

Chapter 14

Low-Rank Approximation of Tensors

Shmuel Friedland* and Venu Tammali

Abstract In many applications such as data compression, imaging or genomic data analysis, it is important to approximate a given tensor by a tensor that is sparsely representable. For matrices, i.e. 2-tensors, such a representation can be obtained via the singular value decomposition, which allows to compute best rank k -approximations. For very big matrices a low rank approximation using SVD is not computationally feasible. In this case different approximations are available. It seems that variants of the CUR-decomposition are most suitable. For d -mode tensors $\mathcal{T} \in \otimes_{i=1}^d \mathbb{R}^{n_i}$, with $d > 2$, many generalizations of the singular value decomposition have been proposed to obtain low tensor rank decompositions. The most appropriate approximation seems to be best (r_1, \dots, r_d) -approximation, which maximizes the ℓ_2 norm of the projection of \mathcal{T} on $\otimes_{i=1}^d \mathbf{U}_i$, where \mathbf{U}_i is an r_i -dimensional subspace \mathbb{R}^{n_i} . One of the most common methods is the *alternating maximization method* (AMM). It is obtained by maximizing on one subspace \mathbf{U}_i , while keeping all other fixed, and alternating the procedure repeatedly for $i = 1, \dots, d$. Usually, AMM will converge to a local best approximation. This approximation is a fixed point of a corresponding map on Grassmannians. We suggest a Newton method for finding the corresponding fixed point. We also discuss variants of CUR-approximation method for tensors. The first part of the paper is a survey on low rank approximation of tensors. The second new part of this paper is a new Newton method for best (r_1, \dots, r_d) -approximation. We compare numerically different approximation methods.

*This work was supported by NSF grant DMS-1216393.

S. Friedland (✉) • V. Tammali

Department of Mathematics, Statistics and Computer Science, University of Illinois at Chicago,
322 Science and Engineering Offices (M/C 249), 851 S. Morgan Street, Chicago, IL 60607-7045,
USA

e-mail: friedlan@uic.edu; vtamma2@uic.edu

14.1 Introduction

Let \mathbb{R} be the field of real numbers. Denote by $\mathbb{R}^{\mathbf{n}} = \mathbb{R}^{n_1 \times \dots \times n_d} := \otimes_{i=1}^d \mathbb{R}^{n_i}$, where $\mathbf{n} = (n_1, \dots, n_d)$, the tensor products of $\mathbb{R}^{n_1}, \dots, \mathbb{R}^{n_d}$. $\mathcal{T} = [t_{i_1, \dots, i_d}] \in \mathbb{R}^{\mathbf{n}}$ is called a *d-mode tensor*. Note that the number of coordinates of \mathcal{T} is $N = n_1 \dots n_d$. A tensor \mathcal{T} is called a *sparsely representable tensor* if it can be represented with a number of coordinates that is much smaller than N .

Apart from sparse matrices, the best known example of a sparsely representable 2-tensor is a low rank approximation of a matrix $A \in \mathbb{R}^{n_1 \times n_2}$. A rank k -approximation of A is given by $A_{\text{appr}} := \sum_{i=1}^k \mathbf{u}_i \mathbf{v}_i^T$, which can be identified with $\sum_{i=1}^k \mathbf{u}_i \otimes \mathbf{v}_i$. To store A_{appr} we need only the $2k$ vectors $\mathbf{u}_1, \dots, \mathbf{u}_k \in \mathbb{R}^{n_1}$, $\mathbf{v}_1, \dots, \mathbf{v}_k \in \mathbb{R}^{n_2}$. A best rank k -approximation of $A \in \mathbb{R}^{n_1 \times n_2}$ can be computed via the *singular value decomposition*, abbreviated here as SVD, [19]. Recall that if A is a real symmetric matrix, then the best rank k -approximation must be symmetric, and is determined by the spectral decomposition of A .

The computation of the SVD requires $\mathcal{O}(n_1 n_2^2)$ operations and at least $\mathcal{O}(n_1 n_2)$ storage, assuming that $n_2 \leq n_1$. Thus, if the dimensions n_1 and n_2 are very large, then the computation of the SVD is often infeasible. In this case other type of low rank approximations are considered, see e.g. [1, 5, 7, 11, 13, 18, 21].

For d -tensors with $d > 2$ the situation is rather unsatisfactory. It is a major theoretical and computational problem to formulate good generalizations of low rank approximation for tensors and to give efficient algorithms to compute these approximations, see e.g. [3, 4, 8, 13, 15, 29, 30, 33, 35, 36, 43].

We now discuss briefly the main ideas of the approximation methods for tensors discussed in this paper. We need to introduce (mostly) standard notation for tensors. Let $[n] := \{1, \dots, n\}$ for $n \in \mathbb{N}$. For $\mathbf{x}_i := (x_{1,i}, \dots, x_{n_i,i})^T \in \mathbb{R}^{n_i}, i \in [d]$, the tensor $\otimes_{i \in [d]} \mathbf{x}_i = \mathbf{x}_1 \otimes \dots \otimes \mathbf{x}_d = \mathcal{X} = [x_{j_1, \dots, j_d}] \in \mathbb{R}^{\mathbf{n}}$ is called a decomposable tensor, or rank one tensor if $\mathbf{x}_i \neq \mathbf{0}$ for $i \in [d]$. That is, $x_{j_1, \dots, j_d} = x_{j_1,1} \dots x_{j_d,d}$ for $j_i \in [n_i], i \in [d]$. Let $\langle \mathbf{x}_i, \mathbf{y}_i \rangle_i := \mathbf{y}_i^T \mathbf{x}_i$ be the standard inner product on \mathbb{R}^{n_i} for $i \in [d]$. Assume that $\mathcal{S} = [s_{j_1, \dots, j_d}]$ and $\mathcal{T} = [t_{j_1, \dots, j_d}]$ are two given tensors in $\mathbb{R}^{\mathbf{n}}$. Then $\langle \mathcal{S}, \mathcal{T} \rangle := \sum_{j_i \in [n_i], i \in [d]} s_{j_1, \dots, j_d} t_{j_1, \dots, j_d}$ is the standard inner product on $\mathbb{R}^{\mathbf{n}}$. Note that

$$\begin{aligned} \langle \otimes_{i \in [d]} \mathbf{x}_i, \otimes_{i \in [d]} \mathbf{y}_i \rangle &= \prod_{i \in [d]} \langle \mathbf{x}_i, \mathbf{y}_i \rangle_i, \\ \langle \mathcal{T}, \otimes_{i \in [d]} \mathbf{x}_i \rangle &= \sum_{j_i \in [n_i], i \in [d]} t_{j_1, \dots, j_d} x_{j_1,1} \dots x_{j_d,d}. \end{aligned}$$

The norm $\|\mathcal{T}\| := \sqrt{\langle \mathcal{T}, \mathcal{T} \rangle}$ is called the Hilbert-Schmidt norm. (For matrices, i.e. $d = 2$, it is called the Frobenius norm.)

Let $I = \{1 \leq i_1 < \dots < i_l \leq d\} \subset [d]$. Assume that $\mathcal{X} = [x_{j_{i_1}, \dots, j_{i_l}}] \in \otimes_{k \in [l]} \mathbb{R}^{n_{i_k}}$. Then the contraction $\mathcal{T} \times \mathcal{X}$ on the set of indices I is given by:

$$\mathcal{T} \times \mathcal{X} = \sum_{j_{i_k} \in [n_{i_k}], k \in [l]} t_{j_1, \dots, j_d} x_{j_{i_1}, \dots, j_{i_l}} \in \otimes_{p \in [d] \setminus I} \mathbb{R}^{n_p}.$$

Assume that $\mathbf{U}_i \subset \mathbb{R}^{n_i}$ is a subspace of dimension r_i with an orthonormal basis $\mathbf{u}_{1,i}, \dots, \mathbf{u}_{r_i,i}$ for $i \in [d]$. Let $\mathbf{U} := \otimes_{i=1}^d \mathbf{U}_i \subset \mathbb{R}^{\mathbf{n}}$. Then $\otimes_{i=1}^d \mathbf{u}_{j_i,i}$, where $j_i \in [n_i], i \in [d]$, is an orthonormal basis in \mathbf{U} . We are approximating $\mathcal{T} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ by a tensor

$$\mathcal{S} = \sum_{j_i \in [r_i], i \in [d]} s_{j_1, \dots, j_d} \mathbf{u}_{j_1,1} \otimes \dots \otimes \mathbf{u}_{j_d,d} \in \mathbb{R}^{\mathbf{n}} \quad (14.1)$$

The tensor $\mathcal{S}' = [s_{j_1, \dots, j_d}] \in \mathbb{R}^{r_1 \times \dots \times r_d}$ is the *core tensor* corresponding to \mathcal{S} in the terminology of [42].

