Subspace Power Method for Symmetric Tensor Decomposition

Joe Kileel†
João M. Pereira‡

Abstract
We introduce the Subspace Power Method (SPM) for calculating the CP decomposition of low-rank real symmetric tensors. This algorithm calculates one new CP component at a time, alternating between applying the shifted symmetric higher-order power method (SS-HOPM) to a certain modified tensor, constructed from a matrix flattening of the original tensor; and using appropriate deflation steps. We obtain rigorous guarantees for SPM regarding convergence and global optima for input tensors of dimension $d$ and order $m$ of rank up to $O(d^{\lfloor m/2 \rfloor})$, via results in classical algebraic geometry and optimization theory. As a by-product of our analysis we prove that SS-HOPM converges unconditionally, settling a conjecture in [36]. Numerical experiments demonstrate that SPM is roughly one order of magnitude faster than state-of-the-art CP decomposition algorithms at moderate ranks. Furthermore, prior knowledge of the CP rank is not required by SPM.

Keywords: CP decomposition, symmetric tensor, power method, rank-one update, trisecant lemma, convergence analysis

1 Introduction

A tensor is a multi-dimensional array [35]. A symmetric tensor is an array unchanged by permutation of indices. That is, $\mathbf{T}$ of order $m$ is symmetric if for each index $(j_1, \ldots, j_m)$ and permutation $\sigma$ it holds $\mathbf{T}_{j_1, \ldots, j_m} = \mathbf{T}_{\sigma(j_1), \ldots, \sigma(j_m)}$.

Symmetric tensors arise naturally in many data processing applications. They occur as higher order moments of a dataset, generalizing the mean and covariance of a random vector; as derivatives of multivariate real-valued functions, generalizing the gradient and Hessian of a function; and as adjacency tensors for hypergraphs, generalizing the adjacency matrix of graphs. Being able to decompose symmetric tensors is critical in domains including blind source separation [12, 19], independent component analysis [30, 51], antenna array processing [20, 15], telecommunications [59, 14], psychometrics [13], chemometrics [10], magnetic resonance imaging [4] and latent variable estimation for Gaussian mixture models [22, 49], topic models and hidden Markov models [2].

The present paper presents a new algorithm for computing the celebrated real symmetric CP decomposition:

$$\mathbf{T} = \sum_{i=1}^{r} \lambda_i \mathbf{a}_i^{\otimes m}. \quad (1)$$
In (1), the left-hand side is the given input, a real symmetric tensor \( T \) of size \( d \times \ldots \times d \) \((m \text{ times})\). The task is to compute the right-hand side, where \( r \) is an integer, \( \lambda_i \in \mathbb{R} \) are scalars, \( a_i \in \mathbb{R}^d \) are unit-norm vectors, and \( a_i^{\otimes m} \) denotes the \( m \)-th tensor power (or outer product) of \( a_i \), that is,

\[
(a_i^{\otimes m})_{j_1 \ldots j_m} = (a_i)_{j_1} \ldots (a_i)_{j_m}.
\]

For any tensor \( T \), there exists an expression of type (1), since \( \{a_i^{\otimes m} : \|a\| = 1\} \) spans the vector space of all symmetric tensors \([38]\). Thus one requires \( r \) to be minimal in (1), and declares this integer to be the CP rank of \( T \). It is a crucial fact that, generically, CP decompositions are unique for rank-deficient tensors: if \( T \) satisfies (1) with \( r < \frac{1}{2}(d^{m-1}) = \mathcal{O}(d^{m-1}), \ d \geq 6 \) and \( (\lambda_i, a_i) \) are Zariski-generic, then the rank of \( T \) is indeed \( r \) and the minimal decomposition (1) is unique (up to permutation and sign flips of \( a_i \)). Generic uniqueness is a result in algebraic geometry \([17; 3; 16]\).

While low-rank symmetric CP rank decompositions exist and are generically unique, computing them is another matter. Hillar-Lim showed tensor decomposition is NP-hard in general \([28]\). Nonetheless, a number of works have sought efficient algorithms for sufficiently low-rank tensors, e.g., \([26; 19; 2; 34; 23; 43; 29; 46; 32]\). Conjecturally in theoretical computer science, there exist efficient algorithms that succeed with high probability in decomposing random tensors of rank on the order of the square root of the number of tensor entries, but not so for ranks substantially more \([61]\). From a numerical linear algebra standpoint, producing practically efficient methods – even with restricted rank – is a further challenge.

1.1 Our contributions

In this paper, we develop a numerical algorithm that accepts \( T \) as input, and aims to output the minimal decomposition (1), up to trivial ambiguities, provided

\[
\begin{cases}
r \leq \left( \frac{d+n-2}{n-1} \right) & \text{if } m = 2n - 1, \\
r \leq \left( \frac{d+n-1}{n} \right) - d & \text{if } m = 2n.
\end{cases}
\]

In devising the method, we assume an exactly low-rank decomposition exists, though simple adjustments allow the method to run on noisy tensors. Notably, the new algorithm outperforms existing state-of-art methods by roughly one order of magnitude in terms of speed for ground-truth tensors of moderate rank. It also does not require knowledge of \( r \) in advance, and instead estimates the rank. Thirdly, the method is robust to additive noise.

We call the method the Subspace Power Method (SPM). To give a glimpse, it consists of three parts.

(A) **Extract Subspace:** We flatten \( T \) to a matrix, and compute its singular vector decomposition. From this, we extract the subspace of order-\( n \) tensors spanned by \( a_i^{\otimes n}, i = 1, \ldots, r \), where \( n = \lceil \frac{m}{2} \rceil \), denoted by \( A \).

(B) **Power Method:** We seek one rank-1 point \( a_i^{\otimes n} \) in the subspace \( A \). For this end, SS-HOPM \([35]\) for computing tensor eigenvectors is applied to an appropriately modified tensor, constructed using \( A \).

(C) **Deflation:** We solve for the corresponding scalar \( \lambda_i \), and update the low-rank matrix factorization to be that of the flattening of \( T - \lambda_i a_i^{\otimes m} \).
The input is the symmetric tensor $T$ (the low-rank decomposition of $T$ is unknown). The output is $(\lambda_i, a_i)_{i=1}^{r}$. SPM has three steps: (A) **Extract Subspace**, (B) **Power Method**, (C) **Deflate**.

The pipeline repeats $r$ times, until all $(\lambda_i, a_i)$ are recovered. Figure 1 shows a schematic of SPM. See Algorithm 1 in Section 3 for full details.

The paper’s other contribution is that we prove various theoretical guarantees. These come in two flavors. Firstly, we characterize the *global optima* in the reformulation of tensor decomposition used here. Specifically in Propositions 3.1 and 3.2, we use the trisecant lemma from algebraic geometry to show that the only rank-1 points in the subspace $A$ (from Step A above) are the CP components tensored-up, i.e., $a_i^{\otimes n}$ (up to scale). Secondly, we establish *convergence guarantees* for the Power Method iteration (Step B above). This analysis is summarized by Theorem 4.1. In particular, using the Łojasiewicz inequality, we prove Power Method converges from any initialization. In fact this is an important technical contribution: while proving it, we also settle a conjecture of Kolda-Mayo that their SS-HOPM method for computing Z-eigenvectors of symmetric tensors always converges (see page 1107 of [36]). Previous convergence results applied to generic tensors [36] or to certain applications [58]. Qualitative bounds are also obtained on the rate of convergence for Step B. Further, we prove that Power Method converges to a second-order optimal point for almost all initializations, and each CP component vector $\pm a_i$ is an attractive fixed point.

1.2 Comparison to prior art

SPM (Algorithm 1) integrates various ideas in the literature into a single natural algorithm for symmetric tensor decomposition, with innovations in computational efficiency and convergence theory.

Step A of SPM is a variation of the classical Catalecticant Method [31]. This goes back to Sylvester of the 19th century [57]. Although Sylvester’s work is quite well-known, SPM is, to our knowledge, the first efficient numerical method for tensor decomposition based on this formulation of tensor decomposition. The work [8] proposed using symbolic techniques with Sylvester’s Catalecticant Method to decompose symmetric tensors, but that algorithm appears slow already on modest-sized examples. Other related works are [6] [32], which generalize [19] and enjoy strong theoretical guarantees, but these lack an implementation or any numerical demonstrations at present. Further algebraic works are [31] and [47]. Perhaps most related to ours is the latter by Oeding-Ottaviani. Importantly however, we differ in that [47] proposes using standard polynomial-solving techniques, e.g., Gröbner bases, to compute $(\lambda_i, a_i)$ after reformulating tensor decomposition via Sylvester’s Catalecticant Method. Unfortunately, a Gröbner basis approach is impractical already for small-sized problem dimensions, since Gröbner bases have doubly exponential running time, and require exact arithmetic. By contrast, SPM consists of fast numerical iterations and optimized...
Next, Step B of SPM connects with Kolda-Mayo’s SS-HOPM method for computing tensor eigenvectors \[36\]. Step B may be viewed as the higher-order power method applied to a different symmetric tensor \(\tilde{T}\), constructed from the subspace extracted in Step A, see \[29\]. SPM identifies the \(Z\)-eigenvectors of \(\tilde{T}\) (computable by SS-HOPM) with the CP tensor components \(a_i\) of \(T\). This is important because the power method applied directly to \(T\) does not recover the CP components of \(T\). In analyzing the convergence of Step B we settle a conjecture of \[36\] on SS-HOPM in general.

Finally, Step C of SPM on deflation is related to Wedderburn’s rank reduction formula \[18\]. For maximal efficiency, we derive an optimized implementation of the procedure using Householder reflections, to avoid recomputation as much as possible.

In other respects, Algorithm 1 bears resemblance to De Lathauwer et al.’s Fourth-Order-Only Blind Identification (FOOBI) algorithm for fourth-order symmetric tensor decomposition \[19\]. It is also related to the methods for finding low-rank elements in subspaces of matrices in \[51, 21, 44\], and the asymmetric decomposition method in \[50\] repeatedly reducing the tensor length in given directions.

For third-order tensors, the simultaneous diagonalization algorithm \[40\] (often attributed to Jennrich) is an efficient and provable decomposition method for tensors of size \(d \times d \times d\) and rank less than or equal to \(d\). It uses matrix eigendecompositions in a straightforward manner. For higher-order tensors of order \(m \geq 4\), one may flatten into a third-order tensor and apply simultaneous diagonalization provided the rank is \(O(d^{(m-1)/2})\). Nevertheless, the algorithm is known to suffer from numerical instabilities \[33\], motivating the study of methods that are more stable to noise.

As far as leading practical algorithms go, many existing ones use direct nonconvex optimization, attempting to minimize the squared residual

\[
\|T - \sum_{i=1}^{r} \lambda_i a_i \otimes m\|_2^2.
\]

This is performed, e.g., by gradient descent or symmetric variants of alternating least squares \[34\]. In Section 5 we compare SPM to state-of-the-art numerical methods, in particular to the Matlab package Tensorlab \[60\]. The comparison is done using standard random ensembles for symmetric tensors, as well as “worse-conditioned” tensors with correlated components. We also compare to quite different but state-of-the-art theoretical methods: the method of generating polynomials \[45, 46\] and FOOBI \[19, 11\].

### 1.3 Organization of the paper

The paper is organized as follows. Section 2 establishes notation and basic definitions. Section 3 details the SPM algorithm. Section 4 analyzes the convergence of the power iteration (Step B in Figure 1). Section 5 presents numerics, with comparisons of runtime and noise sensitivity to other methods. Section 6 concludes. The appendices contain certain technical proofs. Matlab and Python code for SPM is available at https://www.github.com/joampereira/SPM
2 Definitions and Notation

2.1 Tensors and tensor products

Let $\mathcal{T}_d^m = (\mathbb{R}^d)^{\otimes m} \cong \mathbb{R}_d^m$ denote the vector space of real tensors of order $m$ and length $d$ in each mode. It is a Euclidean space, with the Frobenius inner product and norm. If $\mathcal{T} \in \mathcal{T}_d^m$, then $t_{j_1 \ldots j_m}$ is the entry indexed by $(j_1, \ldots, j_m) \in [d]^m$ where $[d] = \{1, \ldots, d\}$. For tensors $\mathcal{T} \in \mathcal{T}_d^m$ and $\mathcal{U} \in \mathcal{T}_d^{n}$, their tensor product in $\mathcal{T}_d^{m+n}$ is

$$(\mathcal{T} \otimes \mathcal{U})_{j_1 \ldots j_{m+n}} = t_{j_1 \ldots j_m} u_{j_{m+1} \ldots j_{m+n}} \quad \forall (j_1, \ldots, j_{m+n}) \in [d]^{m+n}.$$  

The tensor power $\mathcal{T}^{\otimes p} \in \mathcal{T}_d^{pm}$ is the tensor product of $\mathcal{T}$ with itself $d$ times. The tensor dot product (or tensor contraction) between $\mathcal{T} \in \mathcal{T}_d^m$ and $\mathcal{U} \in \mathcal{T}_d^m$, with $m \geq n$, is the tensor in $\mathcal{T}_d^{m-n}$ defined by

$$(\mathcal{T} \cdot \mathcal{U})_{j_{m+1} \ldots j_m} = \sum_{j_1=1}^d \cdots \sum_{j_n=1}^d t_{j_1 \ldots j_m} u_{j_1 \ldots j_m}.$$  

If $m = n$, contraction coincides with the inner product, i.e., $\mathcal{T} \cdot \mathcal{U} = \langle \mathcal{T}, \mathcal{U} \rangle$. For $\mathcal{T} \in \mathcal{T}_d^m$, a (real normalized) Z-eigenvector/eigenvalue pair $(v, \lambda) \in \mathbb{R}^d \times \mathbb{R}$ is a vector/scalar pair satisfying $\mathcal{T} \cdot v^{\otimes (m-1)} = \lambda v$ and $v \in S^{d-1}$, see [41,52]. Here $S^{d-1}$ denotes the unit-sphere in $\mathbb{R}^d$ with respect to the Euclidean norm $|| \cdot || := || \cdot ||_2$.

There is a useful relation between inner and outer products of tensors.

**Lemma 2.1** (Inner product of tensor products [24]). For tensors $\mathcal{T}_1, \mathcal{T}_2 \in \mathcal{T}_d^m$, $\mathcal{U}_1, \mathcal{U}_2 \in \mathcal{T}_d^{n}$, we have

$$\langle \mathcal{T}_1 \otimes \mathcal{U}_1, \mathcal{T}_2 \otimes \mathcal{U}_2 \rangle = \langle \mathcal{T}_1, \mathcal{W}_2 \rangle \langle \mathcal{U}_1, \mathcal{U}_2 \rangle.$$  

In particular, for vectors $u, v \in \mathbb{R}^d$, we have $\langle v^{\otimes m}, u^{\otimes m} \rangle = \langle v, u \rangle^m$.

