pp. 1–24.

[49] ——, *Canonical tensor decompositions*, in Workshop on Tensor Decompositions, Palo Alto, CA, 2004.

[50] P. Comon, G. Golub, L.-H. Lim, and B. Mourrain, *Symmetric tensors and symmetric tensor rank*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1254–1279.

[51] ——, *Symmetric tensors and symmetric tensor rank*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1254–1279.

[52] P. Comon, X. Luciani, and A. L. De Almeida, *Tensor decompositions, alternating least squares and other tales*, J. Chemometrics, 23 (2009), pp. 393–405.

[53] F. Cong, Q.-H. Lin, L.-D. Kuang, X.-F. Gong, P. Astikainen, and T. Ristaniemi, *Tensor decomposition of eeg signals: A brief review*, Journal of Neuroscience Methods, 248 (2015), pp. 59–69.

[54] R. Coppi and S. Bolasco, *Multiway data analysis*, North-Holland Publishing Co., 1989.

[55] A. P. da Silva, P. Comon, and A. L. de Almeida, *A finite algorithm to compute rank-1 tensor approximations*, IEEE Signal Processing Letters, 23 (2016), pp. 959–963.

[56] M. Dana and K. D. Ikramov, *On the codimension of the variety of symmetric matrices with multiple eigenvalues*, J. Math. Sci. (N. Y.), 137 (2006), pp. 4780–4786.

[57] L. De Lathauwer and J. Castaing, *Tensor-based techniques for the blind separation of ds-cdma signals*, Signal Process., 87 (2007), pp. 322–336.

[58] ——, *Blind identification of underdetermined mixtures by simultaneous matrix diagonalization*, IEEE Trans. Signal Process., 56 (2008), pp. 1096–1105.

[59] L. De Lathauwer, J. Castaing, and J.-F. Cardoso, *Fourth-order cumulant-based blind identification of underdetermined mixtures*, IEEE Trans. Signal Process., 55 (2007), pp. 2965–2973.

[60] L. De Lathauwer and A. de Baynast, *Blind deconvolution of DS-CDMA signals by means of decomposition in rank-*$(1, L, L)$ *terms*, IEEE Trans. Signal Process., 56 (2008), pp. 1562–1571.

[61] L. De Lathauwer and B. De Moor, *From matrix to tensor: Multilinear algebra and signal processing*, in Institute of Mathematics and Its Applications Conference Series, vol. 67, Citeseer, 1998, pp. 1–16.

[62] L. De Lathauwer, B. De Moor, and J. Vandewalle, *A multilinear singular value decomposition*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1253–1278 (electronic).

[63] ——, *On the best rank-1 and rank-*$(r_1, r_2, \ldots, r_n)$ *approximation of higher-order tensors*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1324–1342.

[64] L. De Lathauwer and J. Vandewalle, *Dimensionality reduction in higher-order signal processing and rank-*$(R_1, R_2, \ldots, R_N)$ *reduction in multilinear algebra*, Linear Algebra Appl., 391 (2004), pp. 31–55.

[65] V. de Silva and L.-H. Lim, *Tensor rank and the ill-posedness of the best low-rank approximation problem*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1084–1127.

[66] M. De Vos, L. De Lathauwer, B. Vanrumste, S. Van Huffel, and W. Van Paesschen, *Canonical decomposition of ictal scalp EEG and accurate source localisation: principles and simulation study*, Intell. Neuroscience, 2007 (2007), pp. 11:1–11:8.

[67] M. De Vos, A. Vergult, L. De Lathauwer, W. De Clercq, S. Van Huffel, P. Dupont, A. Palmini, and W. Van Paesschen, *Canonical decomposition of ictal scalp eeg reliably detects the seizure onset zone*, NeuroImage, 37 (2007), pp. 844–854.

[68] W. S. DeSarbo and J. D. Carroll, *Three-way metric unfolding via alternating weighted least squares*, Psychometrika, 50 (1985), pp. 275–300.