There are two major problems: The first one is how to choose the subspaces $\mathbf{U}_1, \dots, \mathbf{U}_d$. The second one is the choice of the core tensor \mathcal{S}' . Suppose we already made the choice of $\mathbf{U}_1, \dots, \mathbf{U}_d$. Then $\mathcal{S} = P_{\mathbf{U}}(\mathcal{T})$ is the orthogonal projection of \mathcal{T} on \mathbf{U} :

$$P_{\otimes_{i \in [d]} \mathbf{U}_i}(\mathcal{T}) = \sum_{j_i \in [r_i], i \in [d]} \langle \mathcal{T}, \otimes_{i \in [d]} \mathbf{u}_{j_i,i} \rangle \otimes_{i \in [d]} \mathbf{u}_{j_i,i}. \quad (14.2)$$

If the dimensions of n_1, \dots, n_d are not too big, then this projection can be explicitly carried out. If the dimension n_1, \dots, n_d are too big to compute the above projection, then one needs to introduce other approximations. That is, one needs to compute the core tensor \mathcal{S}' appearing in (14.1) accordingly. The papers [1, 5, 7, 11, 13, 18, 21, 29, 33, 35, 36] essentially choose \mathcal{S}' in a particular way.

We now assume that the computation of $P_{\mathbf{U}}(\mathcal{T})$ is feasible. Recall that

$$\|P_{\otimes_{i \in [d]} \mathbf{U}_i}(\mathcal{T})\|^2 = \sum_{j_i \in [r_i], i \in [d]} |\langle \mathcal{T}, \otimes_{i=1}^d \mathbf{u}_{j_i,i} \rangle|^2. \quad (14.3)$$

The best \mathbf{r} -approximation of \mathcal{T} , where $\mathbf{r} = (r_1, \dots, r_d)$, in Hilbert-Schmidt norm is the solution of the minimal problem:

$$\min_{\mathbf{U}_i, \dim \mathbf{U}_i = r_i, i \in [d]} \min_{\mathcal{X} \in \otimes_{i \in [d]} \mathbf{U}_i} \|\mathcal{T} - \mathcal{X}\|. \quad (14.4)$$

This problem is equivalent to the following maximum

$$\max_{\mathbf{U}_i, \dim \mathbf{U}_i = r_i, i \in [d]} \|P_{\otimes_{i \in [d]} \mathbf{U}_i}(\mathcal{T})\|^2. \quad (14.5)$$

The standard *alternating maximization method*, denoted by AMM, for solving (14.5) is to solve the maximum problem, where all but the subspace \mathbf{U}_i is fixed. Then this maximum problem is equivalent to finding an r_i -dimensional subspace of \mathbf{U}_i containing the r_i biggest eigenvalues of a corresponding nonnegative definite matrix $A_i(\mathbf{U}_1, \dots, \mathbf{U}_{i-1}, \mathbf{U}_{i+1}, \dots, \mathbf{U}_d) \in \mathbf{S}_{n_i}$. Alternating between $\mathbf{U}_1, \mathbf{U}_2, \dots, \mathbf{U}_d$ we obtain a nondecreasing sequence of norms of projections which converges to v . Usually, v is a critical value of $\|P_{\otimes_{i \in [d]} \mathbf{U}_i}(\mathcal{T})\|$. See [4] for details.

Assume that $r_i = 1$ for $i \in [d]$. Then $\dim \mathbf{U}_i = 1$ for $i \in [d]$. In this case the minimal problem (14.4) is called a best rank one approximation of \mathcal{T} . For $d = 2$ a best rank one approximation of a matrix $\mathcal{T} = T \in \mathbb{R}^{n_1 \times n_2}$ is accomplished by the first left and right singular vectors and the corresponding maximal singular value $\sigma_1(T)$. The complexity of this computation is $\mathcal{O}(n_1 n_2)$ [19]. Recall that the maximum (14.5) is equal to $\sigma_1(T)$, which is also called the spectral norm $\|T\|_2$. For $d > 2$ the maximum (14.5) is called the spectral norm of \mathcal{T} , and denoted by $\|\mathcal{T}\|_\sigma$. The fundamental result of Hillar-Lim [27] states that the computation of $\|\mathcal{T}\|_\sigma$ is NP-hard in general. Hence the computation of best \mathbf{r} -approximation is NP-hard in general.

Denote by $\text{Gr}(r, \mathbb{R}^n)$ the variety of all r -dimensional subspaces in \mathbb{R}^n , which is called Grassmannian or Grassmann manifold. Let

$$\mathbf{1}_d := (\underbrace{1, \dots, 1}_d), \quad \text{Gr}(\mathbf{r}, \mathbf{n}) := \text{Gr}(r_1, n_1) \times \dots \times \text{Gr}(r_d, n_d).$$

Usually, the AMM for best \mathbf{r} -approximation of \mathcal{T} will converge to a fixed point of a corresponding map $\mathbf{F}_{\mathcal{T}} : \text{Gr}(\mathbf{r}, \mathbf{n}) \rightarrow \text{Gr}(\mathbf{r}, \mathbf{n})$. This observation enables us to give a new Newton method for finding a best \mathbf{r} -approximation to \mathcal{T} . For best rank one approximation the map $\mathbf{F}_{\mathcal{T}}$ and the corresponding Newton method was stated in [14].

This paper consists of two parts. The first part surveys a number of common methods for low rank approximation methods of matrices and tensors. We did not cover all existing methods here. We were concerned mainly with the methods that the first author and his collaborators were studying, and closely related methods. The second part of this paper is a new contribution to Newton algorithms related to best \mathbf{r} -approximations. These algorithms are different from the ones given in [8, 39, 43]. Our Newton algorithms are based on finding the fixed points corresponding to the map induced by the AMM. In general its known that for big size problem, where each n_i is big for $i \in [d]$ and $d \geq 3$, Newton methods are not efficient. The computation associate the matrix of derivatives (Jacobian) is too expensive in computation and time. In this case AMM or MAMM (modified AMM) are much more cost effective. This well known fact is demonstrated in our simulations.

We now briefly summarize the contents of this paper. In Sect. 14.2 we review the well known facts of singular value decomposition (SVD) and its use for best rank k -approximation of matrices. For large matrices approximation methods

using SVD are computationally unfeasible. Section 14.3 discusses a number of approximation methods of matrices which do not use SVD. The common feature of these methods is sampling of rows, or columns, or both to find a low rank approximation. The basic observation in Sect. 14.3.1 is that, with high probability, a best k -rank approximation of a given matrix based on a subspace spanned by the sampled row is within a relative ϵ error to the best k -rank approximation given by SVD. We list a few methods that use this observation. However, the complexity of finding a particular k -rank approximation to an $m \times n$ matrix is still $\mathcal{O}(kmn)$, as the complexity truncated SVD algorithms using Arnoldi or Lanczos methods [19, 31]. In Sect. 14.3.2 we recall the CUR-approximation introduced in [21]. The main idea of CUR-approximation is to choose k columns and rows of A , viewed as matrices C and R , and then to choose a square matrix U of order k in such a way that CUR is an optimal approximation of A . The matrix U is chosen to be the inverse of the corresponding $k \times k$ submatrix A' of A . The quality of CUR-approximation can be determined by the ratio of $|\det A'|$ to the maximum possible value of the absolute value of all $k \times k$ minors of A . In practice one searches for this maximum using a number of random choices of such minors. A modification of this search algorithm is given in [13]. The complexity of storage of C, R, U is $\mathcal{O}(k \max(m, n))$. The complexity of finding the value of each entry of CUR is $\mathcal{O}(k^2)$. The complexity of computation of CUR is $\mathcal{O}(k^2 mn)$. In Sect. 14.4 we survey CUR-approximation of tensors given in [13]. In Sect. 14.5 we discuss preliminary results on best \mathbf{r} -approximation of tensors. In Sect. 14.5.1 we show that the minimum problem (14.4) is equivalent to the maximum problem (14.5). In Sect. 14.5.2 we discuss the notion of singular tuples and singular values of a tensor introduced in [32]. In Sect. 14.5.3 we recall the well known solution of maximizing $\|P_{\otimes_{i \in [d]} U_i}(\mathcal{T})\|^2$ with respect to one subspace, while keeping other subspaces fixed. In Sect. 14.6 we discuss AMM for best \mathbf{r} -approximation and its variations. (In [4, 15] AMM is called *alternating least squares*, abbreviated as ALS.) In Sect. 14.6.1 we discuss the AMM on a product space. We mention a *modified alternating maximization method* and a *2-alternating maximization method*, abbreviated as MAMM and 2AMM respectively, introduced in [15]. The MAMM method consists of choosing the one variable which gives the steepest ascend of AMM. 2AMM consists of maximization with respect to a pair of variables, while keeping all other variables fixed. In Sect. 14.6.2 we discuss briefly AMM and MAMM for best \mathbf{r} -approximations for tensors. In Sect. 14.6.3 we give the complexity analysis of AMM for $d = 3, r_1 \approx r_2 \approx r_3$ and $n_1 \approx n_2 \approx n_3$. In Sect. 14.7 we state a working assumption of this paper that AMM converges to a fixed point of the induced map, which satisfies certain smoothness assumptions. Under these assumptions we can apply the Newton method, which can be stated in the standard form in \mathbb{R}^L . Thus, we first do a number of AMM iterations and then switch to the Newton method. In Sect. 14.7.1 we give a simple application of these ideas to state a Newton method for best rank one approximation. This Newton method was suggested in [14]. It is different from the Newton method in [43]. The new contribution of this paper is the Newton method which is discussed in