2.2 Symmetric tensors

**Definition 2.2.** A tensor $\mathcal{T} \in \mathcal{T}_d^m$ is symmetric if it is unchanged by any permutation of indices, that is,

$$t_{j_1 \ldots j_m} = t_{j_{\sigma(1)} \ldots j_{\sigma(m)}} \quad \forall (j_1, \ldots, j_m) \in [d]^m \quad \text{and} \quad \sigma \in \Pi^m,$$  

where $\Pi^m$ is the permutation group on $[m]$. We denote by $S_d^m$ the vector space of real symmetric tensors of order $m$ and length $d$. A tensor $\mathcal{T} \in \mathcal{T}_d^m$ may be symmetrized by $\text{sym} : \mathcal{T}_d^m \rightarrow S_d^m$ defined as

$$\text{sym}(\mathcal{T})_{j_1 \ldots j_m} = \frac{1}{m!} \sum_{\sigma \in \Pi^m} t_{j_{\sigma(1)} \ldots j_{\sigma(m)}} \quad \forall (j_1, \ldots, j_m) \in [d]^m.$$  

**Lemma 2.3** ([24]). The sym operation in Definition 2.2 is an orthogonal projection and so self-adjoint. In particular, for a vector $v \in \mathbb{R}^n$ and tensor $\mathcal{T} \in \mathcal{T}_n^d$,

$$\langle \text{sym}(\mathcal{T}), v^{\otimes m} \rangle = \langle \mathcal{T}, v^{\otimes m} \rangle.$$
2.3 Symmetric tensor decomposition

Definition 2.4. For a symmetric tensor \( \mathcal{T} \in S^m_d \), a real symmetric CP decomposition is an expression

\[
\mathcal{T} = \sum_{i=1}^{r} \lambda_i a_i^{\otimes m},
\]

where \( r \in \mathbb{Z} \) is smallest possible, \( \lambda_i \in \mathbb{R} \), and \( a_i \in \mathbb{R}^d \) (without loss of generality, \( \|a_i\|_2 = 1 \)). The minimal \( r \) is the real symmetric CP rank of \( \mathcal{T} \).

Remark 2.5. Some guarantees in this paper hold only for Zariski-generic \( a_i \) and \( \lambda_i \) in (7). This means that there exists a polynomial \( p \) such that the guarantees are valid whenever \( p(a_1, \ldots, \lambda_r) \neq 0 \) occurs. Furthermore, \( p(a_1, \ldots, \lambda_r) \neq 0 \) indeed holds for some unit-norm \( a_i \in \mathbb{R}^d \) and \( \lambda_i \in \mathbb{R} \) (see [25] for background). In particular, this implies the guarantees hold with probability 1 provided \( a_i \) and \( \lambda_i \) are drawn from any absolutely continuous probability distributions on the sphere and real line.

2.4 Unfolding tensors

Definition 2.6. We let \( \zeta_{d_1, \ldots, d_m} \) be the index map that maps multi-indexes in \( \prod [d_i] \) to the corresponding indices in \( \prod d_i \) in lexicographic order. Formally,

\[
\zeta_{d_1, \ldots, d_m}(i_1, \ldots, i_m) = 1 + \sum_{j=1}^{m} (i_j - 1) \prod_{k=0}^{j-1} d_k
\]

with the convention \( d_0 = 1 \).

Definition 2.7. Let \( \mathcal{T} \in T^m_d \). The vectorization of \( \mathcal{T} \) denoted \( \text{vec}(\mathcal{T}) \in \mathbb{R}^{d^m} \) is defined by

\[
\text{vec}(\mathcal{T})_{\zeta_{d_1, \ldots, d_m}(i_1, \ldots, i_m)} = \mathcal{T}_{i_1, \ldots, i_m}.
\]

For integers \( D_1, D_2 \) such that \( D_1 D_2 = d^m \), define the function \( \text{reshape} \) so that \( \text{reshape}(\mathcal{T}) \in \mathbb{R}^{D_1 \times D_2} \) and

\[
\text{reshape}(\mathcal{T}, D_1, D_2)_{i_1 i_2} = \text{vec}(\mathcal{T})_{\zeta_{D_1, D_2}(i_1, i_2)}.
\]

For \( 1 \leq n < m \), we denote the \( n \)-th matrix unfolding (or \( n \)-th matrix flattening) of \( \mathcal{T} \) by \( \text{mat}_n(\mathcal{T}) = \text{reshape}(\mathcal{T}, d^n, d^{m-n}) \).

The next definition is used to describe unfoldings of tensors with low CP rank.

Definition 2.8. Let \( A \in \mathbb{R}^{d \times r} \) be a matrix with columns \( a_1, \ldots, a_r \in \mathbb{R}^d \). The \( n \)-th Khatri-Rao power of \( A \), denoted \( A^{\otimes n} \in \mathbb{R}^{d^n \times r} \), is defined to be the matrix with columns \( \text{vec}(a_1^{\otimes n}), \ldots, \text{vec}(a_r^{\otimes n}) \in \mathbb{R}^{d^n} \).

3 Algorithm Description

In this section, we derive SPM (Algorithm 1). The input is a real symmetric tensor \( \mathcal{T} \in S^m_d \) (\( m \geq 3 \)). For purposes of method development, we assume that \( \mathcal{T} \) admits an exact low-rank
decomposition \([1]\), where \(r = \mathcal{O}(d^{[m/p]})\). The output is \((\lambda_i, a_i)_{i=1}^r\) for \(i = 1, \ldots, r\) (up to sign ambiguity).

For the remainder of this section, we fix a positive integer \(n < m\), and consider only the \(n\)-th matrix flattening \(\text{mat}_n\), abbreviated as \(\text{mat}\). SPM applies to any matrix unfolding, but in our discussion and implementation of SPM we often set \(n = \lceil \frac{m}{2} \rceil\) by default. This choice maximizes the greatest rank for which Algorithm \([1]\) works.

### 3.1 Three steps

It is convenient to divide the algorithm description into three steps: Extract Subspace, Power Method and Deflate.

**Extract Subspace.** Observe the tensor decomposition \([1]\) of \(\mathcal{T}\) is equivalent to a matrix factorization of the flattening of \(\mathcal{T}\) (Definition \([2,7]\)):

\[
\text{mat}(\mathcal{T}) = \sum_{i=1}^r \lambda_i \text{mat}(a_i^{\otimes m}) = \sum_{i=1}^r \lambda_i \text{vec}(a_i^{\otimes n}) \text{vec}(a_i^{\otimes(m-n)})^T
\]

where \(n = \lceil \frac{m}{2} \rceil\). Let \(A \in \mathbb{R}^{d \times r}\) be the matrix with columns \(a_1, \ldots, a_r \in \mathbb{R}^d\), and \(A \in \mathbb{R}^{r \times r}\) be the diagonal matrix with entries \(\lambda_1, \ldots, \lambda_r\). Then \([8]\) reads

\[
\text{mat}(\mathcal{T}) = A^{*n} A (A^{*(m-n)})^T,
\]

Here bullet denotes the Khatri-Rao power (Definition \([2,8]\)). Define the subspace

\[
\mathcal{A} = \text{span}\{a_1^{\otimes n}, \ldots, a_r^{\otimes n}\} \subset S_d^n.
\]

It is the column space of \(A^{*n}\) upon unvectorization.

**Extract Subspace** obtains an orthonormal basis for \(\mathcal{A}\) from \(\text{mat}(\mathcal{T})\) in \([9]\).

**Proposition 3.1.**

- Let \(p = \min(n, m-n)\) and assume \(a_1^{\otimes p}, \ldots, a_r^{\otimes p}\) are linearly independent, and \(\lambda_1, \ldots, \lambda_r\) are nonzero. Then \(\text{mat}(\mathcal{T})\) has rank \(r\). Moreover if \(\text{mat}(\mathcal{T}) = USV^T\) is a thin SVD, then the columns of \(U\) give an orthonormal basis of \(\mathcal{A}\) (after unvectorization).

- If \(r \leq \binom{n+p-1}{p}\) and \(a_1, \ldots, a_r\) are Zariski-generic, then \(a_1^{\otimes p}, \ldots, a_r^{\otimes p}\) are linearly independent.

**Proof.** Assume \(a_1^{\otimes p}, \ldots, a_r^{\otimes p}\) are linearly independent, or equivalently \(A^{*p} \in \mathbb{R}^{d \times r}\) has rank \(r\). We claim this implies \(\text{rank}(A^{*p}) = \text{rank}(A^{*(m-n)}) = r\). Letting \(q = \max(n, m-n)\), it is enough to show \(\text{rank}(A^{*q}) = r\). Suppose \(\alpha_1, \ldots, \alpha_r \in \mathbb{R}\) satisfy \(\alpha_1 a_1^{\otimes p} + \cdots + \alpha_r a_r^{\otimes p} = 0\). Contract both sides \(q-p\) times with a vector \(z \in \mathbb{R}^d\) such that \(z^T a_i \neq 0\) for all \(i = 1, \ldots, r\) to obtain \(\alpha_1 (z^T a_1)^{q-p} a_1^{\otimes p} + \cdots + \alpha_r (z^T a_r)^{q-p} a_r^{\otimes p} = 0\). This implies \(\alpha_i (z^T a_i)^{q-p} = 0\) by linear independence of \(a_1^{\otimes p}, \ldots, a_r^{\otimes p}\) for \(i = 1, \ldots, r\), whence \(\alpha_i = 0\) for \(i = 1, \ldots, r\) as claimed.

Let \(A^{*n} = Q_n W_n\) be a thin QR factorization, i.e., \(Q_n \in \mathbb{R}^{d \times r}\) has orthonormal columns and \(W_n \in \mathbb{R}^{r \times r}\) is upper-triangular and full rank. Likewise, write \(A^{*(m-n)} = Q_{m-n} W_{m-n}\). Let
We then have:

\[ \text{mat}(\mathcal{J}) = A^*nA(A^*(m-n))^T = (Q_nW_n)A(Q_{m-n}W_{m-n})^T \]

\[ = Q_nU\hat{S}\hat{V}^TQ_{m-n} = (Q_n\hat{U})\hat{S}(Q_{m-n}\hat{V})^T. \]

As \( W_nA^*W_{m-n} \) is a product of nonsingular matrices (assuming \( \lambda_1, \ldots, \lambda_r \) are nonzero), \( \hat{S} \) is nonsingular and \( \text{rank(mat(\mathcal{J}))} = r \). Furthermore, \( Q_n\hat{U} \) and \( Q_{m-n}\hat{V} \) have orthonormal columns.

So \( \text{mat}(\mathcal{J}) = (Q_n\hat{U})\hat{S}(Q_{m-n}\hat{V})^T \) is a thin SVD. Since \( USV^T \) is also a thin SVD of \( \text{mat}(\mathcal{J}) \), it holds

\[ \text{colspan(U)} = \text{colspan(Q}_n\hat{U}) = \text{colspan(Q}_n) = \text{colspan(A}^*n), \]

and the columns of \( U \) give an orthonormal basis of \( A \).

For the second bullet, assume \( r \leq \binom{d+m-n-1}{m-n} \). Note \( a_1^{\otimes(m-n)}, \ldots, a_r^{\otimes(m-n)} \) are linearly independent if and only all \( r \times r \) minors of \( A^*(m-n) \) are nonzero. This is a Zariski-open condition on \( a_1, \ldots, a_r \). It holds generically because it holds for some \( a_1, \ldots, a_r \), since rank-1 symmetric tensors \( \text{span} S_d^{m-n} \) and \( \text{dim}(S_d^{m-n}) = \binom{d+m-n-1}{m-n} \).

In view of the proposition, SPM extracts the subspace \( A \subset S_d^n \) from a thin SVD of \( \text{mat}(\mathcal{J}) \), see Algorithm 1. For noisy input tensors, we must truncate the full-rank SVD of \( \text{mat}(\mathcal{J}) \), see Proposition 3.1 in the noiseless case, enables SPM to estimate the rank without prior knowledge of \( r \).

**Power Method.** The next step of SPM is to find a rank-1 point in \( A \). The following result is the essential underpinning.

**Proposition 3.2.** Let \( r \leq \binom{d+n-1}{n} - d \). Then for Zariski-generic \( a_1, \ldots, a_r \), the only rank-1 tensors in \( A = \text{span}\{a_1^{\otimes n}, \ldots, a_r^{\otimes n}\} \subset S_d^n \) are \( a_1^{\otimes n}, \ldots, a_r^{\otimes n} \) (up to scale).

**Proof.** This is a special case of the generalized trisecant lemma in algebraic geometry, see [16, Prop. 2.6] or [27, Exer. IV-3.10]. The set of rank \( \leq 1 \) tensors is an irreducible algebraic cone of dimension \( d \) linearly spanning its ambient space \( S_d^n \). It is the affine cone over the Veronese variety; denote it \( \mathcal{V}_d^n \). Note \( A \) is a secant plane through \( r \) general points on \( \mathcal{V}_d^n \). The dimensions of \( \mathcal{V}_d^n \) and \( A \) are subcomplimentary:

\[ \text{dim}(A) + \text{dim}(A) = d + r \leq \binom{d+n-1}{n} = \text{dim}(S_d^n). \]

Therefore the generalized trisecant lemma applies. It implies that \( \mathcal{V}_d^n \cap A \) have no unexpected intersection points. Precisely, \( \mathcal{V}_d^n \cap A = \text{span}\{a_1^{\otimes n}\} \cup \ldots \cup \text{span}\{a_r^{\otimes n}\} \).

In **Power Method**, we seek a rank-1 element in \( A \) by solving the program:

\[ \max_{x \in \mathbb{R}^d} F_A(x) \quad \text{subject to } \|x\| = 1, \] (12)

where

\[ F_A(x) = \|P_A(x^{\otimes n})\|^2, \] (13)

with \( P_A \) the orthogonal projector from \( T_d^n \) onto \( A \). The next result justifies [12].
Proposition 3.3. For all \( x \) with \( \|x\| = 1 \), we have \( F_A(x) \leq 1 \) with equality if and only if \( x^{\otimes n} \in A \). If \( r \leq \binom{d+n-1}{d} - d \) and \( A = \text{span}\{a_1^{\otimes n}, \ldots, a_r^{\otimes n}\} \subset S^{n}_d \) where \( a_1, \ldots, a_r \) are Zariski-generic, then the global maxima of (12) are precisely \( \pm a_1, \ldots, \pm a_r \) with function value 1.