[69] A. Doostan, G. Iaccarino, and N. Etemadi, *A least-squares approximation of high-dimensional uncertain systems*, Annual Research Briefs, Center for Turbulence Research, Stanford University, (2007), pp. 121–132.

[70] C. Eckart and G. Young, *The approximation of one matrix by another of lower rank*, Psychometrika, 1 (1936), pp. 211–218.

[71] L. Eldén and B. Savas, *A newton-grassmann method for computing the best multilinear rank-(r_1, r_2, r_3) approximation of a tensor*, SIAM J. Matrix Anal. Appl., 31 (2009), pp. 248–271.

[72] N. K. M. Faber, R. Bro, and P. K. Hopke, *Recent developments in candecomp/parafac algorithms: a critical review*, Chemometrics and Intelligent Laboratory Systems, 65 (2003), pp. 119–137.

[73] D. FITZGERALD, M. CRANITCH, and E. COYLE, *Non-negative tensor factorisation for sound source separation*, in IEE conference publication, Institution of Electrical Engineers, 2005, pp. 8–12.

[74] S. Friedland, *Best rank one approximation of real symmetric tensors can be chosen symmetric*, Front. Math. China, 8 (2013), pp. 19–40.

[75] S. Friedland, V. Mehrmann, R. Pajarola, and S. K. Suter, *On best rank one approximation of tensors*, Numer. Linear Algebra Appl., 20 (2013), pp. 942–955.

[76] S. Gourvénec, G. Tomasi, C. Durville, E. Di Crescenzo, C. Saby, D. Massart, R. Bro, and G. Oppenheim, *Cubatch, a matlab® interface for n-mode data analysis*, Chemometrics and Intelligent Laboratory Systems, 77 (2005), pp. 122–130.

[77] V. S. Grigorascu and P. A. Regalia, *Tensor displacement structures and polyspectral matching*, in Fast reliable algorithms for matrices with structure, SIAM, 1999, pp. 245–276.

[78] Y. Guan, M. T. Chu, and D. Chu, *Convergence analysis of an svd-based algorithm for the best rank-1 tensor approximation*, Linear Algebra Appl., 555 (2018), pp. 53–69.

[79] ——, *Svd-based algorithms for the best rank-1 approximation of a symmetric tensor*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 1095–1115.

[80] W. Hackbusch and B. N. Khoromskij, *Tensor-product approximation to operators and functions in high dimensions*, J. Complexity, 23 (2007), pp. 697–714.

[81] W. Hackbusch, B. N. Khoromskij, and E. E. Tyrtyshnikov, *Hierarchical kronecker tensor-product approximations*, Journal of Numerical Mathematics, 13 (2005), pp. 119–156.

[82] R. A. Harshman, *Foundations of the parafac procedure: Models and conditions for an "explanatory" multi-modal factor analysis*, UCLA Working Papers in Phonetics, 16 (1970), pp. 1–84.

[83] R. A. Harshman, *PARAFAC2: Mathematical and technical notes*, UCLA Working Papers in Phonetics, 22 (1972b), pp. 30–44.

[84] R. A. Harshman, *Models for analysis of asymmetrical relationships among n objects or stimuli*, in First Joint Meeting of the Psychometric Society and the Society of Mathematical Psychology, Hamilton, Ontario, 1978.

[85] R. A. Harshman and M. E. Lundy, *Uniqueness proof for a family of models sharing features of tucker's three-mode factor analysis and parafac/candecomp*, Psychometrika, 61 (1996), pp. 133–154.

[86] J. Håstad, *Tensor rank is NP-complete*, J. Algorithms, 11 (1990), pp. 644–654.

[87] T. Hazan, S. Polak, and A. Shashua, *Sparse image coding using a 3d non-negative tensor factorization*, in Computer Vision. ICCV 2005. Tenth IEEE International Conference on, vol. 1, Washington, DC, USA, IEEE Computer Society, pp. 50–57.

[88] R. Henrion, *N-way principal component analysis theory, algorithms and applications*, Chemometrics and Intelligent Laboratory Systems, 25 (1994), pp. 1–23.