Sects. 14.8 and 14.9. The advantage of our Newton method is its simplicity, which avoids the notions and tools of Riemannian geometry as for example in [8, 39]. In simulations that we ran, the Newton method in [8] was 20 % faster than our Newton method for best \mathbf{r} -approximation of 3-mode tensors. However, the number of iterations of our Newton method was 40 % less than in [8]. In the last section we give numerical results of our methods for best \mathbf{r} -approximation of tensors. In Sect. 14.11 we give numerical simulations of our different methods applied to a real computer tomography (CT) data set (the so-called MELANIX data set of OsiriX). The summary of these results are given in Sect. 14.12.

14.2 Singular Value Decomposition

Let $A \in \mathbb{R}^{m \times n} \setminus \{0\}$. We now recall well known facts on the SVD of A [19]. See [40] for the early history of the SVD. Assume that $r = \text{rank } A$. Then there exist r -orthonormal sets of vectors $\mathbf{u}_1, \dots, \mathbf{u}_r \in \mathbb{R}^m, \mathbf{v}_1, \dots, \mathbf{v}_r \in \mathbb{R}^n$ such that we have:

$$\begin{aligned}
 A\mathbf{v}_i &= \sigma_i(A)\mathbf{u}_i, \quad \mathbf{u}_i^\top A = \sigma_i(A)\mathbf{v}_i^\top, \quad i \in [r], \quad \sigma_1(A) \geq \dots \geq \sigma_r(A) > 0, \\
 A_k &= \sum_{i \in [k]} \sigma_i(A)\mathbf{u}_i\mathbf{v}_i^\top, \quad k \in [r], \quad A = A_r.
 \end{aligned}
 \tag{14.6}$$

The quantities $\mathbf{u}_i, \mathbf{v}_i$ and $\sigma_i(A)$ are called the left, right i -th singular vectors and i -th singular value of A respectively, for $i \in [r]$. Note that \mathbf{u}_k and \mathbf{v}_k are uniquely defined up to ± 1 if and only if $\sigma_{k-1}(A) > \sigma_k(A) > \sigma_{k+1}(A)$. Furthermore for $k \in [r - 1]$ the matrix A_k is uniquely defined if and only if $\sigma_k(A) > \sigma_{k+1}(A)$. Denote by $\mathcal{R}(m, n, k) \subset \mathbb{R}^{m \times n}$ the variety of all matrices of rank at most k . Then A_k is a best rank- k approximation of A :

$$\min_{B \in \mathcal{R}(m, n, k)} \|A - B\| = \|A - A_k\|.$$

Let $\mathbf{U} \in \text{Gr}(p, \mathbb{R}^m), \mathbf{V} \in \text{Gr}(q, \mathbb{R}^n)$. We identify $\mathbf{U} \otimes \mathbf{V}$ with

$$\mathbf{UV}^\top := \text{span}\{\mathbf{u}\mathbf{v}^\top, \quad \mathbf{u} \in \mathbf{U}, \mathbf{v} \in \mathbf{V}\} \subset \mathbb{R}^{m \times n}.
 \tag{14.7}$$

Then $P_{\mathbf{U} \otimes \mathbf{V}}(A)$ is identified with the projection of A on \mathbf{UV}^\top with respect to the standard inner product on $\mathbb{R}^{m \times n}$ given by $\langle X, Y \rangle = \text{tr } XY^\top$. Observe that

$$\text{Range } A = \mathbf{U}_r^*, \quad \mathbb{R}^m = \mathbf{U}_r^* \oplus (\mathbf{U}_r^*)^\perp, \quad \text{Range } A^\top = \mathbf{V}_r^*, \quad \mathbb{R}^n = \mathbf{V}_r^* \oplus (\mathbf{V}_r^*)^\perp.$$

Hence

$$P_{\mathbf{U} \otimes \mathbf{V}}(A) = P_{(\mathbf{U} \cup \mathbf{U}_r^*) \otimes (\mathbf{V} \cup \mathbf{V}_r^*)}(A) \Rightarrow \text{rank } P_{\mathbf{U} \otimes \mathbf{V}}(A) \leq \min(\dim \mathbf{U}, \dim \mathbf{V}, r).$$

Thus

$$\begin{aligned} \max_{\mathbf{U} \in \text{Gr}(p, \mathbb{R}^m), \mathbf{V} \in \text{Gr}(q, \mathbb{R}^n)} \|P_{\mathbf{U} \otimes \mathbf{V}}(A)\|^2 &= \|P_{\mathbf{U}_l^* \otimes \mathbf{V}_l^*}(A)\|^2 = \sum_{j \in [l]} \sigma_j(A)^2, \\ \min_{\mathbf{U} \in \text{Gr}(p, \mathbb{R}^m), \mathbf{V} \in \text{Gr}(q, \mathbb{R}^n)} \|A - P_{\mathbf{U} \otimes \mathbf{V}}(A)\|^2 &= \|A - P_{\mathbf{U}_l^* \otimes \mathbf{V}_l^*}(A)\|^2 = \sum_{j \in [r] \setminus [l]} \sigma_j(A)^2, \\ l &= \min(p, q, r). \end{aligned} \tag{14.8}$$

To compute \mathbf{U}_l^* , \mathbf{V}_l^* and $\sigma_1(A), \dots, \sigma_l(A)$ of a large scale matrix A one can use Arnoldi or Lanczos methods [19, 31], which are implemented in the partial singular value decomposition. This requires a substantial number of matrix-vector multiplications with the matrix A and thus a complexity of at least $\mathcal{O}(lmn)$.

14.3 Sampling in Low Rank Approximation of Matrices

Let $A = [a_{i,j}]_{i,j=1}^{m,n} \in \mathbb{R}^{m \times n}$ be given. Assume that $\mathbf{b}_1, \dots, \mathbf{b}_m \in \mathbb{R}^n$, $\mathbf{c}_1, \dots, \mathbf{c}_n \in \mathbb{R}^m$ are the columns of A^\top and A respectively. ($\mathbf{b}_1^\top, \dots, \mathbf{b}_m^\top$ are the rows of A .) Most of the known fast rank k -approximation are using sampling of rows or columns of A , or both.

14.3.1 Low Rank Approximations Using Sampling of Rows

Suppose that we sample a set

$$I := \{1 \leq i_1 < \dots < i_s \leq m\} \subset [m], \quad |I| = s, \tag{14.9}$$

of rows $\mathbf{b}_{i_1}^\top, \dots, \mathbf{b}_{i_s}^\top$, where $s \geq k$. Let $\mathbf{W}(I) := \text{span}(\mathbf{b}_{i_1}, \dots, \mathbf{b}_{i_s})$. Then with high probability the projection of the first i -th right singular vectors \mathbf{v}_i on $\mathbf{W}(I)$ is very close to \mathbf{v}_i for $i \in [k]$, provided that $s \gg k$. In particular, [5, Theorem 2] claims:

Theorem 1 (Deshpande-Vempala) *Any $A \in \mathbb{R}^{m \times n}$ contains a subset I of $s = \frac{4k}{\epsilon} + 2k \log(k+1)$ rows such that there is a matrix \tilde{A}_k of rank at most k whose rows lie in $\mathbf{W}(I)$ and*

$$\|A - \tilde{A}_k\|^2 \leq (1 + \epsilon) \|A - A_k\|^2.$$

To find a rank- k approximation of A , one projects each row of A on $\mathbf{W}(I)$ to obtain the matrix $P_{\mathbb{R}^m \otimes \mathbf{W}(I)}(A)$. Note that we can view $P_{\mathbb{R}^m \otimes \mathbf{W}(I)}(A)$ as an $m \times$

s' matrix, where $s' = \dim \mathbf{W}(I) \leq s$. Then find a best rank k -approximation to $P_{\mathbb{R}^m \otimes \mathbf{W}(I)}(A)$, denoted as $P_{\mathbb{R}^m \otimes \mathbf{W}(I)}(A)_k$. Theorem 1 and the results of [18] yield that

$$\|A - P_{\mathbb{R}^m \otimes \mathbf{W}(I)}(A)_k\|^2 \leq (1 + \epsilon)\|A - A_k\|^2 + \eta\|A - P_{\mathbb{R}^m \otimes \mathbf{W}(I)}(A)\|^2.$$

Here η is proportional to $\frac{k}{s}$, and can be decreased with more rounds of sampling. Note that the complexity of computing $P_{\mathbb{R}^m \otimes \mathbf{W}(I)}(A)_k$ is $\mathcal{O}(ks'm)$. The key weakness of this method is that to compute $P_{\mathbb{R}^m \otimes \mathbf{W}(I)}(A)$ one needs $\mathcal{O}(s'mn)$ operations. Indeed, after having computed an orthonormal basis of $\mathbf{W}(I)$, to compute the projection of each row of A on $\mathbf{W}(I)$ one needs $s'n$ multiplications.