Proof. If \( \|x\| = 1 \), then

\[
\|P_A(x^{\otimes n})\|^2 \leq \|P_A(x^{\otimes n})\|^2 + \|P_{A^\perp}(x^{\otimes n})\|^2 = \|x^{\otimes n}\|^2 = 1,
\]

where \( P_{A^\perp} \) denotes orthogonal projection onto the orthogonal complement \( A^\perp \) of \( A \), thus \( F_A(x) \leq 1 \). Moreover, \( F_A(x) = 1 \) if and only if \( \|P_{A^\perp}(x^{\otimes n})\|^2 = 0 \) if and only if \( x^{\otimes n} \) lies in \( A \). The second sentence is immediate from Proposition 3.2. \( \square \)

In Power Method, we use projected gradient descent to solve (12). We initialize \( x \) as a random vector in the unit-sphere \( S^{d-1} \), and iterate

\[
x \leftarrow \frac{P_A(x^{\otimes n}) \cdot x^{\otimes n-1} + \gamma x}{\|P_A(x^{\otimes n}) \cdot x^{\otimes n-1} + \gamma x\|}
\]

(14) until convergence. Here \( \gamma > 0 \) is a fixed constant, whose reciprocal is the step size; according to Theorem 1.1, we may set \( \gamma > \sqrt{\frac{d-1}{n}} \). The iteration (14) is calculated using \( U \) obtained in Extract Subspace (see Proposition 3.1). Specifically if \( u_1, \ldots, u_r \) are the unvectorized columns of \( U \), then

\[
P_A(x^{\otimes n}) = \sum_{i=1}^{r} \langle u_i, x^{\otimes n} \rangle u_i,
\]

(15) and

\[
P_A(x^{\otimes n}) \cdot x^{\otimes n-1} = \sum_{i=1}^{r} \langle u_i, x^{\otimes n} \rangle u_i \cdot x^{\otimes n-1}.
\]

(16)

Let \( \bar{U} = \text{reshape}(U, d^{n-1}, dr) \) and \( W = \text{reshape}(\text{vec}(x^{\otimes n-1})^T \bar{U}, d, r) \). If \( w_1, \ldots, w_r \) are the columns of \( W \), it holds \( w_i = U_i \cdot x^{\otimes n-1} \) and \( w_i^\top x = \langle U_i, x^{\otimes n} \rangle \). Thus (16) is

\[
P_A(x^{\otimes n}) \cdot x^{\otimes n-1} = WW^\top x.
\]

Suppose now (14) converges to \( \bar{x} \in \mathbb{R}^d \). We check if \( F_A(\bar{x}) = 1 \) (up to a tolerance). If so, we proceed with the Deflate step below, as \( \bar{x} \) is a CP component under the conditions in the last sentence of Proposition 3.3. Else, \( \bar{x} \) is discarded and Power Method is repeated with a fresh random initialization. In Subsection 5.4 it is observed that often Power Method converges to a CP component on its first try.

Remark 3.4. The iteration (14) is equivalent to the shifted symmetric higher-order power method of \( [36] \) applied to a certain modified tensor, different from \( T \). We explain this in Subsection 4.1. That is why we call this step Power Method.

Deflate. The last step of SPM is Deflate. Given one CP component \( \pm a_i \) from Power Method, it calculates the corresponding coefficient \( \pm \lambda_i \). Then it removes the term \( \lambda_i a_i^{\otimes m} \) from \( T \) by appropriately updating the factorization of \( \text{mat}(T) \).
Assume without loss of generality we obtained \( a_1 \) from Power Method. Define

\[
W_\tau = \text{mat}(\tilde{J}) - \tau \text{vec}(a_1^{\otimes n}) \text{vec}(a_1^{\otimes (m-n)})^\top \\
= (\lambda_1 - \tau) \text{vec}(a_1^{\otimes n}) \text{vec}(a_1^{\otimes (m-n)})^\top + \sum_{i=2}^r \lambda_i \text{vec}(a_i^{\otimes n}) \text{vec}(a_i^{\otimes (m-n)})^\top
\]

for \( \tau \in \mathbb{R} \). If \( A^{\otimes n} \) has full column rank, by (17) and Proposition 3.1 it follows that \( W_\tau \) has rank \( r-1 \) if \( \tau = \lambda_1 \) and rank \( r \) otherwise. This property determines \( \lambda_1 \). A formula for \( \lambda_1 \) is given by Wedderburn rank reduction [18, Theorem 1.1]:

\[
\lambda_1 = \frac{1}{\text{vec}(a_1^{\otimes (m-n)})^\top \text{mat}(\tilde{J})^\dagger \text{vec}(a_1^{\otimes n})},
\]

where \( ^\dagger \) denotes the Moore-Penrose pseudo-inverse. In our implementation, we use formulas for updating \( U \) and \( V \) directly without recalculating the thin SVD of the deflated flattened tensor.

\[
(U, S, V) \leftarrow \text{svd}(W_{\lambda_1}) = \text{svd} \left( \sum_{i=2}^r \lambda_i \text{mat}(a_i^{\otimes m}) \right).
\]

However for efficiency reasons, rather than storing and updating \( S \), we store a matrix \( C \), which we set initially to \( S^{-1} \) and update throughout the algorithm enforcing that \( \text{mat}(\tilde{J}) = UC^{-1}V^T \), where \( \tilde{J} \) is the deflated tensor. In the following proposition we witness that it is more convenient to calculate \( \lambda_1 \) in terms of \( C \), and that the update formulas for the factorization are favorable.

**Proposition 3.5.** Let \( U \in \mathbb{R}^{d \times r}, V \in \mathbb{R}^{r \times m} \) have orthonormal columns and \( C \in \mathbb{R}^{r \times r} \) be nonsingular such that \( \text{mat}(\tilde{J}) = UC^{-1}V^T \). (Here \( C \) is not necessarily diagonal.) Suppose, after possibly relabeling and/or flipping sign, we obtain \( a_1 \) from Power Method. Then

- We obtain the corresponding coefficient as
  \[
  \lambda_1 = \frac{1}{\beta^\top C \alpha} \quad \text{where} \quad \alpha = U^\top \text{vec}(a_1^{\otimes n}) \quad \text{and} \quad \beta = V^\top \text{vec}(a_1^{\otimes m-n}).
  \]

- Let \( O_\alpha, O_\beta \) be \( r \times (r-1) \) matrices whose columns form orthonormal bases for \( \text{span}\{C\alpha\}^\perp \) and \( \text{span}\{C^\top \beta\}^\perp \), respectively. We update

\[
(U, C, V) \leftarrow (\tilde{U}, \tilde{C}, \tilde{V}), \quad \text{where} \quad \tilde{U} = UO_\alpha, \quad \tilde{V} = VO_\alpha \quad \text{and} \quad \tilde{C} = O_\alpha^\top CO_\beta.
\]

The update guarantees that \( \tilde{U}, \tilde{V} \) have orthonormal columns, \( \tilde{C} \) is nonsingular and \( \tilde{U}\tilde{C}^{-1}\tilde{V}^T = \text{mat}(\tilde{J}) \), where \( \tilde{J} \) is the deflated tensor

\[
\tilde{J} = J - \lambda_1 a_1^{\otimes m} = \sum_{i=2}^{r} \lambda_i a_i^{\otimes m}.
\]

- The columns of \( \tilde{U} \) give an orthonormal basis for \( \text{span}\{a_2^{\otimes n}, \ldots, a_r^{\otimes n}\} \).
Proof. The formula for \( \lambda_1 \) follows from (18):

\[
\lambda_1 = \frac{1}{\mat{J} \mat{T} \vec{a}_i^{(m-n)}} = \frac{1}{\mat{T} \vec{a}_i^{(m-n)}} = \frac{1}{\beta^\top \alpha}
\]

To show the remaining bullets, let \( \tilde{S} = C^{-1} - \lambda_1 \alpha \beta^\top \). Then

\[
\tilde{S} \vec{V}^\top = U C^{-1} \vec{V} - \lambda_1 \vec{V} \vec{V}^\top \vec{V} = W_{\lambda_1},
\]

where we used \( \vec{V} \vec{V}^\top \vec{V}^{(m-n)} = \vec{a}_i^{(m-n)} \), since \( \vec{a}_i^{(m-n)} \in A = \cospan(U) \), and similarly \( \vec{V} \vec{V}^\top \vec{V}^{(m-n)} = \vec{a}_i^{(m-n)} \). Our proof strategy is to show the following hold:

(21) \( \tilde{S} = O_\alpha O_\beta^\top \tilde{S} O_\alpha O_\beta^\top \),
(22) \( \tilde{C} = (O_\beta^\top \tilde{S} O_\alpha)^{-1} \).

These will imply \( \tilde{U} C^{-1} \tilde{V} = U O_\beta O_\beta^\top \tilde{S} O_\alpha O_\beta^\top \tilde{V} = U \tilde{S} \vec{V}^\top = W_{\lambda_1} \). Since \( U \) and \( O_\alpha \) have orthonormal columns, \( \tilde{U} = U O_\alpha \) also has orthonormal columns (likewise for \( \tilde{V} \)), and so the proposition follows.

We now prove (21). Let \( y = \frac{C \alpha}{\|C \alpha\|} \), then the definition of \( O_\alpha \) implies that \( [O_\alpha \ y] \) is a \( r \times r \) orthogonal matrix, which implies \( O_\alpha O_\alpha^\top = I - yy^\top \). Moreover, since \( \lambda_1 = 1/(\beta^\top \alpha) \), we have

\[
\tilde{S} y = \frac{1}{\|C \alpha\|} (C^{-1} \alpha - \lambda_1 \alpha \beta^\top \alpha) = \frac{1}{\|C \alpha\|} (\alpha - \alpha) = 0.
\]

Therefore, \( \tilde{S} O_\alpha O_\alpha^\top = \tilde{S} - \tilde{S} y y^\top = \tilde{S} \). The verification of \( \tilde{S} = O_\beta O_\beta^\top \tilde{S} \) is analogous, and (21) follows. Regarding (22), using again that \( \tilde{S} = \tilde{S} O_\alpha O_\alpha^\top \), with \( \beta^\top \alpha \beta = 0 \) which follows from the definition of \( O_\beta \), we obtain

\[
(O_\beta^\top \tilde{S} O_\alpha) \tilde{C} = O_\beta^\top \tilde{S} O_\alpha O_\alpha^\top \alpha \beta^\top \alpha = O_\beta^\top \tilde{S} \alpha \beta^\top \alpha = O_\beta^\top \alpha \beta^\top \alpha = O_\beta^\top \alpha = I.
\]

In light of the proposition, DEFLATE proceeds as follows. Set \( \alpha \leftarrow \vec{U} \vec{V} \vec{a}_i^{(m-n)} \), \( \beta \leftarrow \vec{V} \vec{V} \vec{a}_i^{(m-n)} \) and \( \lambda_1 = 1/\beta^\top \alpha \). We then calculate \( O_\alpha \) and \( O_\beta \), and update \( U, C, \vec{V} \leftarrow (U O_\beta, O_\alpha^\top \beta \alpha \beta^\top \alpha, VO_\alpha) \).

By design, the procedure enjoys two nice properties. Firstly, the columns of \( \tilde{U} \) give an orthonormal basis of \( \tilde{A} = \cospan(\tilde{a}_i^{(m-n)}, \ldots, \tilde{a}_r^{(m-n)}) \). Thus, we can use \( \tilde{U} \) for the next run of POWER METHOD. Secondly, the orthogonal matrices \( O_\alpha \) and \( O_\beta \) can be constructed efficiently using Householder reflections. We explain the implementation for \( O_\alpha \); \( O_\beta \) is analogous. Set \( y = \frac{C \alpha}{\|C \alpha\|} \) and define \( z \in \mathbb{R}^r \) by \( z_r = \sqrt{1 + \|y_r\|^2} \) and \( z_i = \text{sign}(y_i)y_i/z_r \) for \( i = 1, \ldots, r - 1 \). It is easily checked \( \|z\|^2 = 2 \), therefore the matrix \( H = I - zz^\top \) is a Householder reflection, and the last column \( H \) is \( \text{sign}(y_r)y_r \). We pick \( O_\alpha \) to be the first \( r - 1 \) columns of \( H \), which form an orthonormal basis for \( \cospan(y)^\perp = \cospan(C \alpha)^\perp \). Using Householder reflections is more efficient than explicitly forming \( O_\alpha \), because we can exploit that \( O_\alpha \) is a rank-1 update of the identity matrix to calculate the matrix product \( VO_\alpha \) in \( O(d^m - n r) \) time. It gives a speed-up compared to calculating this product naively, which takes \( O(d^m - n r^2) \) time.
3.2 Full algorithm

**POWER METHOD** and **DEFlate** repeat as subroutines, such that each CP component \((\lambda_i, a_i)\) is removed one at a time, until all components have been found. The full Subspace Power Method is detailed in Algorithm 1 below.