[89] N. J. Higham, *Computing the polar decomposition—with applications*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 1160–1174.

[90] ——, *Computing a nearest symmetric positive semidefinite matrix*, Linear Algebra Appl., 103 (1988), pp. 103–118.

[91] C. J. Hillar and L.-H. Lim, *Most tensor problems are np-hard*, J. ACM), 60 (2013), pp. 45:1–45:39.

[92] F. L. Hitchcock, *The expression of a tensor or a polyadic as a sum of products*, Journal of Mathematics and Physics, 6 (1927), pp. 164–189.

[93] ——, *Multiple invariants and generalized rank of a p-way matrix or tensor*, Journal of Mathematics and Physics, 7 (1928), pp. 39–79.

[94] I. Ibraghimov, *Application of the three-way decomposition for matrix compression*, Numer. Linear Algebra Appl., 9 (2002), pp. 551–565.

[95] M. Ishteva, P.-A. Absil, and P. Van Dooren, *Jacobi algorithm for the best low multilinear rank approximation of symmetric tensors*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 651–672.

[96] M. Ishteva, P.-A. Absil, S. Van Huffel, and L. De Lathauwer, *Best low multilinear rank approximation of higher-order tensors, based on the riemannian trust-region scheme*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 115–135.

[97] M. Ishteva, L. De Lathauwer, P.-A. Absil, and S. Van Huffel, *Differential-geometric newton method for the best rank-(r_1, r_2, r_3) approximation of tensors*, Numer. Algorithms, 51 (2009), pp. 179–194.

[98] J.-h. Jiang, H.-l. Wu, Y. Li, and R.-q. Yu, *Three-way data resolution by alternating slice-wise diagonalization (asd) method*, Journal of Chemometrics, 14 (2000), pp. 15–36.

[99] Y.-L. Jiang and X. Kong, *On the uniqueness and perturbation to the best rank-one approximation of a tensor*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 775–792.

[100] J. Kamm and J. G. Nagy, *Optimal Kronecker product approximation of block Toeplitz matrices*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 155–172.

[101] A. Kapteyn, H. Neudecker, and T. Wansbeek, *An approach to n -mode components analysis*, Psychometrika, 51 (1986), pp. 269–275.

[102] J. B. Keller, *Closest unitary, orthogonal and hermitian operators to a given operator*, Math. Mag., 48 (1975), pp. 192–197.

[103] B. N. Khoromskij, *Structured rank-*$(R_1, \ldots, R_D)$ *decomposition of function-related tensors in* $\mathbb{R}^D$, Comput. Methods Appl. Math., 6 (2006), pp. 194–220 (electronic).

[104] B. N. Khoromskij and V. Khoromskaia, *Low rank Tucker-type tensor approximation to classical potentials*, Cent. Eur. J. Math., 5 (2007), pp. 523–550.

[105] B. N. Khoromskij, V. Khoromskaia, S. R. Chinnamsetty, and H.-J. Flad, *Tensor decomposition in electronic structure calculations on 3D Cartesian grids*, J. Comput. Phys., 228 (2009), pp. 5749–5762.

[106] H. A. Kiers and I. V. Mechelen, *Three-way component analysis: Principles and illustrative application*, Psychological methods, 6 (2001), pp. 84–110.

[107] H. A. L. Kiers, *Towards a standardized notation and terminology in multiway analysis*, Journal of Chemometrics, 14 (2000), pp. 105–122.

[108] D. E. Knuth, *The art of computer programming, 2: seminumerical algorithms, addision wesley*, Reading, MA, (1998).

[109] E. Kofidis and P. A. Regalia, *On the best rank-1 approximation of higher-order supersymmetric tensors*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 863–884.

[110] T. G. Kolda, *Orthogonal tensor decompositions*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 243–255 (electronic).

[111] ——, *A counterexample to the possibility of an extension of the eckart-young low-rank approximation theorem for the orthogonal rank tensor decomposition*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 762–767.

[112] ——, *Numerical optimization for symmetric tensor decomposition*, Math. Program., 151 (2015), pp. 225–248.