An approach for finding low rank approximations of A using random sampling of rows or columns is given in Friedland-Kave-Niknejad-Zare [11]. Start with a random choice of I rows of A , where $|I| \geq k$ and $\dim \mathbf{W}(I) \geq k$. Find $P_{\mathbb{R}^m \otimes \mathbf{W}(I)}(A)$ and $B_1 := P_{\mathbb{R}^m \otimes \mathbf{W}(I)}(A)_k$ as above. Let $\mathbf{U}_1 \in \text{Gr}(k, \mathbb{R}^m)$ be the subspace spanned by the first k left singular vectors of B_1 . Find $B_2 = P_{\mathbf{U}_1 \otimes \mathbb{R}^n}(A)$. Let $\mathbf{V}_2 \in \text{Gr}(k, \mathbb{R}^n)$ correspond to the first k right singular vectors of B_2 . Continuing in this manner we obtain a sequence of rank k -approximations B_1, B_2, \dots . It is shown in [11] that $\|A - B_1\| \geq \|A - B_2\| \geq \dots$ and $\|B_1\| \leq \|B_2\| \leq \dots$. One stops the iterations when the relative improvement of the approximation falls below the specified threshold. Assume that $\sigma_k(A) > \sigma_{k+1}(A)$. Since best rank- k approximation is a unique local minimum for the function $\|A - B\|$, $B \in \mathcal{R}(m, n, k)$ [19], it follows that in general the sequence $B_j, j \in \mathbb{N}$ converges to A_k . It is straightforward to show that this algorithm is the AMM for low rank approximations given in Sect. 14.6.2. Again, the complexity of this method is $\mathcal{O}(kmn)$.

Other suggested methods as [1, 7, 37, 38] seem to have the same complexity $\mathcal{O}(kmn)$, since they project each row of A on some k -dimensional subspace of \mathbb{R}^n .

14.3.2 CUR-Approximations

Let

$$J := \{1 \leq j_1 < \dots < j_t \leq n\} \subset [n], \quad |J| = t, \tag{14.10}$$

and $I \subset [m]$ as in (14.9) be given. Denote by $A[I, J] := [a_{i_p, j_q}]_{p=q=1}^{s,t} \in \mathbb{R}^{s \times t}$. CUR-approximation is based on sampling simultaneously the set of I rows and J columns of A and the approximation matrix to $A(I, J, U)$ given by

$$A(I, J, U) := CUR, \quad C := A[[m], J], \quad R := A[I, [n]], \quad U \in \mathbb{R}^{t \times s}. \tag{14.11}$$

Once the sets I and J are chosen the approximation $A(I, J, U)$ depends on the choice of U . Clearly the row and the column spaces of $A(I, J, U)$ are contained

in the row and column spaces of $A[I, [n]]$ and $A[[m], J]$ respectively. Note that to store the approximation $A(I, J, U)$ we need to store the matrices C , R and U . The number of these entries is $tm + sn + st$. So if n, m are of order 10^5 and s, t are of order 10^2 the storages of C, R, U can be done in Random Access Memory (RAM), while the entries of A are stored in external memory. To compute an entry of $A(I, J, U)$, which is an approximation of the corresponding entry of A , we need st flops.

Let \mathbf{U} and \mathbf{V} be subspaces spanned by the columns of $A[[m], J]$ and $A[I, [n]]^\top$ respectively. Then $A(I, J, U) \in \mathbf{UV}^\top$, see (14.7).

Clearly, a best CUR approximation is chosen by the least squares principle:

$$\begin{aligned} A(I, J, U^*) &= A([m], J)U^*A(I, [n]), \\ U^* &= \arg \min\{\|A - A([m], J)UA(I, [n])\|, U \in \mathbb{R}^{|J| \times |I|}\}. \end{aligned} \quad (14.12)$$

The results in [17] show that the least squares solution of (14.12) is given by:

$$U^* = A([m], J)^\dagger A A(I, [n])^\dagger. \quad (14.13)$$

Here F^\dagger denotes the Moore-Penrose pseudoinverse of a matrix F . Note that U^* is unique if and only if

$$\text{rank } A([m], J) = |J|, \quad \text{rank } A(I, [n]) = |I|. \quad (14.14)$$

The complexity of computation of $A([m], J)^\dagger$ and $A(I, [n])^\dagger$ are $\mathcal{O}(t^2m)$ and $\mathcal{O}(s^2n)$ respectively. Because of the multiplication formula for U^* , the complexity of computation of U^* is $\mathcal{O}(stmn)$.

One can significantly improve the computation of U , if one tries to best fit the entries of the submatrix $A[I', J']$ for given subsets $I' \subset [m], J' \subset [n]$. That is, let

$$\begin{aligned} U^*(I', J') &:= \arg \min\{\|A[I', J'] - A(I', J)UA(I, J')\|, U \in \mathbb{R}^{|J| \times |I|}\} = \\ &A[I', J]^\dagger A[I', J'] A^\dagger[I, J']. \end{aligned} \quad (14.15)$$

(The last equality follows from (14.13).) The complexity of computation of $U^*(I', J')$ is $\mathcal{O}(st|I'||J'|)$.

Suppose finally, that $I' = I$ and $J' = J$. Then (14.15) and the properties of the Moore-Penrose inverse yield that

$$U^*(I, J) = A[I, J]^\dagger, \quad A(I, J, U^*(I, J)) = B(I, J) := A([m], J)A[I, J]^\dagger A[I, [n]]. \quad (14.16)$$

In particular $B(I, J)[I, J] = A[I, J]$. Hence

$$\begin{aligned} A[[m], J] &= B(I, J)[[m], J], & A[I, [n]] &= B(I, J)[I, [n]], \\ B(I, J) &= A[[m], J]A[I, J]^{-1}A[I, [n]] & \text{if } |I| = |J| = k \text{ and } \det A[I, J] \neq 0. \end{aligned} \quad (14.17)$$

The original *CUR* approximation of rank k has the form $B(I, J)$ given by (14.17) [21].

Assume that $\text{rank } A \geq k$. We want to choose an approximation $B(I, J)$ of the form (14.17) which gives a good approximation to A . It is possible to give an upper estimate for the maximum of the absolute values of the entries of $A - B(I, J)$ in terms of $\sigma_{k+1}(A)$, provided that $\det A[I, J]$ is relatively close to

$$\mu_k := \max_{I \subset [m], J \subset [n], |I|=|J|=k} |\det A[I, J]| > 0. \quad (14.18)$$

Let

$$\|F\|_{\infty, e} := \max_{i \in [m], j \in [n]} |f_{i,j}|, \quad F = [f_{i,j}] \in \mathbb{R}^{m \times n}. \quad (14.19)$$

The results of [21, 22] yield:

$$\|A - B(I, J)\|_{\infty, e} \leq \frac{(k+1)\mu_k}{\det A[I, J]} \sigma_{p+1}(A). \quad (14.20)$$

(See also [10, Chapter 4, §13].)

To find μ_k is probably an NP-hard problem in general [9]. A standard way to find μ_k is either a random search or greedy search [9, 20]. In the special case when A is a symmetric positive definite matrix one can give the exact conditions when the greedy search gives a relatively good result [9].

In the paper by Friedland-Mehrmann-Miedlar-Nkengla [13] a good approximation $B(I, J)$ of the form (14.17) is obtained by a random search on the maximum value of the product of the significant singular values of $A[I, J]$. The approximations found in this way are experimentally better than the approximations found by searching for μ_k .

14.4 Fast Approximation of Tensors

The fast approximation of tensors can be based on several decompositions of tensors such as: Tucker decomposition [42]; matricizations of tensors, as unfolding and applying SVD one time or several time recursively, (see below); higher order singular value decomposition (HOSVD) [3], Tensor-Train decompositions [34, 35];

hierarchical Tucker decomposition [23, 25]. A very recent survey [24] gives an overview on this dynamic field. In this paper we will discuss only the CUR-approximation.