**Algorithm 1** Subspace Power Method (SPM)

**Input:** generic \(T \in S^m_d\) of rank \(r\) satisfying (23)

Hyperparameters: \(\kappa > 0, \zeta > 0, 1 \leq n < m, \gamma > \sqrt{\frac{n-1}{n}}\)

**Output:** rank \(r\) and tensor decomposition \(\{(\lambda_i, a_i)\}_{i=1}^r\)

\[
(U, S, V) \leftarrow \text{svd}(\text{mat}(T))
\]

\[C \leftarrow S^{-1}\]

\[r \leftarrow \text{rank}\left(\text{mat}(T)\right)\]

for \(i = 1\) to \(r\) do

\[\bar{U} \leftarrow \text{reshape}(U, d^{n-1}, dr)\]

\[x \leftarrow \text{random}(S^{d-1})\]

repeat

\[\tilde{x} \leftarrow x\]

\[W \leftarrow \text{reshape}(\text{vec}(x^\otimes n-1)^\dagger \bar{U}, d, r)\]

\[x \leftarrow WW^\dagger x + \gamma x\]

until \(\|x - \tilde{x}\| < \kappa\)

**Extract Subspace**

if \(F_A(x) > \zeta\) then \(a_i \leftarrow x\)
else repeat **POWER METHOD**

\[\alpha \leftarrow U^\dagger \text{vec}(a_i^\otimes n), \beta \leftarrow V^\dagger \text{vec}(a_i^\otimes m-n)\]

\[\lambda_i \leftarrow 1/\beta^\dagger C\alpha\]

\[O_{\alpha} \leftarrow \text{orthonormal basis of span}\{C\alpha\}^\perp\]

\[O_{\beta} \leftarrow \text{orthonormal basis of span}\{C^\dagger \beta\}^\perp\]

\[(U, C, V) \leftarrow (UO_{\beta}, O_{\alpha}^\dagger CO_{\beta}, VO_{\alpha})\]

**DEFlate**

return \(r\) and \(\{(\lambda_i, a_i)\}_{i=1}^r\)

3.3 Practical considerations

Computational and storage costs

The computational costs of Algorithm 1 are as follows. **Extract Subspace** computes \(\text{svd}(\text{mat}(T))\) upfront in \(O(d^m+n)\). If an upper bound for the rank \(\bar{r} \geq r\) is known a priori, this drops to \(O(d^m \bar{r})\) (e.g., using randomized linear algebra). Suppose \(s \leq r\) components \((\lambda_i, a_i)\) are yet to be found. **POWER METHOD**, each iteration of \(O(14)\) costs \(O(sd^n)\), the price of applying \(P_A\). In **DEFlate**, computing \(\alpha\) and \(\beta\) cost \(O(sd^n)\) and \(O(sd^{m-n})\) respectively, computing \(\lambda_i\) is \(O(s^2)\) and updating \((U, C, V)\) is \(O(s(d^m+n + d^n + s))\). The storage costs are \(O(s(d^{m-n} + d^n + s))\), corresponding to storing the matrix factorization of \(\text{mat}(T)\). The storage of this matrix factorization dominates the other storage costs.
Maximal rank

If the CP components are Zariski-generic, SPM can work up to the ranks in Propositions 3.1 and 3.2. These require \( r \leq \binom{d + p - 1}{p} \) (where \( p = \min(n, m - n) \)), and \( r \leq \binom{d + n - 1}{n - d} \), respectively. Together, the conditions imply

\[
r \leq \binom{d + p - 1}{p} - \delta_{p=n} d,
\]

where \( \delta_{p=n} = 1 \) if \( p = n \) and \( \delta_{p=n} = 0 \) otherwise. The maximal rank is obtained when \( n = [m/2] \) and \( p = [m/2] \). The formulae for the maximal tensor rank for the first few tensor orders are

- \( m = 3 : \quad r \leq d \)
- \( m = 4 : \quad r \leq \frac{1}{2}d(d - 1) \)
- \( m = 5 : \quad r \leq \frac{1}{2}d(d + 1) \)
- \( m = 6 : \quad r \leq \frac{1}{2}d(d^2 + 3d - 4) \).

Eigendecomposition

When \( m \) is even and \( n = m/2 \), \( \text{mat}(\mathcal{J}) \) is a symmetric matrix. Then an eigendecomposition algorithm may be used in place of SVD in Extract Subspace. Furthermore, a symmetric variant of Proposition 3.5 holds where \( \alpha = \beta \), \( U = V \) and \( C \) is symmetric.

Unique rows and columns

Since \( \mathcal{J} \) is a symmetric tensor, \( \text{mat}(\mathcal{J}) \) has many repeated rows and columns. While the total number of rows is \( d^n \), it only has \( \binom{d + n - 1}{n} \) unique rows corresponding to \( [d]^n \) up to the action of \( \Pi^n \). Similarly, it has only \( \binom{d + m - n - 1}{m - n} \) unique columns. We calculate the SVD only on the subset of unique rows and unique columns. We rescale each of the unique rows by the square root of the number of times it appears. This preserves the dot-product between the columns and rows, so the SVD of \( \text{mat}(\mathcal{J}) \) is recovered from the SVD of the submatrix. The submatrix has approximately \( \frac{1}{n!} \) of the rows and \( \frac{1}{(m-n)!} \) of the columns of the full matrix, so this speeds up Extract Subspace by about a factor of \( n!/(n-m)! \).

Picking the rank

In applications, the tensor \( \mathcal{J} \) is often only approximately low-rank. In that case, we need to select a cut-off, and truncate all singular values below the cut-off. Choosing the cut-off depends on the application, similarly to PCA.

Hyperparameters

The implementation of SPM uses hyperparameters:

- \( \kappa \), the distance between Power Method iterates below which we declare Power Method has converged (default = \( 1 \times 10^{-14} \));
- \( \zeta \), the minimum function value for \( F_A \) for which we accept the Power Method as having converged to a CP component (default = 0.99);
- the maximum number of Power Method iterations (default = 5000);
the maximum number of repetitions of the overall Power Method step, if the function values are below ζ, after which we pick the iterate with the biggest function value (default = 3).

4 Power Method Analysis

The object of this section is to prove the following convergence guarantee for Power Method, which is only part of SPM using nonconvex optimization.

Theorem 4.1. Assume m ≥ 3 and let $\mathcal{F} \in S^n_m$ satisfy a CP decomposition \cite{1}, where $\lambda_i \in \mathbb{R}$ and $a_i \in S^{d-1}$ for $i = 1, \ldots, r$. Let $A = \text{span}\{a_1^{\otimes n}, \ldots, a_r^{\otimes n}\} \subset S^n_m$ as in \cite{10}, where $n = \lceil \frac{m}{2} \rceil$. Set $F_A(x) = \|P_A(x^{\otimes n})\|^2$ as in \cite{13}, and consider the constrained optimization problem:

$$
\max_{x \in S^{d-1}} F_A(x).
$$

Following Power Method, define the sequence:

$$
x_{k+1} = \frac{P_A(x_k^{\otimes n}) \cdot x_k^{\otimes (n-1)} + \gamma x_k}{\|P_A(x_k^{\otimes n}) \cdot x_k^{\otimes (n-1)} + \gamma x_k\|}.
$$

Here $x_1 \in S^{d-1}$ is an initialization, and $\gamma \in \mathbb{R}_{>0}$ is a fixed shift such that $F_A(x) + \gamma (x^T x)^n$ is a strictly convex function on $\mathbb{R}^n$. For example, $\gamma > \sqrt{\frac{a-1}{n}}$ is a sufficiently large shift. Then

- For all initializations $x_1$, (26) is well-defined and converges monotonically to a first-order critical point $x_*$ of (25) at no less than an algebraic rate. That is, $F_A(x_{k+1}) \geq F_A(x_k)$ for all $k$, and there exist constants $\tau = \tau(A, \gamma, x_1) > 1$ and $C = C(A, \gamma, x_1) > 0$ such that $\|x_k - x_*\| \leq C k^{-\tau}$ for all $k$.

- For a full Lebesgue-measure subset of initializations $x_1$, (26) converges to a second-order critical point of (25).

- If $r \leq (d+n-1) - d$ and $a_1, \ldots, a_r$ are Zariski-generic, the global maxima of (25) are precisely $\pm a_i$. If $r \leq (d+n-1) - d + 1$ and $a_1, \ldots, a_r$ are Zariski-generic, each $\pm a_i$ is locally attractive: for all initializations $x_1$ sufficiently close to $\pm a_i$, (26) converges to $\pm a_i$ at no less than an exponential rate. That is, there exist positive constants $\delta = \delta(A, \gamma, i), \tau = \tau(A, \gamma, i)$ and $C = C(A, \gamma, i)$ such that $\|x_k - \pm a_i\| \leq \delta$ implies $\|x_k - \pm a_i\| \leq C e^{-k\tau}$ for all $k$.

Remark 4.2. It is easy to see that if $\gamma > 0$ then the Power Method sequence (26) is well-defined. The denominator in (26) does not vanish, because $(P_A(x_k^{\otimes n}) \cdot x_k^{\otimes (n-1)} + \gamma x_k, x_k) = \|P_A(x_k^{\otimes n})\|^2 + \gamma > 0$.

Remark 4.3. In Theorem 4.1, critical points are understood in the usual sense of manifold optimization. That is, $x_* \in S^{d-1}$ is first-order critical if $\text{grad } F_A(x_*) = 0$, and it is second-order critical if in addition $\text{Hess } F_A(x_*) \succeq 0$, where grad and Hess denote the Riemannian gradient and Riemannian Hessian on $S^{d-1}$ respectively (see \cite{7} Prop. 4.6 and \cite{7} Prop. 6.3). More concretely, from \cite{1} Sec. 4.2 it holds

$$
\text{grad } F_A(x_*) = (I - x_* x_*^T) \nabla F_A(x_*),
$$

$$
\text{Hess } F_A(x_*) = (I - x_* x_*^T) \nabla^2 F_A(x_*)(I - x_* x_*^T) - (x_*^T \nabla F_A(x_*))(I - x_* x_*^T),
$$
where $\nabla$ and $\nabla^2$ denote the Euclidean gradient and Hessian respectively.

The proof of Theorem 4.1 spans the remainder of Section 4, with details appearing in the appendices. As an outline, we identify the iteration (26) with SS-HOPM [36] for computing tensor $\mathbf{Z}$-eigenvectors, except that now SS-HOPM is applied to a certain modification of the tensor $\mathbf{T}$ denoted $\mathbf{\tilde{T}}$. Then we prove the first two bullets of Theorem 4.1 by a new-and-improved general analysis of SS-HOPM. For the third bullet we linearize (26) and make a geometric argument exploiting properties of $\mathbf{\tilde{T}}$.

4.1 Connection with SS-HOPM

In Algorithm 1, let $\mathbf{U}_1, \ldots, \mathbf{U}_r \in S_n^d$ be the orthonormal basis of $\mathcal{A} \subset S_n^d$ given by the columns of $\mathbf{U}$. Equation (15) implies

$$F_A(x) = \|P_A(x^{\otimes n})\|^2 = \langle P_A(x^{\otimes n}), x^{\otimes n} \rangle = \sum_{i=1}^{r} \langle U_i, x^{\otimes n} \rangle^2.$$ 

We define the even-order tensor

$$\mathbf{\tilde{T}} = \sum_{i=1}^{r} \text{sym}(U_i \otimes U_i) \in S_n^{2d}.$$ 

(29)

Notice that by Lemmas 2.1 and 2.3,

$$\langle \mathbf{\tilde{T}}, x^{\otimes 2n} \rangle = \sum_{i=1}^{r} \langle U_i, x^{\otimes n} \rangle^2 = F_A(x),$$

and

$$\mathbf{\tilde{T}} : x^{\otimes (2n-1)} = \frac{1}{2n} \nabla F_A(x) = \sum_{i=1}^{r} \langle U_i, x^{\otimes n} \rangle U_i \cdot x^{\otimes (n-1)} = P_A(x^{\otimes n}) \cdot x^{\otimes (n-1)}.$$ 

It follows that the Power Method iterations (14) and (26) for $\mathbf{T}$ coincide with SS-HOPM iterations applied to the tensor $\mathbf{\tilde{T}}$, with shift $\gamma$. Moreover, from a tensor standpoint such iterations make sense: Proposition 3.3 implies that the CP components $a_i$ of $\mathbf{T}$ are $\mathbf{Z}$-eigenvectors of $\mathbf{\tilde{T}}$, and SS-HOPM computes $\mathbf{Z}$-eigenvectors.

In [36] on SS-HOPM, the shift is chosen so the corresponding homogeneous polynomial becomes a convex function on $\mathbb{R}^d$. Here $\mathbf{\tilde{T}}$ corresponds to the polynomial $F_A(x)$, and to $F_A(x) + \gamma(x^\top x)^n$ with the shift. So the next lemma lets us choose $\gamma$.

**Lemma 4.4.** Let $\mathcal{A} \subset S_n^d$ be any linear subspace (spanned by rank-1 points or not). Let $P_A : S_n^d \to \mathcal{A}$ be orthogonal projection onto $\mathcal{A}$, $F_A(x) = \|P_A(x^{\otimes n})\|^2$ and $\nu \in [0, 1]$. If $\|x\| = 1$ and $F_A(x) \geq \nu$, then

$$\frac{1}{2n} \min_{y \in S^{d-1}} y^\top \nabla^2 F_A(x) y \geq -\sqrt{\frac{n-1}{n}} h(\nu),$$

(30)

where

$$h(\nu) = \begin{cases} 1 - \frac{\nu}{2} & \text{if } \nu \leq \frac{2}{3} \\ \sqrt{2\nu(1-\nu)} & \text{if } \nu > \frac{2}{3}. \end{cases}$$

(31)

In particular, $F_A(x) + \gamma(x^\top x)^n$ is strictly convex on $\mathbb{R}^d$ whenever $\gamma > \sqrt{\frac{n-1}{n}}$.

Lemma 4.4 is proven in Appendix A.1 by a lengthy but direct calculation.
4.2 Global convergence of SS-HOPM

Here we sharpen the analysis of SS-HOPM in general. In this subsection only, \( F(\mathbf{x}) \) stands for any homogeneous polynomial function on \( \mathbb{R}^d \) of degree \( 2n \), that is, not necessarily of the form \( F_A(\mathbf{x}) \) for some subspace \( A \). It corresponds to an arbitrary symmetric tensor \( \mathbf{T} \in S_{2n} \) through \( F(\mathbf{x}) = \langle \mathbf{T}, \mathbf{x}^{2n} \rangle \), rather than \( \mathbf{T} \) as in [29]. We consider the optimization problem

\[
\max_{\mathbf{x} \in \mathbb{S}^{d-1}} F(\mathbf{x}), \tag{32}
\]

whose critical points are the \( Z \)-eigenvectors of \( \mathbf{T} \). Like in [26], SS-HOPM follows the sequence

\[
\mathbf{x}_{k+1} = \frac{\frac{1}{2n} \nabla F(\mathbf{x}_k) + \gamma \mathbf{x}_k}{\left\| \frac{1}{2n} \nabla F(\mathbf{x}_k) + \gamma \mathbf{x}_k \right\|}, \tag{33}
\]

where \( \mathbf{x}_1 \in \mathbb{S}^{d-1} \) is an initialization and \( \gamma \in \mathbb{R} \) is a fixed shift. Assume \( \gamma \) is chosen so

\[
G(\mathbf{x}) = F(\mathbf{x}) + \gamma (\mathbf{x}^\top \mathbf{x})^n \quad \text{is a strictly convex function on } \mathbb{R}^d. \tag{34}
\]

For \( \mathbf{x} \in \mathbb{S}^{d-1} \) denote

\[
\Psi(\mathbf{x}) = \frac{\nabla G(\mathbf{x})}{\|\nabla G(\mathbf{x})\|}, \tag{35}
\]

so that (33) may be written

\[
\mathbf{x}_{k+1} = \Psi(\mathbf{x}_k). \tag{36}
\]

Note that \( \mathbf{x}_* \in \mathbb{S}^{d-1} \) is first-order critical for (32) if and only if \( \mathbf{x}_* \) is a fixed point of \( \Psi \).