[113] T. G. Kolda and B. W. Bader, *The tophits model for higher-order web link analysis*, in Workshop on link analysis, counterterrorism and security, vol. 7, 2006, pp. 26–29.

[114] ——, *Tensor decompositions and applications*, SIAM Rev., 51 (2009), pp. 455–500.

[115] T. G. Kolda, B. W. Bader, and J. P. Kenny, *Higher-order web link analysis using multilinear algebra*, in Proceedings of the Fifth IEEE International Conference on Data Mining, ICDM '05, Washington, DC, USA, 2005, IEEE Computer Society, pp. 242–249.

[116] W. P. Krijnen, T. K. Dijkstra, and A. Stegeman, *On the non-existence of optimal solutions and the occurrence of "degeneracy" in the CANDECOMP/PARAFAC model*, Psychometrika, 73 (2008), pp. 431–439.

[117] W. P. Krijnen and J. M. Ten Berge, *A constrained parafac method for positive manifold data*, Applied Psychological Measurement, 16 (1992), pp. 295–305.

[118] P. M. Kroonenberg, *Three-mode Principal Component Analysis: Theory and Applications*, M & T series, DSWO Press, 1983.

[119] P. M. Kroonenberg, *Applied Multiway Data Analysis*, Wiley Series in Probability and Statistics, John Wiley & Sons, 2008.

[120] P. r. Kroonenberg and J. Leeuw, *Principal component analysis of three-mode data by means of alternating least squares algorithms*, Psychometrika, 45 (1980), pp. 69–97.

[121] J. B. Kruskal, *Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics*, Linear Algebra Appl., 18 (1977), pp. 95–138.

[122] W. Landry, *Implementing a high performance tensor library*, Sci. Program., 11 (2003), pp. 273–290.

[123] J. M. Landsberg, *Tensors: geometry and applications*, Representation theory, 381 (2012), p. 402.

[124] A. N. Langville and W. J. Stewart, *A kronecker product approximate preconditioner for sans*, Numer. Linear Algebra Appl., 11 (2004), pp. 723–752.

[125] C. L. Lawson and R. J. Hanson, *Solving least squares problems*, vol. 15 of Classics in Applied Mathematics, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1995.

[126] D. Leibovici and R. Sabatier, *A singular value decomposition of a k-way array for a principal component analysis of multiway data, pta-k*, Linear Algebra Appl., 269 (1998), pp. 307–329.

[127] S. Leurgans and R. T. Ross, *Multilinear models: Applications in spectroscopy*, Statistical Science, 7 (1992), pp. 289–310.

[128] J. Levin, *Three-mode Factor Analysis*, Project on techniques for investigation of structure of individual differences in psychological phenomena, University of Illinois at Urbana-Champaign, 1963.

[129] W. Li, D. Liu, M. K. Ng, and S.-W. Vong, *The uniqueness of multilinear pagerank vectors*, Numer. Linear Algebra Appl., 24 (2017), p. e2107.

[130] L.-H. Lim, *Singular values and eigenvalues of tensors: a variational approach*, in Computational Advances in Multi-Sensor Adaptive Processing, 2005 1st IEEE International Workshop on, dec. 2005, pp. 129–132.

[131] N. Liu, B. Zhang, J. Yan, Z. Chen, W. Liu, F. Bai, and L. Chien, *Text representation: From vector to tensor*, in Proceedings of the Fifth IEEE International Conference on Data Mining, ICDM '05, Washington, DC, USA, 2005, IEEE Computer Society, pp. 725–728.

[132] S. Łojasiewicz, *Une propriété topologique des sous-ensembles analytiques réels*, Les équations aux dérivées partielles, 117 (1963), pp. 87–89.

[133] S. Łojasiewicz and M.-A. Zurro, *On the gradient inequality*, Bull. Polish Acad. Sci. Math., 47 (1999), pp. 143–145.

[134] M. W. Mahoney, M. Maggioni, and P. Drineas, *Tensor-CUR decompositions for tensor-based data*, in Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, 2006, ACM, pp. 327–336.