14.4.1 CUR-Approximations of Tensors

Let $\mathcal{T} \in \mathbb{R}^{n_1 \times \dots \times n_d}$. In this subsection we denote the entries of \mathcal{T} as $\mathcal{T}(i_1, \dots, i_d)$ for $i_j \in [n_j]$ and $j \in [d]$. CUR-approximation of tensors is based on matricizations of tensors. The unfolding of \mathcal{T} in the mode $l \in [d]$ consists of rearranging the entries of \mathcal{T} as a matrix $T_l(\mathcal{T}) \in \mathbb{R}^{n_l \times N_l}$, where $N_l = \prod_{i \in [d], i \neq l} n_i$. More general, let $K \cup L = [d]$ be a partition of $[d]$ into two disjoint nonempty sets. Denote $N(K) = \prod_{i \in K} n_i$, $N(L) = \prod_{j \in L} n_j$. Then unfolding \mathcal{T} into the two modes K and L consists of rearranging the entries of \mathcal{T} as a matrix $T(K, L, \mathcal{T}) \in \mathbb{R}^{N(K) \times N(L)}$.

We now describe briefly the CUR-approximation of 3 and 4-tensors as described by Friedland-Mehrmann-Miedlar-Nkengla [13]. (See [33] for another approach to CUR-approximations for tensors.) We start with the case $d = 3$. Let I_i be a nonempty subset of $[n_i]$ for $i \in [3]$. Assume that the following conditions hold:

$$|I_1| = k^2, \quad |I_2| = |I_3| = k, \quad J := I_2 \times I_3 \subset [n_2] \times [n_3].$$

We identify $[n_2] \times [n_3]$ with $[n_2 n_3]$ using a lexicographical order. We now take the CUR-approximation of $T_1(\mathcal{T})$ as given in (14.17):

$$B(I_1, J) = T_1(\mathcal{T})[[n_1], J]T_1(\mathcal{T})[I_1, J]^{-1}T_1(\mathcal{T})[I_1, [n_2 n_3]].$$

We view $T_1(\mathcal{T})[[n_1], J]$ as an $n_1 \times k^2$ matrix. For each $\alpha_1 \in I_1$ we view $T_1(\mathcal{T})[\{\alpha_1\}, [n_2 n_3]]$ as an $n_2 \times n_3$ matrix $Q(\alpha_1) := [\mathcal{T}(\alpha_1, i_2, i_3)]_{i_2 \in [n_2], i_3 \in [n_3]}$. Let $R(\alpha_1)$ be the CUR-approximation of $Q(\alpha_1)$ based on the sets I_2, I_3 :

$$R(\alpha_1) := Q(\alpha_1)[[n_2], I_3]Q(\alpha_1)[I_2, I_3]^{-1}Q(\alpha_1)[I_2, [n_3]].$$

Let $F := T_1(\mathcal{T})[I_1, J]^{-1} \in \mathbb{R}^{k^2 \times k^2}$. We view the entries of this matrix indexed by the row $(\alpha_2, \alpha_3) \in I_2 \times I_3$ and column $\alpha_1 \in I_1$. We write these entries as $\mathcal{F}(\alpha_1, \alpha_2, \alpha_3)$, $\alpha_j \in I_j$, $j \in [3]$, which represent a tensor $\mathcal{F} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$. The entries of $Q(\alpha_1)[I_2, I_3]^{-1}$ are indexed by the row $\alpha_3 \in I_3$ and column $\alpha_2 \in I_2$. We write these entries as $\mathcal{G}(\alpha_1, \alpha_2, \alpha_3)$, $\alpha_2 \in I_2, \alpha_3 \in I_3$, which represent a tensor $\mathcal{G} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$. Then the approximation tensor $\mathcal{B} = [\mathcal{B}(j_1, j_2, j_3)] \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is given by:

$$\begin{aligned} \mathcal{B}(i_1, i_2, i_3) &= \sum_{\alpha_1 \in I_1, \alpha_j, \beta_j \in I_j, j=2,3} \\ &\mathcal{T}(i_1, \alpha_2, \alpha_3) \mathcal{F}(\alpha_1, \alpha_2, \alpha_3) \mathcal{T}(\alpha_1, j_2, \beta_3) \mathcal{G}(\alpha_1, \beta_2, \beta_3) \mathcal{T}(\alpha_1, \beta_2, j_3). \end{aligned}$$

We now discuss a CUR-approximation for 4-tensors, i.e. $d = 4$. Let $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times n_4}$ and $K = \{1, 2\}, L = \{3, 4\}$. The rows and columns of $X := T(K, L, \mathcal{T}) \in \mathbb{R}^{(n_1 n_2) \times (n_3 n_4)}$ are indexed by pairs (i_1, i_2) and (i_3, i_4) respectively. Let

$$I_j \subset [n_j], |I_j| = k, j \in [4], \quad J_1 := I_1 \times I_2, \quad J_2 := I_3 \times I_4.$$

First consider the CUR-approximation $X[[n_1 n_2], J_2]X[J_1, J_2]^{-1}X[J_1, [n_3 n_4]]$ viewed as tensor $\mathcal{C} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times n_4}$. Denote by $\mathcal{H}(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ the $((\alpha_3, \alpha_4), (\alpha_1, \alpha_2))$ entry of the matrix $X[J_1, J_2]^{-1}$. So $\mathcal{H} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times I_4}$. Then

$$\mathcal{C}(i_1, i_2, i_3, i_4) = \sum_{\alpha_j \in I_j, j \in [4]} \mathcal{T}(i_1, i_2, \alpha_3, \alpha_4) \mathcal{H}(\alpha_1, \alpha_2, \alpha_3, \alpha_4) \mathcal{T}(\alpha_1, \alpha_2, i_3, i_4).$$

For $\alpha_j \in I_j, j \in [4]$ view vectors $X[[n_1 n_2], (\alpha_3, \alpha_4)]$ and $X[(\alpha_1, \alpha_2), [n_3 n_4]]$ as matrices $Y(\alpha_3, \alpha_4) \in \mathbb{R}^{n_1 \times n_2}$ and $Z(\alpha_1, \alpha_2) \in \mathbb{R}^{n_3 \times n_4}$ respectively. Next we find the CUR-approximations to these two matrices using the subsets (I_1, I_2) and (I_3, I_4) respectively:

$$Y(\alpha_3, \alpha_4)[[n_1], I_2]Y(\alpha_3, \alpha_4)[I_1, I_2]^{-1}Y(\alpha_3, \alpha_4)[I_1, [n_2]],$$

$$Z(\alpha_1, \alpha_2)[[n_3], I_4]Z(\alpha_1, \alpha_2)[I_3, I_4]^{-1}Z(\alpha_1, \alpha_2)[I_3, [n_4]].$$

We denote the entries of $Y(\alpha_3, \alpha_4)[I_1, I_2]^{-1}$ and $Z(\alpha_1, \alpha_2)[I_3, I_4]^{-1}$ by

$$\mathcal{F}(\alpha_1, \alpha_2, \alpha_3, \alpha_4), \quad \alpha_1 \in I_1, \alpha_2 \in I_2,$$

$$\text{and } \mathcal{G}(\alpha_1, \alpha_2, \alpha_3, \alpha_4), \quad \alpha_3 \in I_3, \alpha_4 \in I_4,$$

respectively. Then the CUR-approximation tensor \mathcal{B} of \mathcal{T} is given by:

$$\mathcal{B}(i_1, i_2, i_3, i_4) = \sum_{\alpha_j, \beta_j \in I_j, j \in [4]} \mathcal{T}(i_1, \beta_2, \alpha_3, \alpha_4) \mathcal{F}(\beta_1, \beta_2, \alpha_3, \alpha_4) \mathcal{T}(\beta_1, i_2, \alpha_3, \alpha_4)$$

$$\mathcal{H}(\alpha_1, \alpha_2, \alpha_3, \alpha_4) \mathcal{T}(\alpha_1, \alpha_2, i_3, \beta_4) \mathcal{G}(\alpha_1, \alpha_2, \beta_3, \beta_4) \mathcal{T}(\alpha_1, \alpha_2, \beta_3, i_4).$$

We now discuss briefly the complexity of the storage and computing an entry of the CUR-approximation \mathcal{B} . Assume first that $d = 3$. Then we need to store k^2 columns of the matrices $T_1(\mathcal{T})$, k^3 columns of $T_2(\mathcal{T})$ and $T_3(\mathcal{T})$, and k^4 entries of the tensors \mathcal{F} and \mathcal{G} . The total storage space is $k^2 n_1 + k^3(n_2 + n_3) + 2k^4$. To compute each entry of \mathcal{B} we need to perform $4k^6$ multiplications and k^6 additions.