In the next result we resolve Kolda-Mayo’s conjecture [36, p. 1107] for even-order tensors, corresponding to even-degree homogeneous polynomials, by establishing that SS-HOPM converges for all initializations. A main tool comes from Łojasiewicz’s inequality for real analytic functions [42].

**Theorem 4.5** (Unconditional convergence of SS-HOPM). Assume the setting of (32)–(35). Then for all initializations \( \mathbf{x}_1 \in \mathbb{S}^{d-1} \), the sequence (33) is well-defined and (33) converges monotonically to a first-order critical point of (32) at no less than an algebraic rate.

**Proof.** Note the denominators in (33) do not vanish, as \( \nabla G(\mathbf{x}_k) = \nabla F(\mathbf{x}_k) + 2n\gamma (\mathbf{x}_k^\top \mathbf{x}_k)^{n-1} \mathbf{x}_k = \nabla F(\mathbf{x}_k) + 2n\gamma \mathbf{x}_k \), so \( \langle \mathbf{x}_k, \frac{1}{2n} \nabla F(\mathbf{x}_k) + \gamma \mathbf{x}_k \rangle = \langle \mathbf{x}_k, \frac{1}{2n} \nabla G(\mathbf{x}_k) \rangle = \frac{1}{2n} G(\mathbf{x}_k) \) which is positive since \( G \) is even and strictly convex. It follows that (33) is well-defined. Also clearly we may prove \( (\mathbf{x}_k)_{k=1}^\infty \) converges monotonically at (at least) a power rate to a first-order critical point of

\[
\max_{\mathbf{x} \in \mathbb{S}^{d-1}} G(\mathbf{x}), \tag{37}
\]

because \( G(\mathbf{x}) = F(\mathbf{x}) + \gamma \) for \( \mathbf{x} \in \mathbb{S}^{d-1} \).

By convexity of \( G \),

\[
G(\mathbf{x}_{k+1}) - G(\mathbf{x}_k) \geq \nabla G(\mathbf{x}_k)^\top (\mathbf{x}_{k+1} - \mathbf{x}_k). \tag{38}
\]

From (36) and the Cauchy-Schwarz inequality,

\[
\nabla G(\mathbf{x}_k)^\top (\mathbf{x}_{k+1} - \mathbf{x}_k) = \|\nabla G(\mathbf{x}_k)\| - \nabla G(\mathbf{x}_k)^\top \mathbf{x}_k \geq \|\nabla G(\mathbf{x}_k)\| - \|\nabla G(\mathbf{x}_k)\| \|\mathbf{x}_k\| = 0.
\]

16
Thus \((G(x_k))_{k=1}^\infty\) monotonically increases.

Next suppose \((x_k)_{k=1}^\infty\) indeed converges to \(x_\ast \in S^{d-1}\). Taking limits in (36), continuity of \(G\) implies

\[
x_\ast = \frac{\nabla G(x_\ast)}{\|\nabla G(x_\ast)\|}.
\]

In particular, \((I - x_\ast x_\ast^T)\nabla G(x_\ast) = 0\), and so \(x_\ast\) is a first-order critical point of (37) (see Remark 4.3).

It remains to show that \((x_k)_{k=1}^\infty\) actually converges, and that it does so at at least a power rate. To this end we apply a convergence result of Schneider-Uschmajew [54] Theorem 2.3 based on the Łojasiewicz inequality for real analytic functions, see a precise statement in Appendix B.1.

We take \(M\) in Theorem B.1 to be \(S^{d-1} \subset \mathbb{R}^d\).

To verify condition (A1) in Theorem B.1 we must verify there exists \(\sigma > 0\) such that for large enough \(k\),

\[
G(x_{k+1}) - G(x_k) \geq \sigma \|\nabla G(x_k)\|\|x_{k+1} - x_k\|,
\]

where \(\nabla G(x_k) = (I - x_k x_k^T) \nabla G(x_k)\) denotes the Riemannian gradient. In fact, \(\sigma = \frac{1}{2}\) works. Indeed from (38) and (36),

\[
G(x_{k+1}) - G(x_k) \geq \nabla G(x_k)^T (x_{k+1} - x_k) = \|\nabla G(x_k)\| x_{k+1}^T (x_{k+1} - x_k)
\]

\[
= \|\nabla G(x_k)\| (1 - (x_{k+1}, x_k)) = \frac{1}{2} \|\nabla G(x_k)\| \|x_{k+1} - x_k\|^2.
\]

On the other hand,

\[
\|\nabla G(x_k)\|^2 = \|(I - x_k x_k^T) \nabla G(x_k)\|^2 = \|\nabla G(x_k)\|^2 \|x_{k+1} - (x_{k+1}, x_k) x_k\|^2
\]

\[
= \|\nabla G(x_k)\|^2 (1 - (x_{k+1}, x_k)^2) = \|\nabla G(x_k)\|^2 (1 - (x_{k+1}, x_k))(1 + (x_{k+1}, x_k))
\]

\[
\leq 2 \|\nabla G(x_k)\|^2 (1 - (x_{k+1}, x_k)) = \|\nabla G(x_k)\|^2 \|x_{k+1} - x_k\|^2.
\]

Substituting the square root of this into (40) yields (39) with \(\sigma = \frac{1}{2}\).

To check condition (A2) in Theorem B.1 it is required to verify if \(k\) is large enough then \(G(x_k) = 0\) implies \(x_{k+1} = x_k\). But this is immediate from (36), because \(\nabla G(x_k) = (I - x_k x_k^T) \nabla G(x_k)\) is 0 if and only if \(\nabla G(x_k)\) is parallel to \(x_k\).

For condition (A3) in Theorem B.1 we must verify there exists a constant \(\rho > 0\) such that for large enough \(k\) it holds \(\|x_{k+1} - x_k\| \geq \rho \|\nabla G(x_k)\|\). However by the above, we may take \(\rho = \left(\max_{\|x\|=1} \|\nabla G(x)\|\right)^{-1}\).

Theorem B.1 implies the sequence converges at least an algebraic rate.

\[\square\]

Next we prove that SS-HOPM converges to second-order critical points of (32) for almost all initializations. In the language of [32], SS-HOPM converges to stable eigenvectors. We adopt the proof strategy of [48] based on the center-stable manifold theorem from dynamical systems. The following lemma is a key calculation.

**Lemma 4.6.** Assume the setting of (32) - (36). For all \(x \in S^{d-1}\) the Jacobian \(D \Psi(x)\) as a linear map between tangent spaces to the sphere is

\[
D \Psi(x) = (I - \Psi(x) \Psi(x)^T) \frac{\nabla^2 G(x)}{\|\nabla G(x)\|}(I - xx^T),
\]

(41)
where \(\nabla^2 G(x)\) denotes the Euclidean Hessian. At a first-order critical point \(x_* \in S^{d-1}\) of (32) it holds
\[
D\Psi(x_*) = \frac{\text{Hess} F(x_*)}{2n(F(x_*) + \gamma)} + I - x_*x_\top,
\]
where Hess denotes the Riemannian Hessian.

The proof of Lemma 4.6 is given in Appendix A.2. It implies two additional lemmas, recorded next.

**Lemma 4.7.** Assume the setting of (32)-(36). Then \(\Psi\) is a local diffeomorphism from \(S^{d-1}\) to \(S^{d-1}\).

**Proof.** It suffices to show that \(D\Psi(x)\) is an isomorphism on tangent spaces for all \(x \in S^{d-1}\). From (41) we need to show that if \(\langle y, x \rangle = 0 \) and \(y \neq 0\) then \(\nabla^2 G(x)y\) is not parallel to \(\Psi(x)\). But this follows from the facts that \(\nabla^2 G(x)\) is nonsingular (since \(G\) is strictly convex), and \(\nabla^2 G(x)x = (2n - 1)\nabla G(x)\) is parallel to \(\Psi(x)\).

**Lemma 4.8.** Assume the setting of (32)-(36). Let \(x_* \in S^{d-1}\) be a first-order critical point of (32) but not a second-order point. Then there exist an open neighborhood \(B_{x_*} \subset S^{d-1}\) of \(x_*\) and a smoothly embedded disk \(D_{x_*}\) containing \(x_*\) of dimension strictly less than \(d - 1\) satisfying
\[
(\Psi^k(x) \in B_{x_*} \forall k \geq 0) \Rightarrow x \in D_{x_*}.
\]

**Proof.** By assumption, \(\text{Hess} F(x_*)\) has an eigenvalue which is strictly positive. Then (42) implies \(D\Psi(x_*)\) has an eigenvalue that exceeds 1, using \(F(x_*) + \gamma = G(x_*) > 0\) from strict convexity of \(G\). The conclusion is now immediate from the center-stable manifold theorem; see Theorem B.2 in which we take \(M = S^{d-1}\).

We have all ingredients to prove that SS-HOPM almost always converges to stable eigenvectors.

**Theorem 4.9** (Almost always convergence of SS-HOPM to second-order critical point). Assume the setting of (32)-(36). Then there is a Lebesgue-measure zero subset \(\Omega \subset S^{d-1}\) such that for all initializations \(x_1 \in S^{d-1} \setminus \Omega\), the sequence (33) converges to a second-order critical point of (32).

**Proof.** By Theorem 4.5 for all initializations \(x_1\) the sequence (33) converges to a first-order critical point \(x_*\) of (32). So it suffices to show the set \(\Omega\) of “bad” initializations converging to a first but not second-order critical point is measure zero.

Let \(x_*\) be a first but not second-order critical point of (32), and take \(D_{x_*} \subset B_{x_*}\) as in Lemma 4.8. Suppose \(x_1 \in S^{d-1}\) is such that \((\Psi^k(x_1))_{k \geq 0}\) converges to \(x_*\). Then there exists \(K \geq 0\) so that \(\Psi^k(x_1) \in B_{x_*}\) for all \(k \geq K\). Then \(\Psi^K(x_1) \in D_{x_*}\) by (43); equivalently \(x_1 \in \Psi^{-K}(D_{x_*})\) where \(\Psi^{-K}\) denotes preimage under \(\Psi^K\).

Ranging over \(x_*\) and \(K\), we deduce
\[
\Omega \subset \bigcup_{x_* \text{ first but not second-order critical}} \bigcup_{K \geq 0} \Psi^{-K}(D_{x_*})
\]
(44)

By second-countability of \(S^{d-1}\), we can pass to a countable subcover, i.e., there exists a countable set \(C\) of first but not second-order critical points of (32) such that the right-hand side of (44) equals
\[
\bigcup_{x_* \in C} \bigcup_{K \geq 0} \Psi^{-K}(D_{x_*}).
\]
(45)
Notice that each $D_{x_i}$ has measure zero in $S^{d-1}$ since its dimension is strictly less than $d - 1$. Further, each of $\Psi^{-K}(D_{x_i})$ has measure zero because $\Psi^K$ is a local diffeomorphism by Lemma 4.7. Thus (45) has measure zero, being a countable union of measure zero subsets, and therefore $\Omega$ has measure zero too.

The results on SS-HOPM from this subsection may be of independent interest.

4.3 Local linear convergence to $\pm a_i$

We return to SPM specifically, and prove the local attractiveness claim in Theorem 4.1.

**Theorem 4.10.** Let $r \leq (d+n-1) - d + 1$, and $a_1, \ldots, a_r \in S^{d-1}$ be Zariski-generic. Then the POWER METHOD (26) has local linear convergence to each of $\pm a_i$.

**Proof.** Denote (26) as $x_{k+1} = \Psi(x_k)$ like in Subsection 4.2 but with $F = F_A$ where $A = \text{span}\{a_1^{\otimes n}, \ldots, a_r^{\otimes n}\}$. Clearly $\pm a_i$ are fixed points of $\Psi$, thus by [53, p. 18] we need to show that $D\Psi(\pm a_i)$ has spectral norm strictly less than 1. For notational ease, we just consider $a_i$ as the result for $a_i$ will be immediate by evenness of $F_A$.

From (42) and (28),

$$D\Psi(a_i) = \frac{1}{2n(1 + \gamma)}((I - a_ia_i^\top)\nabla^2 F_A(a_i)(I - a_ia_i^\top) + 2n\gamma(I - a_ia_i^\top)),$$

where we used $F_A(a_i) = 1$ and $a_i^\top \nabla F_A(a_i) = 2nF_A(a_i) = 2n$. Notice that $D\Psi(a_i)$ is self-adjoint. Thus to bound its spectral norm, we can show that for all $y$ with $\langle y, a_i \rangle = 0$ and $\|y\| = 1$ it holds $|y^\top D\Psi(a_i)y| < 1$. Inserting (52) into (46),

$$|y^\top D\Psi(a_i)y| = \frac{1}{2n(1 + \gamma)}(y^\top \nabla^2 F_A(a_i)y + 2\gamma)$$

$$= \frac{1}{1 + \gamma} (n\|P_A(a_i^{\otimes(n-1)} \otimes y)\|^2 + (n - 1)\langle P_A(a_i^{\otimes n}), a_i^{\otimes(n-2)} \otimes y^{\otimes 2} \rangle + \gamma)$$

$$= \frac{1}{1 + \gamma} (n\|P_A(a_i^{\otimes(n-1)} \otimes y)\|^2 + (n - 1)\langle a_i^{\otimes n}, a_i^{\otimes(n-2)} \otimes y^{\otimes 2} \rangle + \gamma)$$

$$= \frac{1}{1 + \gamma} (n\|P_A(\text{sym}(a_i^{\otimes(n-1)} \otimes y))\|^2 + \gamma),$$

where we used $P_A(a_i^{\otimes n}) = a_i^{\otimes n}$, that $P_A$ is a projector onto a subspace of symmetric tensors, and $\langle a_i^{\otimes n}, a_i^{\otimes(n-2)} \otimes y^{\otimes 2} \rangle = \langle a_i, a_i \rangle^{n-2}\langle a_i, y \rangle^2 = 0$ by Lemma 2.1. Therefore

$$|y^\top D\Psi(a_i)y| \leq \frac{1}{1 + \gamma} (n\|\text{sym}(a_i^{\otimes(n-1)} \otimes y)\|^2 + \gamma) = 1,$$

with equality if and only if $\text{sym}(a_i^{\otimes(n-1)} \otimes y) \in A$. Thus the theorem follows from the proposition below.