[135] C. D. M. Martin and C. F. Van Loan, *A jacobi-type method for computing orthogonal tensor decompositions*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1219–1232.

[136] E. Martinez-Montes, P. A. Valdés-Sosa, F. Miwakeichi, R. I. Goldman, and M. S. Cohen, *Concurrent eeg/fmri analysis by multiway partial least squares*, NeuroImage, 22 (2004), pp. 1023–1034.

[137] F. Miwakeichi, E. Martinez-Montes, P. A. Valdés-Sosa, N. Nishiyama, H. Mizuhara, and Y. Yamaguchi, *Decomposing eeg data into space–time–frequency components using parallel factor analysis*, NeuroImage, 22 (2004), pp. 1035–1045.

[138] J. Mocks, *Topographic components model for event-related potentials and some biophysical considerations*, Biomedical Engineering, IEEE Transactions on, 35 (1988), pp. 482–484.

[139] M. J. Mohlenkamp, *Musings on multilinear fitting*, Linear Algebra Appl., 438 (2013), pp. 834–852.

[140] J. J. Moré and D. C. Sorensen, *Computing a trust region step*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 553–572.

[141] M. Mørup, L. K. Hansen, and S. M. Arnfred, *Erpwavelab: A toolbox for multi-channel analysis of time–frequency transformed event related potentials*, Journal of neuroscience methods, 161 (2007), pp. 361–368.

[142] M. Mørup, L. K. Hansen, and S. M. Arnfred, *Algorithms for sparse non-negative TUCKER*, Neural Comput., (2008), pp. 2112–2131.

[143] M. Mørup, L. K. Hansen, C. S. Herrmann, J. Parnas, and S. M. Arnfred, *Parallel factor analysis as an exploratory tool for wavelet transformed event-related eeg*, NeuroImage, 29 (2006), pp. 938–947.

[144] M. Mørup, L. K. Hansen, J. Parnas, and S. M. Arnfred, *Decomposing the time-frequency representation of eeg using non-negative matrix and multi-way factorization*, Technical University of Denmark Technical Report, (2006).

[145] D. Muti and S. Bourennane, *Multidimensional filtering based on a tensor approach*, Signal Process., 85 (2005), pp. 2338–2353.

[146] J. Nagy and M. Kilmer, *Kronecker product approximation for preconditioning in three-dimensional imaging applications*, Image Processing, IEEE Transactions on, 15 (2006), pp. 604–613.

[147] C. Navasca, L. De Lathauwer, and S. Kindermann, *Swamp reducing technique for tensor decomposition*, in Proc. 16th European Signal Processing Conference (EUSIPCO 2008), Lausanne, Switzerland, Aug. 25–29 2008.

[148] G. Ni and Y. Wang, *On the best rank-1 approximation to higher-order symmetric tensors*, Math. Comput. Modelling, 46 (2007), pp. 1345–1352.

[149] D. Nion and L. D. Lathauwer, *An enhanced line search scheme for complex-valued tensor decompositions. application in ds-cdma*, Signal Processing, 88 (2008), pp. 749–755.

[150] J. M. Ortega, *Numerical analysis*, vol. 3 of Classics in Applied Mathematics, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second ed., 1990. A second course.

[151] P. Paatero, *A weighted non-negative least squares algorithm for three-way "parafac" factor analysis*, Chemometrics and Intelligent Laboratory Systems, 38 (1997), pp. 223–242.

[152] P. Paatero, *The multilinear engine: A table-driven, least squares program for solving multilinear problems, including the n-way parallel factor analysis model*, Journal of Computational and Graphical Statistics, 8 (1999), pp. 854–888.

[153] ——, *Construction and analysis of degenerate parafac models*, Journal of Chemometrics: A Journal of the Chemometrics Society, 14 (2000), pp. 285–299.

[154] L. Qi, *The best rank-one approximation ratio of a tensor space*, SIAM J. Matrix Analysis Applications, (2011), pp. 430–442.