Assume now that $d = 4$. Then we need to store k^3 columns of $T_l(\mathcal{T}), l \in [4]$ and k^4 entries of $\mathcal{F}, \mathcal{G}, \mathcal{H}$. Total storage needed is $k^3(n_1 + n_2 + n_3 + n_4 + 3k)$. To compute each entry of \mathcal{B} we need to perform $6k^8$ multiplications and k^8 additions.

14.5 Preliminary Results on Best r -Approximation

14.5.1 The Maximization Problem

We first show that the best approximation problem (14.4) is equivalent to the maximum problem (14.5), see [4] and [26, §10.3]. The Pythagoras theorem yields that

$$\begin{aligned} \|\mathcal{T}\|^2 &= \|P_{\otimes_{i=1}^d \mathbf{U}_i}(\mathcal{T})\|^2 + \|P_{(\otimes_{i=1}^d \mathbf{U}_i)^\perp}(\mathcal{T})\|^2, \\ \|\mathcal{T} - P_{\otimes_{i=1}^d \mathbf{U}_i}(\mathcal{T})\|^2 &= \|P_{(\otimes_{i=1}^d \mathbf{U}_i)^\perp}(\mathcal{T})\|^2. \end{aligned}$$

(Here $(\otimes_{i=1}^d \mathbf{U}_i)^\perp$ is the orthogonal complement of $\otimes_{i=1}^d \mathbf{U}_i$ in $\otimes_{i=1}^d \mathbb{R}^{n_i}$.) Hence

$$\min_{\mathbf{U}_i \in \text{Gr}(r_i, \mathbb{R}^{n_i}), i \in [d]} \|\mathcal{T} - P_{\otimes_{i=1}^d \mathbf{U}_i}(\mathcal{T})\|^2 = \|\mathcal{T}\|^2 - \max_{\mathbf{U}_i \in \text{Gr}(r_i, \mathbb{R}^{n_i}), i \in [d]} \|P_{\otimes_{i=1}^d \mathbf{U}_i}(\mathcal{T})\|^2. \quad (14.21)$$

This shows the equivalence of (14.4) and (14.5).

14.5.2 Singular Values and Singular Tuples of Tensors

Let $S(n) = \{\mathbf{x} \in \mathbb{R}^n, \|\mathbf{x}\| = 1\}$. Note that one dimensional subspace $\mathbf{U} \in \text{Gr}(1, \mathbb{R}^n)$ is $\text{span}(\mathbf{u})$, where $\mathbf{u} \in S(n)$. Let $S(\mathbf{n}) := S(n_1) \times \cdots \times S(n_d)$. Then best rank one approximation problem for $\mathcal{T} \in \mathbb{R}^{\mathbf{n}}$ is equivalent to finding

$$\|\mathcal{T}\|_\sigma := \max_{(\mathbf{x}_1, \dots, \mathbf{x}_d) \in S(\mathbf{n})} \mathcal{T} \times (\otimes_{i \in [d]} \mathbf{x}_i). \quad (14.22)$$

Let $f_{\mathcal{T}} : \mathbb{R}^{\mathbf{n}} \rightarrow \mathbb{R}$ is given by $f_{\mathcal{T}}(\mathcal{X}) = \langle \mathcal{X}, \mathcal{T} \rangle$. Denote by $S'(\mathbf{n}) \subset \mathbb{R}^{\mathbf{n}}$ all rank one tensors of the form $\otimes_{i \in [d]} \mathbf{x}_i$, where $(\mathbf{x}_1, \dots, \mathbf{x}_d) \in S(\mathbf{n})$. Let $f_{\mathcal{T}}(\mathbf{x}_1, \dots, \mathbf{x}_d) := f_{\mathcal{T}}(\otimes_{i \in [d]} \mathbf{x}_i)$. Then the critical points of $f_{\mathcal{T}}|_{S'(\mathbf{n})}$ are given by the Lagrange multipliers formulas [32]:

$$\mathcal{T} \times (\otimes_{j \in [d] \setminus \{i\}} \mathbf{u}_j) = \lambda \mathbf{u}_i, \quad i \in [d], \quad (\mathbf{u}_1, \dots, \mathbf{u}_d) \in S(\mathbf{n}). \quad (14.23)$$

One calls λ and $(\mathbf{u}_1, \dots, \mathbf{u}_d)$ a singular value and singular tuple of \mathcal{T} . For $d = 2$ these are the singular values and singular vectors of \mathcal{T} . The number of complex singular values of a generic \mathcal{T} is given in [16]. This number increases exponentially with d . For example for $n_1 = \cdots = n_d = 2$ the number of distinct singular values is $d!$. (The number of real singular values as given by (14.23) is bounded by the numbers given in [16].)

Consider first the maximization problem of $f_{\mathcal{T}}(\mathbf{x}_1, \dots, \mathbf{x}_d)$ over $S(\mathbf{n})$ where we vary $\mathbf{x}_i \in S(n_i)$ and keep the other variables fixed. This problem is equivalent to

the maximization of the linear form $\mathbf{x}_i^\top (\mathcal{T} \times (\otimes_{j \in [d] \setminus \{i\}} \mathbf{x}_j))$. Note that if $\mathcal{T} \times (\otimes_{j \in [d] \setminus \{i\}} \mathbf{x}_j) \neq \mathbf{0}$ then this maximum is achieved for $\mathbf{x}_i = \frac{1}{\|\mathcal{T} \times (\otimes_{j \in [d] \setminus \{i\}} \mathbf{x}_j)\|} \mathcal{T} \times (\otimes_{j \in [d] \setminus \{i\}} \mathbf{x}_j)$.

Consider second the maximization problem of $f_{\mathcal{T}}(\mathbf{x}_1, \dots, \mathbf{x}_d)$ over $S(\mathbf{n})$ where we vary $(\mathbf{x}_i, \mathbf{x}_j) \in S(n_i) \times S(n_j)$, $1 \leq i < j \leq d$ and keep the other variables fixed. This problem is equivalent to finding the first singular value and the corresponding right and left singular vectors of the matrix $\mathcal{T} \times (\otimes_{k \in [d] \setminus \{i, j\}} \mathbf{x}_k)$. This can be done by using use Arnoldi or Lanczos methods [19, 31]. The complexity of this method is $\mathcal{O}(n_i n_j)$, given the matrix $\mathcal{T} \times (\otimes_{k \in [d] \setminus \{i, j\}} \mathbf{x}_k)$.

14.5.3 A Basic Maximization Problem for Best \mathbf{r} -Approximation

Denote by $S_n \subset \mathbb{R}^{n \times n}$ the space of real symmetric matrices. For $A \in S_n$ denote by $\lambda_1(A) \geq \dots \geq \lambda_n(A)$ the eigenvalues of A arranged in a decreasing order and repeated according to their multiplicities. Let $O(n, k) \subset \mathbb{R}^{n \times k}$ be the set of all $n \times k$ matrices X with k orthonormal columns, i.e. $X^\top X = I_k$, where I_k is $k \times k$ identity matrix. We view $X \in \mathbb{R}^{n \times k}$ as composed of k -columns $[\mathbf{x}_1 \dots \mathbf{x}_k]$. The column space of $X \in O(n, k)$ corresponds to a k -dimensional subspace $\mathbf{U} \subset \mathbb{R}^n$. Note that $\mathbf{U} \in \text{Gr}(k, \mathbb{R}^n)$ is spanned by the orthonormal columns of a matrix $Y \in O(n, k)$ if and only if $Y = XO$, for some $O \in O(k, k)$.

For $A \in S_n$ one has the Ky-Fan maximal characterization [28, Cor. 4.3.18]

$$\max_{\{\mathbf{x}_1 \dots \mathbf{x}_k\} \in O(n, k)} \sum_{i=1}^k \mathbf{x}_i^\top A \mathbf{x}_i = \sum_{i=1}^k \lambda_i(A). \tag{14.24}$$

Equality holds if and only if the column space of $X = [\mathbf{x}_1 \dots \mathbf{x}_k]$ is a subspace spanned by k eigenvectors corresponding to k -largest eigenvalues of A .