**Proposition 4.11.** Assume $r \leq (d+n-1) - d + 1$. Let $a_1, \ldots, a_r \in S^{d-1}$ be Zariski-generic and $A = \text{span}\{a_1^{\otimes n}, \ldots, a_r^{\otimes n}\}$. Then for each $i$ it holds

$$\{\text{sym}(a_i^{\otimes(n-1)} \otimes y) : \langle y, a_i \rangle = 0\} \cap A = \emptyset.$$
Proof. Without loss of generality, $i = 1$. A point in the intersection is of the form

$$\alpha_1 a_1^{\otimes n} + \alpha_2 a_2^{\otimes n} + \ldots + \alpha_r a_r^{\otimes n} = \text{sym}(y \otimes a_1^{\otimes (n-1)})$$  \hspace{1cm} (47)

for some $\alpha_j \in \mathbb{R}$ and $y$ with $\langle y, a_i \rangle = 0$. Write $\pi$ for the projector from $T^m_d$ to the orthogonal complement of $\text{sym}(\mathbb{R}^d \otimes a_1^{\otimes (n-1)})$, and apply it to (47) to obtain

$$\alpha_2 \pi(a_2^{\otimes n}) + \ldots + \alpha_r \pi(a_r^{\otimes n}) = 0.$$  \hspace{1cm} (48)

Note that $\pi(a_2^{\otimes n}), \ldots, \pi(a_r^{\otimes n})$ are $r - 1$ generic points in the closure of the projection of the affine cone over the Veronese variety under $\pi$, and that this variety linearly spans its ambient space which has linear dimension $(d+n-1) - d$. Since $r - 1 \leq (d+n-1) - d$ by assumption, $\pi(a_i^{\otimes n})$ are linearly independent. Hence (48) implies $\alpha_2 = \ldots = \alpha_r = 0$. Returning to (47), it follows $\alpha_1 = 0$ and $y = 0$, hence the intersection is zero. □

We remark that Proposition 4.11 has a geometrical interpretation. Namely, when the projectivization of $A$ and the Veronese variety are expected to have a zero-dimensional intersection, they in fact intersect transversely at each of $a_i^{\otimes n}$ (see [25]).

4.4 Putting it together

To sum up, the convergence analysis is complete.

Proof of Theorem 4.1. That $\gamma > \sqrt{\frac{n-1}{n}}$ is a sufficient shift to guarantee strict convexity follows from Lemma 4.3. The first bullet of Theorem 4.1 follows from Theorem 4.11. The second bullet follows from Theorem 4.9. The first sentence of the third bullet is due to Proposition 3.3, and the rest is by Theorem 4.10. □

Beyond the results proven here, in the next section we see empirically that the Power Method is robust to noise (see Subsection 5.3) and that it very often converges to global maxima (see Subsection 5.4).

5 Numerical Experiments

In this section, we perform various numerical tests of SPM and provide comparisons of runtime, accuracy and noise stability against existing state-of-the-art algorithms for symmetric CP decomposition. We compare against FOOBI [19], the method described in the paper [46] abbreviated here as LRSTA, and the method $\text{ccpd}_\text{nls}$ in the Tensorlab package [60]. The first two methods are considered leading among provable algorithms, while the latter is a leading method among heuristic algorithms. We use the most-square matrix flattening in the EXTRACT SUBSPACE step of SPM (i.e., $n = \frac{m}{2}$ in (8)). For the fairest comparison we provide all methods with the ground-truth rank, although this can be computed by SPM and FOOBI (see Subsection 5.3). All numerical experiments are performed on a personal laptop with an Intel® Core™ i7-7700HQ CPU and 16.0GB of RAM. The implementation of the Subspace Power Method is available at https://www.github.com/joaompereira/SPM.
Remark 5.1. For a modest performance benefit, for the experiments we implement SPM using adaptive shifts $\gamma_k$, rather than a constant step size $\gamma$. Similarly to [37], we choose a smaller shift at $x_k$ according to how close $F_A(x)$ is to being locally convex around $x_k$. Specifically, by Lemma 4.4 we modify Power Method to the following:

$$x_{k+1} = \frac{P_A(x_k^n) \cdot x_k^{(n-1)} + \gamma_k x_k}{\|P_A(x_k^n) \cdot x_k^{(n-1)} + \gamma_k x_k\|}$$

where $\gamma_k = \sqrt{\frac{n-1}{n}} h(F_A(x_k))$, (49)

with $h$ defined as in Lemma 4.4. This leads to a slight improvement but doesn’t affect the results qualitatively.

5.1 Breakdown of SPM’s performance

Before comparing with other methods, we first illustrate the performance of SPM on different randomly generated low-rank tensors. We apply SPM to ten tensors $T_1, \ldots, T_{10} \in S_m^n$, generated as follows:

- For tensors $T_1, \ldots, T_6, T_9$ and $T_{10}$, we plant a low-rank rank decomposition (1) with $d, m, r$ as in Table 1, by drawing $\{a_i\}_{r=1}^r$ independently and from the uniform distribution on the unit sphere in $\mathbb{R}^d$, and by drawing $\{\lambda_i\}_{r=1}^r$ independently from the standard Gaussian distribution on $\mathbb{R}$.

- For tensors $T_7$ and $T_8$, we plant a low-rank rank decomposition (1) with correlated components. Specifically, we draw $\{\lambda_i\}_{r=1}^r$ as above, and the vectors $\{a_i\}_{r=1}^r$ are drawn independently but now from different distributions:
  - For $T_7$, we sample a vector $v$ from a Gaussian distribution with identity covariance matrix and mean vector the all-ones vector, and set $a_i = v / \|v\|$. The expected correlation between the CP components is $1/2$.
  - For $T_8$, we sample each $\{a_i\}_{r=1}^r$ from the uniform distribution on the intersection of the positive orthant and the unit sphere in $\mathbb{R}^d$.

The results are displayed in Table 1. For each example tensor $T_1, \ldots, T_{10}$, we record the time per each step of SPM, as well as the total time. We also record the average number of iterations per CP component of the power method, and the total number of times the Power Method step was restarted because the function value was less than $1 - 10^{-6}$. Finally, we report the error of the decomposition of the tensor $T$ obtained by SPM, $(\hat{\lambda}_i, \hat{a}_i)_{r=1}^r$, in terms of the Frobenius norm of the tensor residual:

$$\text{error} := \|T - \hat{T}\| \quad \text{where} \quad \hat{T} = \sum_{i=1}^r \hat{\lambda}_i \hat{a}_i^{\otimes m}$$

Inspecting Table 1 it stands out that most of the time Power Method converges to a global maximum on the first try and usually in $\approx 100$ iterations. Also, the error of the decomposition on exactly low rank tensors is quite low, which suggests that SPM is numerically stable (see below for experiments with noise in the input $\mathcal{T}$).

It is also noticeable that SPM seems to take more iterations for random tensors with correlated entries than for those with uncorrelated entries (specifically, see $T_2$ versus $T_7$ and $T_8$). This is
possibly due to a higher condition number [9] for the symmetric tensor decomposition problem in the case of tensor correlated components. We expect this corresponds to the optimization landscape for \( F_A \) being “more flat” in the vicinity of \( \pm a_i \) when the components are correlated. Further, when comparing the average number of iterations for \( J_2 \) through \( J_6 \), we observe it increases with the rank and decreases with the dimension. Based on these and other empirical observations, the average number of iterations appears to be an increasing function of \( r/d^n \). We suspect that this is also related to the condition number of the tensor.

Finally, comparing \( J_3, J_5 \) and \( J_6 \), Table 1 illustrates that the time of the Extract Subspace step significantly increases with the length \( d \). This occurs because in this step we calculate either SVD or eigendecompositions of an \( d \times d \) matrix. As \( d \) increases, calculating the decomposition becomes a bottleneck for the SPM algorithm, and procedures that compute only the top eigenvectors (e.g., the routine eigs in Matlab) or methods from randomized linear algebra should be considered.

### 5.2 Runtime comparison

We compare computation times for computing CP decompositions by SPM, FOOBI [19], the Tensorlab package [60] and LRSTA [46].

While in [19] there are two variants of FOOBI, we consider only the fastest method (FOOBI-1), and use our own further optimized implementation of the algorithm. Tensorlab has two specialized algorithms for symmetric tensor decompositions, \texttt{ccpd_nls} and \texttt{ccpd_minf}. Our experiments focus on \texttt{ccpd_nls}, since it always performed better than \texttt{ccpd_minf} in our comparisons. The \texttt{ccpd_nls} function employs second-order methods to solve the non-convex least-squares optimization problem:

\[
\argmin_{\mathbf{A}} \| \mathbf{J} - \sum_{i=1}^{r} a_i \otimes a_i \|_F^2,
\]

where \( a_i \) are the columns of \( \mathbf{A} \) which are generally not unit-norm in \texttt{ccpd_nls}. We set the maximum

| \( J \) | \( m \) | \( d \) | \( r \) | time (s) | time (s) | time (s) | avg. # iter. / restarts | error |
|---|---|---|---|---|---|---|---|---|
| \( J_1 \) | 3 | 136 | 136 | 0.06 | 1.69 | 1.15 | 2.90 | 26 / 0 | \( 5.16 \times 10^{-13} \) |
| \( J_2 \) | 4 | 40 | 200 | 0.05 | 0.48 | 0.22 | 0.75 | 45 / 0 | \( 1.74 \times 10^{-13} \) |
| \( J_3 \) | 4 | 40 | 400 | 0.06 | 2.43 | 0.96 | 3.44 | 70 / 0 | \( 5.45 \times 10^{-13} \) |
| \( J_4 \) | 4 | 40 | 600 | 0.06 | 15.36 | 2.67 | 18.09 | 119 / 0 | \( 1.70 \times 10^{-12} \) |
| \( J_5 \) | 4 | 60 | 400 | 0.63 | 4.24 | 1.90 | 6.77 | 40 / 0 | \( 5.26 \times 10^{-13} \) |
| \( J_6 \) | 4 | 80 | 400 | 3.51 | 6.49 | 3.18 | 13.18 | 33 / 0 | \( 2.97 \times 10^{-13} \) |
| \( J_7 \) | 4 | 40 | 200^2 | 0.05 | 2.44 | 0.23 | 2.72 | 244 / 1 | \( 1.73 \times 10^{-12} \) |
| \( J_8 \) | 4 | 40 | 200^4 | 0.05 | 4.92 | 0.21 | 5.18 | 546 / 9 | \( 1.54 \times 10^{-12} \) |
| \( J_9 \) | 5 | 19 | 190 | 0.02 | 1.46 | 0.82 | 2.29 | 35 / 0 | \( 1.15 \times 10^{-13} \) |
| \( J_{10} \) | 6 | 11 | 250 | 0.01 | 3.01 | 0.30 | 3.31 | 228 / 0 | \( 2.73 \times 10^{-12} \) |
number of the second-order iterations to 500, and the gradient and function tolerances (as stop criteria) to $1 \times 10^{-12}$ and $1 \times 10^{-24}$ respectively, in order to obtain $\mathbf{A}$ up to an error bounded by $10^{-12}$. For LRSTA, we use the Matlab code available on the author Nie’s webpage, with its default parameters.

In Figure 2a, we plot the computation time (in seconds) for computing the CP decomposition of $\mathcal{T}$ by SPM, LRSTA and Tensorlab as a function of $d$. For this experiment, we set $m = 3$. FOOBI is omitted from this plot as it does not apply to third-order tensors. For several values of $d$ ranging from 10 to 300, we generated $\mathcal{T}$ as follows. The rank $r$ is set as $d$, and for each $i = 1, \ldots, r$ we independently sample a vector $\mathbf{v}$ from a standard multivariate Gaussian distribution, put $\lambda_i = \|\mathbf{v}\|^m$ and $\mathbf{a}_i = \mathbf{v}/\|\mathbf{v}\|$, and then form $\mathcal{T}$ as in (1). To obtain more accurate runtimes, we have computed the CP decomposition of each tensor 5 times with SPM and LRSTA, and 100 times with Tensorlab (the runtime of Tensorlab varies considerably between runs, so we run it more times to estimate its average computation time). For each method we report the average and the standard deviation of the runtimes, as well as the frequency of runs when the method converged to the right tensor decomposition. We say that it converged if $\|\mathcal{T} - \hat{\mathcal{T}}\|_2 < 10^{-9}$.

In Figure 2b, we plot a similar comparison for SPM, FOOBI and Tensorlab. We set $m = 4$, we generated $\mathcal{T}$ as above, but with $d$ ranging from 10 to 55 and $r = \lfloor d^2/3 \rfloor$. (We do not include LRSTA in this comparison as it is not able to decompose tensors of order 4 with such high rank.) Akin to the previous comparison, we computed the CP decomposition of each tensor 5 times with SPM and with FOOBI, 100 times with Tensorlab, and report the average, the standard deviation and convergence frequency.

A main takeaway from Figure 2 is that SPM outperforms the other algorithms in terms time, at least for this level of rank (notice the logarithmic scales in the plots). In Figure 2b, SPM is, on average, 20 times faster than Tensorlab for tensors of order 4, which in terms of empirical performance, is a leading package for tensor decomposition in Matlab. We also observe that, while FOOBI’s complexity is polynomial in $d$, the exponent is high ($O(r^4d^2)$), which makes this algorithm slower than alternatives especially for large fourth-order tensors. For tensors of order 3

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure2.png}
\caption{Comparison of the computation time between various CP decomposition algorithms. We plot the average computation time as curves, and the shaded areas correspond to the 20% to 80% quantiles. The percentages in the legends denote the average fraction of runs each algorithm returned a correct decomposition.}
\end{figure}
5.3 Noise stability

Here we compare the sensitivity to noise of SPM, FOOBI and Tensorlab. Most tensors that arise in real-data applications are not exactly low-rank, and therefore knowing that SPM works for approximately low-rank tensors is important. For FOOBI and SPM we use the best rank-$r$ approximation of the tensor flattening, computed from the truncated SVD (or eigendecomposition when applicable). For the experiment we fixed $m = 4$, $d = 15$, $r = 30$, generated $\mathcal{T}$ as in Subsection 5.2 and added independent centered Gaussian noise to the entries of $\mathcal{T}$, with standard deviation ranging from $10^{-4}$ to 1, in a symmetric manner.

We measured the error using formula (50), and display the results in Figure 3. We also report the noise stability for noise levels below $10^{-1}$, calculated by averaging the error divided by the noise variance.

As can be observed in Figure 3 Tensorlab is the most stable to noise, which may be expected since it explicitly minimizes the $\ell^2$-norm. However, the sensitivity to noise of SPM and FOOBI are comparable to Tensorlab up to the noise level $10^{-1}$, with SPM’s errors being slightly smaller than FOOBI’s. For larger noise, SPM and FOOBI are less accurate. We suspect that this is related to a spiked eigenvalue model for the flattening: at that noise level, the eigenvectors coming from noise start to overwhelm eigenvectors of the flattening of the signal. At such noise levels, methods that rely on spectral properties of the flattening, like FOOBI and SPM, start to fail.