[155] L. Qi, W. Sun, and Y. Wang, *Numerical multilinear algebra and its applications*, Front. Math. China, 2 (2007), pp. 501–526.

[156] M. Rajih, P. Comon, and R. A. Harshman, *Enhanced line search: A novel method to accelerate parafac*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1128–1147.

[157] P. A. Regalia, *Monotonically convergent algorithms for symmetric tensor approximation*, Linear Algebra Appl., 438 (2013), pp. 875–890.

[158] B. Savas, *Analyses and tests of handwritten digit recognition algorithms*, LiTH-MAT-EX-2003-01, Linköping University, Department of Mathematics, (2003).

[159] A. Shashua and T. Hazan, *Non-negative tensor factorization with applications to statistics and computer vision*, in Proceedings of the 22nd international conference on Machine learning, ICML '05, New York, NY, USA, 2005, ACM, pp. 792–799.

[160] N. Sidiropoulos, R. Bro, and G. Giannakis, *Parallel factor analysis in sensor array processing*, Signal Processing, IEEE Transactions on, 48 (2000), pp. 2377–2388.

[161] N. Sidiropoulos and R. Budampati, *Khatri-rao space-time codes*, Signal Processing, IEEE Transactions on, 50 (2002), pp. 2396–2407.

[162] N. Sidiropoulos, G. Giannakis, and R. Bro, *Blind parafac receivers for ds-cdma systems*, Signal Processing, IEEE Transactions on, 48 (2000), pp. 810–823.

[163] A. Smilde, R. Bro, and P. Geladi, *Multi-way Analysis with Applications in the Chemical Sciences*, J. Wiley, 2004.

[164] A. J. Sommese and C. W. Wampler, II, *The numerical solution of systems of polynomials arising in engineering and science*, World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2005.

[165] Y. Song and L. Qi, *Infinite and finite dimensional hilbert tensors*, Linear Algebra Appl., 451 (2014), pp. 1–14.

[166] M. Sørensen, L. D. Lathauwer, P. Comon, S. Icart, and L. Deneire, *Canonical polyadic decomposition with a columnwise orthonormal factor matrix*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 1190–1213.

[167] A. Stegeman, *Degeneracy in candecomp/parafac explained for $p{\times}p{\times}2$ arrays of rank $p+1$ or higher*, Psychometrika, 71 (2006), pp. 483–501.

[168] ——, *Degeneracy in candecomp/parafac and indscal explained for several three-sliced arrays with a two-valued typical rank*, Psychometrika, 72 (2007), pp. 601–619.

[169] ——, *Low-rank approximation of generic $p \times q \times 2$ arrays and diverging components in the Candecomp/Parafac model*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 988–1007.

[170] J. Sun, S. Papadimitriou, and P. S. Yu, *Window-based tensor analysis on high-dimensional and multi-aspect streams*, in Proceedings of the Sixth International Conference on Data Mining, ICDM '06, Washington, DC, USA, 2006, IEEE Computer Society, pp. 1076–1080.

[171] J. Sun, D. Tao, and C. Faloutsos, *Beyond streams and graphs: dynamic tensor analysis*, in Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '06, New York, NY, USA, 2006, ACM, pp. 374–383.

[172] J.-T. Sun, H.-J. Zeng, H. Liu, Y. Lu, and Z. Chen, *Cubesvd: a novel approach to personalized web search*, in Proceedings of the 14th international

conference on World Wide Web, WWW '05, New York, NY, USA, 2005, ACM, pp. 382–390.

[173] J. M. F. ten Berge, *Least squares optimization in multivariate analysis*, vol. 25 of M&T Series, D.S.W.O. Press, Leiden, 1993.

[174] G. Tomasi, *Use of the properties of the khatri-rao product for the computation of jacobian, hessian, and gradient of the parafac model under matlab*, Private communication, (2005).

[175] G. Tomasi, *Practical and Computational Aspects in Chemometric Data Analysis: Ph.D. Dissertation*, Department of Food Science, Royal Veterinary and Agricultural University, 2006.