We now reformulate the maximum problem (14.5) in terms of orthonormal bases of \mathbf{U}_i , $i \in [d]$. Let $\mathbf{u}_{1,i}, \dots, \mathbf{u}_{n_i,i}$ be an orthonormal basis of \mathbf{U}_i for $i \in [d]$. Then $\otimes_{i=1}^d \mathbf{u}_{j_i,i}$, $j_i \in [n_i]$, $i \in [d]$ is an orthonormal basis of $\otimes_{i=1}^d \mathbf{U}_i$. Hence

$$\|P_{\otimes_{i=1}^d \mathbf{U}_i}(\mathcal{T})\|^2 = \sum_{j_i \in [n_i], i \in [d]} \langle \mathcal{T}, \otimes_{i=1}^d \mathbf{u}_{j_i,i} \rangle^2.$$

Hence (14.5) is equivalent to

$$\begin{aligned} & \max_{\{\mathbf{u}_{1,i} \dots \mathbf{u}_{r_i,i}\} \in O(n_i, r_i), i \in [d]} \sum_{j_i \in [n_i], i \in [d]} \langle \mathcal{T}, \otimes_{i=1}^d \mathbf{u}_{j_i,i} \rangle^2 = \tag{14.25} \\ & \max_{\mathbf{U}_i \in \text{Gr}(r_i, \mathbb{R}^{n_i}), i \in [d]} \|P_{\otimes_{i=1}^d \mathbf{U}_i}(\mathcal{T})\|^2. \end{aligned}$$

A simpler problem is to find

$$\begin{aligned} \max_{\{\mathbf{u}_{1,i}, \dots, \mathbf{u}_{r_i,i}\} \in \mathcal{O}(n_i, r_i)} \sum_{j_i \in [n_i], i \in [d]} \langle \mathcal{T}, \otimes_{i=1}^d \mathbf{u}_{j_i,i} \rangle^2 = \\ \max_{\mathbf{U}_i \in \text{Gr}(r_i, \mathbb{R}^{n_i})} \|P_{\otimes_{i=1}^d \mathbf{U}_i}(\mathcal{T})\|^2, \end{aligned} \quad (14.26)$$

for a fixed $i \in [d]$. Let

$$\begin{aligned} \underline{\mathbf{U}} &:= (\mathbf{U}_1, \dots, \mathbf{U}_d) \in \text{Gr}(\mathbf{r}, \mathbf{n}), \\ \text{Gr}_i(\mathbf{r}, \mathbf{n}) &:= \text{Gr}(r_1, n_1) \times \dots \times \text{Gr}(r_{i-1}, n_{i-1}) \times \text{Gr}(r_{i+1}, n_{i+1}) \times \dots \times \text{Gr}(r_d, n_d), \\ \underline{\mathbf{U}}_i &:= (\mathbf{U}_1, \dots, \mathbf{U}_{i-1}, \mathbf{U}_{i+1}, \dots, \mathbf{U}_d) \in \text{Gr}_i(\mathbf{r}, \mathbf{n}), \\ A_i(\underline{\mathbf{U}}_i) &:= \sum_{j_l \in [r_l], l \in [d] \setminus \{i\}} (\mathcal{T} \times \otimes_{k \in [d] \setminus \{i\}} \mathbf{u}_{j_k,k}) (\mathcal{T} \times \otimes_{k \in [d] \setminus \{i\}} \mathbf{u}_{j_k,k})^\top. \end{aligned} \quad (14.27)$$

The maximization problem (14.25) reduces to the maximum problem (14.24) with $A = A_i(\underline{\mathbf{U}}_i)$. Note that each $A_i(\underline{\mathbf{U}}_i)$ is a positive semi-definite matrix. Hence $\sigma_j(A_i(\underline{\mathbf{U}}_i)) = \lambda_j(A_i(\underline{\mathbf{U}}_i))$ for $j \in [n_i]$. Thus the complexity to find the first r_i eigenvectors of $A_i(\underline{\mathbf{U}}_i)$ is $\mathcal{O}(r_i n_i^2)$. Denote by $\mathbf{U}_i(\underline{\mathbf{U}}_i) \in \text{Gr}(r_i, \mathbb{R}^{n_i})$ a subspace spanned by the first r_i eigenvectors of $A_i(\underline{\mathbf{U}}_i)$. Note that this subspace is unique if and only if

$$\lambda_{r_i}(A_i(\underline{\mathbf{U}}_i)) > \lambda_{r_i+1}(A_i(\underline{\mathbf{U}}_i)). \quad (14.28)$$

Finally, if $\mathbf{r} = \mathbf{1}_d$ then each $A_i(\underline{\mathbf{U}}_i)$ is a rank one matrix. Hence $\mathbf{U}_i(\underline{\mathbf{U}}_i) = \text{span}(\mathcal{T} \times \otimes_{k \in [d] \setminus \{i\}} \mathbf{u}_{1,k})$. For more details see [12].

14.6 Alternating Maximization Methods for Best \mathbf{r} -Approximation

14.6.1 General Definition and Properties

Let Ψ_i be a compact smooth manifold for $i \in [d]$. Define

$$\Psi := \Psi_1 \times \dots \times \Psi_d, \quad \hat{\Psi}_i = (\Psi_1 \times \dots \times \Psi_{i-1} \times \Psi_{i+1} \times \dots \times \Psi_d) \text{ for } i \in [d].$$

We denote by ψ_i , $\psi = (\psi_1, \dots, \psi_d)$ and $\hat{\psi}_i = (\psi_1, \dots, \psi_{i-1}, \psi_{i+1}, \dots, \psi_d)$ the points in Ψ_i , Ψ and $\hat{\Psi}_i$ respectively. Identify ψ with $(\psi_i, \hat{\psi}_i)$ for each $i \in [d]$. Assume that $f : \Psi \rightarrow \mathbb{R}$ is a continuous function with continuous first and second partial derivatives. (In our applications it may happen that f has discontinuities in

first and second partial derivatives.) We want to find the maximum value of f and a corresponding maximum point ψ^* :

$$\max_{\psi \in \Psi} f(\psi) = f(\psi^*). \tag{14.29}$$

Usually, this is a hard problem, where f has many critical points and a number of these critical points are local maximum points. In some cases, as best \mathbf{r} approximation to a given tensor $\mathcal{T} \in \mathbb{R}^n$, we can solve the maximization problem with respect to one variable ψ_i for any fixed $\hat{\psi}_i$:

$$\max_{\psi_i \in \Psi_i} f((\psi_i, \hat{\psi}_i)) = f((\psi_i^*(\hat{\psi}_i), \hat{\psi}_i)), \tag{14.30}$$

for each $i \in [d]$.

Then the *alternating maximization method*, abbreviated as AMM, is as follows. Assume that we start with an initial point $\psi^{(0)} = (\psi_1^{(0)}, \dots, \psi_d^{(0)}) = (\psi_1^{(0)}, \hat{\psi}_1^{(0,1)})$. Then we consider the maximal problem (14.30) for $i = 1$ and $\hat{\psi}_1 := \hat{\psi}_1^{(0,1)}$. This maximum is achieved for $\psi_1^{(1)} := \psi_1^*(\hat{\psi}_1^{(0,1)})$. Assume that the coordinates $\psi_1^{(1)}, \dots, \psi_j^{(1)}$ are already defined for $j \in [d - 1]$. Let $\hat{\psi}_{j+1}^{(0,j+1)} := (\psi_1^{(1)}, \dots, \psi_j^{(1)}, \psi_{j+2}^{(0)}, \dots, \psi_d^{(0)})$. Then we consider the maximum problem (14.30) for $i = j + 1$ and $\hat{\psi}_{j+1} := \hat{\psi}_{j+1}^{(0,j+1)}$. This maximum is achieved for $\psi_{j+1}^{(1)} := \psi_{j+1}^*(\hat{\psi}_{j+1}^{(0,j+1)})$. Executing these d iterations we obtain $\psi^{(1)} := (\psi_1^{(1)}, \dots, \psi_d^{(1)})$. Note that we have a sequence of inequalities:

$$f(\psi^{(0)}) \leq f(\psi_1^{(1)}, \hat{\psi}_1^{(0,1)}) \leq f(\psi_2^{(1)}, \hat{\psi}_2^{(0,2)}) \leq \dots \leq f(\psi_d^{(1)}, \hat{\psi}_d^{(0,d)}) = f(\psi^{(1)}).$$

Replace $\psi^{(0)}$ with $\psi^{(1)}$ and continue these iterations to obtain a sequence $\psi^{(l)} = (\psi_1^{(l)}, \dots, \psi_d^{(l)})$ for $l = 0, \dots, N$. Clearly,

$$f(\psi^{(l-1)}) \leq f(\psi^{(l)}) \text{ for } l \in \mathbb{N} \Rightarrow \lim_{l \rightarrow \infty} f(\psi^{(l)}) = M. \tag{14.31}$$

Usually, the sequence $\psi^{(l)}, l = 0, \dots$, will converge to 1-semi maximum point $\phi = (\phi_1, \dots, \phi_d) \in \Psi$. That is, $f(\phi) = \max_{\psi_i \in \Psi} f((\psi_i, \hat{\phi}_i))$ for $i \in [d]$. Note that if f is differentiable at ϕ then ϕ is a critical point of f . Assume that f is twice differentiable at ϕ . Then ϕ does not have to be a local maximum point [15, Appendix].