5.4 Optimization landscape

In this subsection we do an experiment to test the optimization landscape arising in the Power Method step. Recalling that we showed the Power Method iterates converge to a second order critical point of (25), we study how often these iterates converge to global maxima as compared
Figure 4: Relative frequency of convergence of the POWER METHOD at different ranks $r$ over 10000 trials when $n = 2$ and $d = 20$. Each run, the CP components are random and we use only one random initialization. The dashed line is the threshold given by Proposition 3.2.

To spurious second-order critical points (that is, ones that are not global maxima). When the algorithm converges to a second order critical point $x_\ast \in \mathbb{S}^{d-1}$, we know it is a global maximum if and only if $F_A(x_\ast) = 1$. By Proposition 3.2, we know up to which rank it holds that the global maxima of (25) are generically precisely the desired tensor components (up to sign).

Our experiment is as follows. We consider fourth-order tensors ($m = 4$ and $n = 2$), fix the length $d = 20$ and vary the CP rank $r$ from 120 to 200. Note that the rank threshold given by Proposition 3.2 is $r = \binom{21}{2} = 190$. For each rank, we sample $r$ independent and identically distributed standard Gaussian vectors $\{a_i\}_{i=1}^r$ and form the corresponding subspace $A = \text{span}\{a_i^\otimes 2, i = 1, \ldots, r\}$. Since the experiment just tests POWER METHOD, there is no need to generate coefficients $\lambda_i$ because the POWER METHOD just operates on $A$. Next we sample a single initialization vector $x_0$ uniformly on $\mathbb{S}^{d-1}$ and run POWER METHOD starting at $x_0$. We regenerate $A$ and $x_0$ ten thousand times, and report the relative frequency of times the POWER METHOD:

- converged to a tensor component: $\|x_\ast - y\| \leq 10^{-10}$ for some $y \in \{\pm a_i\}_{i=1}^r$;
- converged to a second-order critical point $x_\ast$ that is not a global maximum: $F_A(x_\ast) < 1 - 10^{-10}$;
- converged to a global maximum $x_\ast$ that is not a tensor component: $F_A(x_\ast) \geq 1 - 10^{-10}$ and $\|x_\ast - y\| > 10^{-10}$ for all $y \in \{\pm a_i\}_{i=1}^r$;
- did not converge in 5000 power method iterations: $\|x_{5000} - x_{4999}\| > 10^{-10}$.

The results are shown in Figure 4. For all values of $r$ smaller than 140, POWER METHOD converges to one of the tensor’s components every time over the 10000 trials. Between ranks 140 to 160 the frequency is at least 99.8%, and it decreases to 97.6% at $r = 170$. Then, we observe a sharp transition when the rank varies between 170 and 190. Near the cutoff, many runs do
not converge. Though by Theorem 4.1 the runs eventually would converge, Power Method needs more than 5000 iterations. Note the width of the transition in Figure 4 is \( \approx d \). The results suggest that if the rank scales like \( c_n d^n \), for a constant \( c_n < \frac{1}{n!} \) (since \( \binom{d+n-1}{n} \approx \frac{1}{n!} d^n \)), Power Method converges to a CP component from a single initialization, with high probability. Thus the optimization landscape for (25) seems well-behaved.

In [33] we investigate this phenomenon and characterize local maxima of the SPM functional. There we show that if \( \{a_i\}_{i=1}^r \) are drawn i.i.d. uniformly from \( S^{d-1} \), then with high probability should spurious local maxima exist their function value is rather small: \( O(r \log^n(r)/d^n) \). We also provide guarantees under deterministic frame conditions, holding for example when \( a_i \) are approximately orthogonal.

6 Discussion

This paper introduced a new algorithm, called SPM, for low-rank symmetric tensor CP decomposition. The new algorithm is performant: it outperforms state-of-the-art algorithms notably, at least for moderate ranks. Another advantage is that SPM does not require prior knowledge of the target rank. The algorithm brings together ideas from algebraic geometry and nonconvex optimization. As such, we were able to establish a rich mathematical foundation for its properties.

Several aspects of SPM warrant further analysis. We study the optimization landscape of SPM (25) in a follow-up paper [33], where we obtained guarantees for low-rank and approximately low-rank tensors in a random overcomplete setting as well as under deterministic conditions. As another direction, there are many algorithmic extensions of SPM worth exploring, including using it for decomposing non-symmetric tensors and calculating a block term decomposition. We also want to extend it to settings where the tensor is presented in a specialized manner, for example as a moment tensor of given data samples [55]. The idea would be exploit the structure, especially to reduce storage costs and the upfront expense of EXTRACT SUBSPACE.

Acknowledgements

We thank Emmanuel Abbe, Amir Ali Ahmadi, Tamir Bendory, Nicolas Boumal, Tamara G. Kolda, and Amit Singer for helpful comments. JK was supported in part by the Simons Collaboration on Algorithms and Geometry. JP was supported in part by NSF BIGDATA IIS-1837992. JK and JP were supported in part by start-up grants to JK from the College of Natural Sciences and Oden Institute at UT Austin.

References

[1] P.-A. Absil, R. Mahony, and J. Trumpf, An extrinsic look at the Riemannian Hessian, in International Conference on Geometric Science of Information, Springer Berlin Heidelberg, 2013, pp. 361–368.

[2] A. Anandkumar, R. Ge, D. J. Hsu, S. M. Kakade, and M. Telgarsky, Tensor decompositions for learning latent variable models, Journal of Machine Learning Research, 15 (2014), pp. 2773–2832.
[3] E. Ballico, *On the weak non-defectivity of Veronese embeddings of projective spaces*, Central European Journal of Mathematics, 3 (2005), pp. 183–187.

[4] P. J. Basser and S. Pajevic, *Spectral decomposition of a 4th-order covariance tensor: Applications to diffusion tensor MRI*, Signal Processing, 87 (2007), pp. 220–236.

[5] C. Beltrán, P. Breiding, and N. Vannieuwenhoven, *Pencil-based algorithms for tensor rank decomposition are not stable*, SIAM Journal on Matrix Analysis and Applications, 40 (2019), pp. 739–773.

[6] A. Bhaskara, A. Chen, A. Perreault, and A. Vijayaraghavan, *Smoothed analysis in unsupervised learning via decoupling*, in 2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS), IEEE, November 2019, pp. 582–610.

[7] N. Boumal, *An introduction to optimization on smooth manifolds*, Cambridge University Press, March 2023.

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

[9] P. Breiding and N. Vannieuwenhoven, *The condition number of join decompositions*, SIAM Journal on Matrix Analysis and Applications, 39 (2018), pp. 287–309.

[10] R. Bro, *PARAFAC. Tutorial and applications*, Chemometrics and Intelligent Laboratory Systems, 38 (1997), pp. 149–171.

[11] J.-F. Cardoso, *Super-symmetric decomposition of the fourth-order cumulant tensor. blind identification of more sources than sensors*, in International Conference on Acoustics, Speech, and Signal Processing (ICASSP), IEEE, 1991, pp. 3109–3112.

[12] ———, *Blind signal separation: Statistical principles*, Proceedings of the IEEE, 86 (1998), pp. 2009–2025.

[13] 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.

[14] P. Chevalier, *Optimal separation of independent narrow-band sources: Concept and performance*, Signal Processing, 73 (1999), pp. 27–47.

[15] P. Chevalier, L. Albera, A. Ferréol, and P. Comon, *On the virtual array concept for higher order array processing*, IEEE Transactions on Signal Processing, 53 (2005), pp. 1254–1271.

[16] L. Chiantini and C. Ciliberto, *Weakly defective varieties*, Transactions of the American Mathematical Society, 354 (2001), pp. 151–178.

[17] L. Chiantini, G. Ottaviani, and N. Vannieuwenhoven, *On generic identifiability of symmetric tensors of subgeneric rank*, Transactions of the American Mathematical Society, 369 (2016), pp. 4021–4042.
[18] M. T. Chu, R. E. Funderlic, and G. H. Golub, *A rank–one reduction formula and its applications to matrix factorizations*, SIAM Review, 37 (1995), pp. 512–530.

[19] L. De Lathauwer, J. Castaing, and J.-F. Cardoso, *Fourth-order cumulant-based blind identification of underdetermined mixtures*, IEEE Transactions on Signal Processing, 55 (2007), pp. 2965–2973.

[20] M. C. Dogan and J. M. Mendel, *Applications of cumulants to array processing. I. Aperture extension and array calibration*, IEEE Transactions on Signal Processing, 43 (1995), pp. 1200–1216.

[21] M. Fornasier, J. Vybíral, and I. Daubechies, *Robust and resource efficient identification of shallow neural networks by fewest samples*, Information and Inference: A Journal of the IMA, 10 (2021), pp. 625–695.

[22] M. Fornasier, J. Vybíral, and I. Daubechies, *Robust and resource efficient identification of shallow neural networks by fewest samples*, Information and Inference: A Journal of the IMA, 10 (2021), pp. 625–695.

[23] M. Fornasier, J. Vybíral, and I. Daubechies, *Robust and resource efficient identification of shallow neural networks by fewest samples*, Information and Inference: A Journal of the IMA, 10 (2021), pp. 625–695.

[24] M. Fornasier, J. Vybíral, and I. Daubechies, *Robust and resource efficient identification of shallow neural networks by fewest samples*, Information and Inference: A Journal of the IMA, 10 (2021), pp. 625–695.

[25] M. Fornasier, J. Vybíral, and I. Daubechies, *Robust and resource efficient identification of shallow neural networks by fewest samples*, Information and Inference: A Journal of the IMA, 10 (2021), pp. 625–695.

[26] R. A. Harshman, *Foundations of the PARAFAC procedure: Models and conditions for an “explanatory” multi-model factor analysis*, UCLA Working Papers in Phonetics, 16 (1970), pp. 1–84.

[27] R. A. Harshman, *Foundations of the PARAFAC procedure: Models and conditions for an “explanatory” multi-model factor analysis*, UCLA Working Papers in Phonetics, 16 (1970), pp. 1–84.

[28] R. A. Harshman, *Foundations of the PARAFAC procedure: Models and conditions for an “explanatory” multi-model factor analysis*, UCLA Working Papers in Phonetics, 16 (1970), pp. 1–84.

[29] R. A. Harshman, *Foundations of the PARAFAC procedure: Models and conditions for an “explanatory” multi-model factor analysis*, UCLA Working Papers in Phonetics, 16 (1970), pp. 1–84.

[30] R. A. Harshman, *Foundations of the PARAFAC procedure: Models and conditions for an “explanatory” multi-model factor analysis*, UCLA Working Papers in Phonetics, 16 (1970), pp. 1–84.

[31] N. Johnston, B. Lovitz, and A. Vijayaraghavan, *Computing linear sections of varieties: Quantum entanglement, tensor decompositions and beyond*, in 2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS), IEEE, November 2023.
[33] J. Kileel, T. Klock, and J. M Pereira, Landscape analysis of an improved power method for tensor decomposition, in Advances in Neural Information Processing Systems, vol. 34, 2021, pp. 6253–6265.

[34] T. G. Kolda, Numerical optimization for symmetric tensor decomposition, Mathematical Programming, 151 (2015), pp. 225–248.

[35] T. G. Kolda and B. W. Bader, Tensor decompositions and applications, SIAM Review, 51 (2009), pp. 455–500.

[36] T. G. Kolda and J. R. Mayo, Shifted power method for computing tensor eigenpairs, SIAM Journal on Matrix Analysis and Applications, 32 (2011), pp. 1095–1124.

[37] ——, An adaptive shifted power method for computing generalized tensor eigenpairs, SIAM Journal on Matrix Analysis and Applications, 35 (2014), pp. 1563–1581.

[38] J. M. Landsberg, Tensors: Geometry and applications, vol. 128, AMS Graduate Studies in Mathematics, 2012.

[39] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht, Gradient descent only converges to minimizers, in Conference on Learning Theory (COLT), vol. 49, PMLR, 2016, pp. 1246–1257.

[40] S. E. Leurgans, R. T. Ross, and R. B. Abel, A decomposition for three-way arrays, SIAM Journal on Matrix Analysis and Applications, 14 (1993), pp. 1064–1083.

[41] L.-H. Lim, Singular values and eigenvalues of tensors: A variational approach, in 1st IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, 2005., IEEE, 2005, pp. 129–132.

[42] S. Łojasiewicz, Ensembles semi-analytiques, Institute Des Hautes Études Scientifiques (IHES) Notes, Bures-sur-Yvette, France, (1965).

[43] T. Ma, J. Shi, and D. Steurer, Polynomial-time tensor decompositions with sum-of-squares, in 2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS), IEEE, October 2016, pp. 438–446.

[44] Y. Nakatsukasa, T. Soma, and A. Uschmajew, Finding a low-rank basis in a matrix subspace, Mathematical Programming, 162 (2016), pp. 325–361.

[45] J. Nie, Generating polynomials and symmetric tensor decompositions, Foundations of Computational Mathematics, 17 (2015), pp. 423–465.

[46] ——, Low rank symmetric tensor approximations, SIAM Journal on Matrix Analysis and Applications, 38 (2017), pp. 1517–1540.

[47] L. Oeding and G. Ottaviani, Eigenvectors of tensors and algorithms for waring decomposition, Journal of Symbolic Computation, 54 (2013), pp. 9–35.