[176] L. R. Tucker, *Implications of factor analysis of three-way matrices for measurement of change*, Problems in measuring change, 15 (1963), pp. 122–137.

[177] ——, *The extension of factor analysis to three-dimensional matrices*, Contributions to mathematical psychology, (1964), pp. 110–127.

[178] ——, *Some mathematical notes on three-mode factor analysis*, Psychometrika, 31 (1966), pp. 279–311.

[179] A. Uschmajew, *Local convergence of the alternating least squares algorithm for canonical tensor approximation*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 639–652.

[180] C. F. Van Loan, *The ubiquitous Kronecker product*, J. Comput. Appl. Math., 123 (2000), pp. 85–100.

[181] C. F. Van Loan and N. Pitsianis, *Approximation with kronecker products*, in Linear algebra for large scale and real-time applications, Springer, 1993, pp. 293–314.

[182] N. Vannieuwenhoven, R. Vandebril, and K. Meerbergen, *A new truncation strategy for the higher-order singular value decomposition*, SIAM J. Sci. Comput., 34 (2012), pp. A1027–A1052.

[183] M. Vasilescu and D. Terzopoulos, *Multilinear image analysis for facial recognition*, in Pattern Recognition. Proceedings. 16th International Conference on, vol. 2, 2002, pp. 511–514.

[184] M. A. O. Vasilescu and D. Terzopoulos, *Multilinear analysis of image ensembles: Tensorfaces*, in Proceedings of the 7th European Conference on Computer Vision-Part I, ECCV'02, London, UK, 2002, Springer-Verlag, pp. 447–460.

[185] ——, *Multilinear subspace analysis of image ensembles*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 2 of CVPR'03, Madison, WI, 2003, pp. 93–99.

[186] D. Vlasic, M. Brand, H. Pfister, and J. Popović, *Face transfer with multilinear models*, ACM Trans. Graph., 24 (2005), pp. 426–433.

[187] H. Wang and N. Ahuja, *Facial expression decomposition*, in Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2, ICCV '03, Washington, DC, USA, 2003, IEEE Computer Society, pp. 958–965.

[188] ——, *Compact representation of multidimensional data using tensor rank-one decomposition*, in Pattern Recognition, ICPR 2004. Proceedings of the 17th International Conference on, vol. 1, IEEE, 2004, pp. 44–47.

[189] L. Wang and M. T. Chu, *On the global convergence of the alternating least squares method for rank-one approximation to generic tensors*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 1058–1072.

[190] L. Wang, M. T. Chu, and B. Yu, *Orthogonal low rank tensor approximation: alternating least squares method and its global convergence*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 1–19.

[191] B. M. Wise, N. Gallagher, R. Bro, J. Shaver, W. Windig, and R. S. Koch, *Pls toolbox 4.0*, Eigenvector Research Incorporated: Wenatchee, WA, USA, (2007).

[192] K. Wright, *Differential equations for the analytic singular value decomposition of a matrix*, Numer. Math., 3 (1992), pp. 283–295.

[193] Y. Yang, S. Hu, L. De Lathauwer, and J. A. Suykens, *Convergence study of block singular value maximization methods for rank-1 approximation to higher order tensors*, tech. rep., Internal Report 16-149, ESAT-SISTA, KU Leuven, 2016.

[194] E. Zander and H. Matthies, *Tensor product methods for stochastic problems*, in PAMM: Proceedings in Applied Mathematics and Mechanics, vol. 7, Wiley Online Library, 2007, pp. 2040067–2040068.

[195] R. Zass, *Huji tensor library*, 2006.

[196] E. Zhang, J. Hays, and G. Turk, *Interactive tensor field design and visualization on surfaces*, IEEE Transactions on Visualization and Computer Graphics, 13 (2007), pp. 94–107.

[197] T. Zhang and G. H. Golub, *Rank-one approximation to high order tensors*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 534–550 (electronic).

[198] X. Zhang, C. Ling, and L. Qi, *The best rank-1 approximation of a symmetric tensor and related spherical optimization problems*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 806–821.