[48] I. Panageas and G. Piliouras, Gradient descent only converges to minimizers: Non-isolated critical points and invariant regions, in 8th Innovations in Theoretical Computer Science (ITCS), 2017.
A Additional Proofs

A.1 Proof of Lemma 4.4

Proof. The Euclidean Hessian of $F_A$ is given by

$$\nabla^2 F_A(x) = 2n \sum_{i=1}^r n(u_i \cdot x^{\otimes (n-1)})(u_i \cdot x^{\otimes (n-1)})^\top + (n - 1) \langle u_i, x^{\otimes n} \rangle u_i \cdot x^{\otimes (n-2)},$$
where $U_1, \ldots, U_r \in S_d^q$ are an orthonormal basis of $A$. For the rest of the proof denote $xy := x \otimes y$ and $x^n := x \otimes^n$. Let $x, y \in \mathbb{R}^d$ such that $\|x\| = \|y\| = 1$. We have

\[
\frac{1}{2n} y^T \nabla^2 F_A(x)y = n \sum_{i=1}^{r} \langle U_i, x^{n-1}y \rangle^2 + (n-1) \sum_{i=1}^{r} \langle U_i, x^n \rangle \langle U_i, x^{n-2}y^2 \rangle
\]

\[
= n\|P_A(x^{n-1}y)\|^2 + (n-1) \langle P_A(x^n), x^{n-2}y^2 \rangle,
\]

\[
\geq (n-1) \langle P_A(x^n), x^{n-2}y^2 \rangle,
\]

where (52) follows from (4.1). Define $\bar{y} \in \mathbb{R}^d$ and $\alpha, \beta \in \mathbb{R}$ such that $\langle \bar{y}, x \rangle = 0$, $\|\bar{y}\| = 1$, $\alpha^2 + \beta^2 = 1$ and $y = \alpha \bar{y} + \beta x$. Substituting this expression for $y$ into (53), and rearranging using $P_A$ is self-adjoint and $P_A^2 = P_A$,

\[
\frac{1}{2n} y^T \nabla^2 F_A(x)y \geq (n-1) \langle P_A(x^n), \beta^2 x^n + 2\alpha\beta x^{n-1}\bar{y} + \alpha^2 x^{n-2}\bar{y}^2 \rangle.
\]

Define $z = \|P_A(x^n)\|^2 = \langle P_A(x^n), x^n \rangle \geq \nu$. Letting $Y = 2\alpha\beta x^{n-1}\bar{y} + \alpha^2 x^{n-2}\bar{y}^2$, we have

\[
\frac{1}{2n} y^T \nabla^2 F_A(x)y \geq (n-1)\beta^2 z + (n-1) \langle P_A(x^n), Y \rangle
\]

\[
= (n-1)\beta^2 z + (n-1) \langle P_A(x^n), \text{sym}(Y) \rangle,
\]

where the last equation follows from $P_A(x^n)$ being a symmetric tensor and Lemma 2.3. Again using Lemma 2.3, Lemma 2.1 and $\langle \bar{y}, x \rangle = 0$, we obtain $\langle x^n, \text{sym}(Y) \rangle = \langle x^n, Y \rangle = 0$. Thus the following Bessel’s inequality holds:

\[
\|P_A(x^n)\|^2 \geq \langle P_A(x^n), x^n \rangle^2 + \langle P_A(x^n), \frac{\text{sym}(Y)}{\|\text{sym}(Y)\|} \rangle^2,
\]

which implies

\[
\langle P_A(x^n), \text{sym}(Y) \rangle \geq -\sqrt{z-z^2}\|\text{sym}(Y)\|.
\]

We plug this in (54) to obtain

\[
\frac{1}{2n} y^T \nabla^2 f(x)y \geq (n-1)\beta^2 z - (n-1)\sqrt{z-z^2}\|\text{sym}(Y)\|.
\]

Now we calculate

\[
\|\text{sym}(Y)\|^2 = \|2\alpha\beta \text{sym}(x^{n-1}\bar{y}) + \alpha^2 \text{sym}(x^{n-2}\bar{y}^2)\|^2
\]

\[
= 4\alpha^2\beta^2 \|\text{sym}(x^{n-1}\bar{y})\|^2 + 4\alpha^3\beta \langle \text{sym}(x^{n-1}\bar{y}), \text{sym}(x^{n-2}\bar{y}^2) \rangle
\]

\[
+ \alpha^4 \|\text{sym}(x^{n-2}\bar{y}^2)\|^2.
\]

We start by calculating $\|\text{sym}(x^{n-2}\bar{y}^2)\|^2$. Denoting by $\binom{[n]}{2}$ the set of subsets of $\{1, \ldots, n\}$ of cardinality two, we have

\[
\text{sym}(x^{n-2}\bar{y}^2) = \binom{n}{2}^{-1} \sum_{S \in \binom{[n]}{2}} \pi_S(x^{n-2}\bar{y}^2).
\]
By this notation, we mean the following: if we let \( S = \{ s_1, s_2 \} \subset \{ 1, \ldots, n \} \), then \( \pi_S(x^n-2\vec{y}^2) \) permutes the tensor product \( x^n-2\vec{y}^2 \) such that \( \vec{y} \) appears in positions \( s_1 \) and \( s_2 \). Now from Lemma 2.1 \( \langle \vec{y}, x \rangle = 0 \) and \( \| x \| = \| \vec{y} \| = 1 \),

\[
\langle \pi_{S_1}(x^n-2\vec{y}^2), \pi_{S_2}(x^n-2\vec{y}^2) \rangle = \delta_{S_1=S_2},
\]

where \( \delta_{S_1=S_2} = 1 \) if \( S_1 = S_2 \) and 0 otherwise. Thus

\[
\| \text{sym}(x^n-2\vec{y}^2) \|^2 = \left( \frac{n}{2} \right)^{-1} = \frac{2}{n(n-1)}.
\]

Analogously, \( \| \text{sym}(x^n-1\vec{y}) \|^2 = \frac{1}{n} \) and \( \langle \text{sym}(x^n-2\vec{y}^2), \text{sym}(x^n-1\vec{y}) \rangle = 0 \), therefore

\[
\| \text{sym}(Y) \|^2 = \frac{4\alpha^2 \beta^2}{n} + \frac{2\alpha^4}{n(n-1)}.
\] (56)

Now plugging in (56) in (55), letting \( t = \beta^2 \) and noting that \( \alpha^2 = 1 - t \), we have

\[
\frac{1}{2n}y^\top\nabla^2 f(x)y \geq (n-1)tz - (n-1)\sqrt{z-z^2} \sqrt{\frac{4t(1-t)}{n} + \frac{2(1-t)^2}{n(n-1)}}
\]

\[
=: g(t, z) \geq \min_{t \in [0,1]} g(t, z)
\]

We now estimate this minimum. We overview the computation, but omit the full details for brevity. First we check, by calculating the second derivative that, for \( z \) fixed, \( g \) is a convex function of \( t \). Then we find the minimum in \( t \) by solving \( \frac{\partial g}{\partial t} = 0 \), which amounts to solving a quadratic equation in \( t \). Finally, because this is constrained minimization, we have to check if the minimum lies inside the interval, else the minimum is achieved at the boundary. This leads to a case by case expression:

\[
\hat{g}(z) := \min_{t \in [0,1]} g(t, z) = \begin{cases} 
-\sqrt{\frac{2(n-1)z(1-z)}{n}} & \text{if } z \geq \frac{2(n-2)^2}{3n^2-9n+8} \\
\frac{1}{(n-1)} \left( -\sqrt{\frac{(n-1)z(4n-6+3n^2-5n+6)}{n}} - \frac{n-3}{2n-3} \right) & \text{if } z < \frac{2(n-2)^2}{3n^2-9n+8}
\end{cases}
\]

It can be checked by taking derivatives that \( \hat{g}(z) \) is also a convex function, and that we always have \( \frac{2(n-2)^2}{3n^2-9n+8} \leq \frac{2}{3} \). Furthermore, \( z \geq \nu \), hence

\[
\frac{1}{2n}y^\top\nabla^2 F_A(x)y \geq \hat{g}(z) \geq \min_{z \in [\nu,1]} \hat{g}(z)
\]

\[
\begin{cases} 
\hat{g}(\nu) & \text{if } \nu > \frac{2}{3} \\
\hat{g}\left( \frac{2}{3} \right) + (\nu - \frac{2}{3})\hat{g}'\left( \frac{2}{3} \right) & \text{if } \nu \leq \frac{2}{3}
\end{cases}
\]

\[
= -\sqrt{\frac{n-1}{n}} h(\nu),
\] (57)

where \( h(\nu) \) is defined in (31). Therefore (30) holds.

For the last sentence of Lemma 4.4 by homogeneity it suffices to check that for \( x \in \mathbb{S}^{d-1} \) the eigenvalues of \( \nabla^2 (F_A(x) + \gamma (x^\top x)^n) \) are strictly positive. We compute

\[
\nabla^2 (\gamma (x^\top x)^n) = 4n(n-1)\gamma (x^\top x)^{n-2}xx^\top + 2n\gamma \| x \|^{2n-2}I = 4n(n-1)\gamma xx^\top + 2n\gamma I.
\] (58)

32
From (58) and (57) we obtain
\[
\frac{1}{2n} y^T \nabla^2 (F_A(x) + \gamma (x^T x)^n) y \geq -\sqrt{\frac{n-1}{n}} h(\nu) + \gamma \geq -\sqrt{\frac{n-1}{n}} + \gamma,
\]
which is strictly positive if \( \gamma > \sqrt{\frac{n-1}{n}} \). This finishes the proof of Lemma 4.4. \( \square \)

### A.2 Proof of Lemma 4.6

**Proof.** Let \( x \in S^{d-1} \). Simple calculations show that
\[
\frac{\partial \Psi_x}{\partial x_j} = \frac{\partial}{\partial x_j} \left( \frac{\partial G(x)}{\|\nabla G(x)\|} \right) = \left( \frac{\partial^2 G(x)}{\|\nabla G(x)\|} \frac{\partial G(x) \partial \nabla G(x)}{\partial x_j} \frac{\partial G(x)}{\partial x_j} \right) \|\nabla G(x)\|^2,
\]
and
\[
\frac{\partial \|\nabla G(x)\|}{\partial x_j} = \frac{\|\nabla G(x)\|^{-1} \sum_{k=1}^d \frac{\partial G(x) \partial G(x)}{\partial x_j \partial x_k}}{\|\nabla G(x)\|^2} \frac{\partial^2 G(x) \partial G(x)}{\partial x_j \partial x_k} = \left( \frac{\nabla^2 G(x) \nabla G(x)}{\partial x_j} \right)_{ij}.
\]

Therefore,
\[
\frac{\partial \Psi_x}{\partial x_j} = \left( \frac{\nabla^2 G(x)}{\|\nabla G(x)\|^3} \right) \frac{\nabla^2 G(x)}{\partial x_j} = \left( (I - \Psi(x) \Psi(x)^T) \frac{\nabla^2 G(x)}{\|\nabla G(x)\|} \right)_{ij}.
\]

Viewing \( D\Psi \) as a linear map between tangent spaces to the sphere, its domain is the tangent space to \( S^{d-1} \) at \( x \). By a harmless abuse we express this by right-multiplying with the projector onto this tangent space. It yields
\[
D\Psi(x) = (I - \Psi(x) \Psi(x)^T) \frac{\nabla^2 G(x)}{\|\nabla G(x)\|}(I - xx^T),
\]
as wanted.

Next let \( x_* \in S^{d-1} \) be a first-order critical point of (32). Then \( \Psi(x_*) = x_* \), thus
\[
D\Psi(x) = (I - x_* x_*^T) \frac{\nabla^2 G(x_*)}{\|\nabla G(x_*\|}(I - x_* x_*^T).
\]

(59)

Also \( x_* \) is a first-order critical point of (37). So the Riemannian gradient satisfies \( \nabla G(x_*) = 0 \), implying (see (27)) \( \nabla G(x_*) = (x_*^T \nabla G(x_*))^T x_* = 2nG(x_*)x_* = 2n(F(x_*) + \gamma)x_* \), whence \( \|\nabla G(x_*)\| = 2n(F(x_*) + \gamma) \). We also have \( \nabla^2 G(x_*) = \nabla^2 (F(x_*) + \gamma (x_*^T x_*)^n) = \nabla^2 F(x_*) + 4n(n-1)\gamma x_* x_*^T + 2n\gamma I \) by (58). Substituting into (59) we obtain
\[
D\Psi(x_*) = (I - x_* x_*^T) \frac{\nabla^2 F(x_*)}{2n(F(x_*) + \gamma)}(I - x_* x_*^T) + \frac{\gamma}{F(x_*) + \gamma}(I - x_* x_*^T).
\]

(60)

On the other hand, as in (28) the Riemannian Hessian is
\[
\text{Hess } F(x_*) = (I - x_* x_*^T) \nabla^2 F(x_*)(I - x_* x_*^T) - (x_*^T \nabla F(x_*))(I - x_* x_*^T)
\]
\[
= (I - x_* x_*^T) \nabla^2 F(x_*)(I - x_* x_*^T) - 2nF(x_*)(I - x_* x_*^T).
\]

(61)

Comparing (60) and (61) we conclude (42). This proves Lemma 4.6. \( \square \)
B Background Statements

B.1 Convergence result \cite[Thm. 2.3]{54} 

For simplicity we state a special case of Schneider-Uschmajew’s result, which is enough for our purposes.

**Theorem B.1.** \cite[Thm. 2.3]{54} Let $M \subseteq \mathbb{R}^d$ be a compact Riemannian submanifold that is locally the image of a real analytic map out of a Euclidean space. Let $D \subseteq \mathbb{R}^d$ be an open neighborhood of $M$. Suppose $F : D \to \mathbb{R}$ is a real analytic function that is bounded below. Let $\nabla F$ denote the Riemannian gradient of the restriction of $F$ to $M$. Consider the problem

$$\max_{x \in M} F(x).$$  \hfill (62)

Suppose that $(x_k)_{k=1}^\infty \subseteq M$ is a sequence satisfying the following three assumptions:

(A1) There exists $\sigma > 0$ such that for $k$ large enough,

$$F(x_{k+1}) - F(x_k) \geq \sigma \| \nabla F(x_k) \| \| x_{k+1} - x_k \|.$$  

(A2) For large enough $k$,

$$\nabla F(x_k) = 0 \implies x_{k+1} = x_k.$$

(A3) There exists $\rho > 0$ such that for large enough $k$,

$$\| x_{k+1} - x_k \| \geq \rho \| \nabla F(x_k) \|.$$  

Then $(x_k)_{k=1}^\infty$ converges to a point $x_* \in M$ and there exist constants $C > 0$ and $\tau > 1$ such that

$$\| x_k - x_* \| \leq C k^{-\tau}$$  \hfill (63)

for all $k$. Moreover, $\| \nabla F(x_k) \| \to 0$ as $k \to \infty$.

B.2 Center-stable manifold theorem

This is stated for open sets in Euclidean space in \cite{56}. However, by taking charts it holds on manifolds as below.

**Theorem B.2.** \cite[Thm. III.7(2)]{56} Let $M$ be a smooth manifold, $\Psi : M \to M$ a local diffeomorphism, and $x \in M$ a fixed point of $\Psi$. Then there exist an open neighborhood $B_x \subseteq M$ of $x$ and smoothly embedded disk $D_x \subseteq M$ containing $x$ such that the following properties hold:

- $\{ y \in M : \Psi^k(y) \in B_x \ \forall k \geq 0 \} \subseteq D_x$;
- $\dim(D_x)$ is the number of linearly independent eigenvectors of $D\Psi(x)$ whose eigenvalues don’t exceed 1 in magnitude.