The *modified alternating maximization method*, abbreviated as MAMM, is as follows. Assume that we start with an initial point $\psi^{(0)} = (\psi_1^{(0)}, \dots, \psi_d^{(0)})$. Let $\psi^{(0)} = (\psi_i^{(0)}, \hat{\psi}_i^{(0)})$ for $i \in [d]$. Compute $f_{i,0} = \max_{\psi_i \in \Psi_i} f((\psi_i, \hat{\psi}_i^{(0)}))$ for $i \in [d]$. Let $j_1 \in \arg \max_{i \in [d]} f_{i,0}$. Then $\psi^{(1)} = (\psi_{j_1}^*(\hat{\psi}_{j_1}^{(0)}), \hat{\psi}_{j_1}^{(0)})$ and $f_1 = f_{1,j_1} = f(\psi^{(1)})$. Note that it takes d iterations to compute $\psi^{(1)}$. Now replace $\psi^{(0)}$ with $\psi^{(1)}$ and compute $f_{i,1} = \max_{\psi_i \in \Psi_i} f((\psi_i, \hat{\psi}_i^{(1)}))$ for $i \in [d] \setminus \{j_1\}$. Continue as above to find

$\psi^{(l)}$ for $l = 2, \dots, N$. Note that for $l \geq 2$ it takes $d - 1$ iterations to determine $\psi^{(l)}$. Clearly, (14.31) holds. It is shown in [15] that the limit ϕ of each convergent subsequence of the points $\psi^{(j)}$ is 1-semi maximum point of f .

In certain very special cases, as for best rank one approximation, we can solve the maximization problem with respect to any pair of variables ψ_i, ψ_j for $1 \leq i < j \leq d$, where $d \geq 3$ and all other variables are fixed. Let

$$\begin{aligned}\hat{\Psi}_{i,j} &:= \Psi_1 \times \cdots \times \Psi_{i-1} \times \Psi_{i+1} \times \cdots \times \Psi_{j-1} \times \Psi_{j+1} \times \cdots \times \Psi_d, \\ \hat{\psi}_{i,j} &= (\psi_1, \dots, \psi_{i-1}, \psi_{i+1}, \dots, \psi_{j-1}, \psi_{j+1}, \dots, \psi_d) \in \hat{\Psi}_{i,j}, \\ \psi_{i,j} &= (\psi_i, \psi_j) \in \Psi_i \times \Psi_j.\end{aligned}$$

View $\psi = (\psi_1, \dots, \psi_d)$ as $(\psi_{i,j}, \hat{\psi}_{i,j})$ for each pair $1 \leq i < j \leq d$. Then

$$\max_{\psi_{i,j} \in \Psi_i \times \Psi_j} f((\psi_{i,j}, \hat{\psi}_{i,j})) = f((\psi_{i,j}^*, \hat{\psi}_{i,j}), \hat{\psi}_{i,j}). \quad (14.32)$$

A point ψ is called *2-semi maximum point* if the above maximum equals to $f(\psi)$ for each pair $1 \leq i < j \leq d$.

The *2-alternating maximization method*, abbreviated here as 2AMM, is as follows. Assume that we start with an initial point $\psi^{(0)} = (\psi_1^{(0)}, \dots, \psi_d^{(0)})$. Suppose first that $d = 3$. Then we consider the maximization problem (14.32) for $i = 2, j = 3$ and $\hat{\psi}_{2,3} = \psi_1^{(0)}$. Let $(\psi_2^{(0,1)}, \psi_3^{(0,1)}) = \psi_{2,3}^*(\psi_1^{(0)})$. Next let $i = 1, j = 3$ and $\psi_{1,3} = \psi_2^{(0,1)}$. Then $(\psi_1^{(0,2)}, \psi_3^{(0,2)}) = \psi_{1,3}^*(\psi_2^{(0,1)})$. Next let $i = 1, 2$ and $\hat{\psi}_{1,2} = \psi_3^{(0,2)}$. Then $\psi^{(1)} = (\hat{\psi}_{1,2}^*(\psi_3^{(0,2)}), \psi_3^{(0,2)})$. Continue these iterations to obtain $\psi^{(l)}$ for $l = 2, \dots$. Again, (14.31) holds. Usually the sequence $\psi^{(l)}, l \in \mathbb{N}$ will converge to a 2-semi maximum point ϕ . For $d \geq 4$ the 2AMM can be defined appropriately see [15].

A *modified 2-alternating maximization method*, abbreviated here as M2AMM, is as follows. Start with an initial point $\psi^{(0)} = (\psi_1^{(0)}, \dots, \psi_d^{(0)})$ viewed as $(\psi_{i,j}^{(0)}, \hat{\psi}_{i,j}^{(0)})$, for each pair $1 \leq i < j \leq d$. Let $f_{i,j,0} := \max_{\psi_{i,j} \in \Psi_i \times \Psi_j} f((\psi_{i,j}, \hat{\psi}_{i,j}^{(0)}))$. Assume that $(i_1, j_1) \in \arg \max_{1 \leq i < j \leq d} f_{i,j,0}$. Then $\psi^{(1)} = (\psi_{i_1, j_1}^*(\hat{\psi}_{i_1, j_1}^{(0)}), \hat{\psi}_{i_1, j_1}^{(0)})$. Let $f_{i_1, j_1, 1} := f(\psi^{(1)})$. Note that it takes $\binom{d}{2}$ iterations to compute $\psi^{(1)}$. Now replace $\psi^{(0)}$ with $\psi^{(1)}$ and compute $f_{i,j,1} = \max_{\psi_{i,j} \in \Psi_i \times \Psi_j} f((\psi_{i,j}, \hat{\psi}_{i,j}^{(1)}))$ for all pairs $1 \leq i < j \leq d$ except the pair (i_1, j_1) . Continue as above to find $\psi^{(l)}$ for $l = 2, \dots, N$. Note that for $l \geq 2$ it takes $\binom{d}{2} - 1$ iterations to determine $\psi^{(l)}$. Clearly, (14.31) holds. It is shown in [15] that the limit ϕ of each convergent subsequence of the points $\psi^{(j)}$ is 2-semi maximum point of f .

14.6.2 AMM for Best \mathbf{r} -Approximations of Tensors

Let $\mathcal{T} \in \mathbb{R}^{\mathbf{n}}$. For best rank one approximation one searches for the maximum of the function $f_{\mathcal{T}} = \mathcal{T} \times (\otimes_{i \in [d]} \mathbf{x}_i)$ on $S(\mathbf{n})$, as in (14.22). For best \mathbf{r} -approximation one searches for the maximum of the function $f_{\mathcal{T}} = \|\mathcal{P}_{\otimes_{i \in [d]} \mathbf{U}_i}\|^2$ on $\text{Gr}(\mathbf{r}, \mathbf{n})$, as in (14.26). A solution to the basic maximization problem with respect to one subspace \mathbf{U}_i is given in Sect. 14.5.3.

The AMM for best \mathbf{r} -approximation were studied first by de Lathauwer-Moor-Vandewalle [4]. The AMM is called in [4] *alternating least squares*, abbreviated as ALS. A crucial problem is the starting point of AMM. A high order SVD, abbreviated as HOSVD, for \mathcal{T} , see [3], gives a good starting point for AMM. That is, let $T_l(\mathcal{T}) \in \mathbb{R}^{n_l \times N_l}$ be the unfolded matrix of \mathcal{T} in the mode l , as in Sect. 14.4.1. Then \mathbf{U}_l is the subspace spanned by the first l -left singular vectors of $T_l(\mathcal{T})$. The complexity of computing \mathbf{U}_l is $\mathcal{O}(r_l N)$, where $N = \prod_{i \in [d]} n_i$. Hence for large N the complexity of computing partial HOSVD is high. Another approach is to choose the starting subspaces at random, and repeat the AMM for several choices of random starting points.

MAMM for best rank one approximation of tensors was introduced by Friedland-Mehrmann-Pajarola-Suter in [15] by the name *modified alternating least squares*, abbreviated as MALs. 2AMM for best rank one approximation was introduced in [15] by the name *alternating SVD*, abbreviated as ASVD. It follows from the observation that $A := \mathcal{T} \times (\otimes_{l \in [d] \setminus \{i, j\}} \mathbf{x}_l)$ is an $n_i \times n_j$ matrix. Hence the maximum of the bilinear form $\mathbf{x}^T A \mathbf{y}$ on $S((n_i, n_j))$ is $\sigma_1(A)$. See Sect. 14.2. M2AMM was introduced in [15] by the names MASVD.

We now introduce the following variant of 2AMM for best \mathbf{r} -rank approximation, called *2-alternating maximization method variant* and abbreviated as 2AMMV. Consider the maximization problem for a pair of variables as in (14.32). Since for $\mathbf{r} \neq \mathbf{1}_d$ we do not have a closed solution to this problem, we apply the AMM for two variables ψ_i and ψ_j , while keeping $\hat{\psi}_{i,j}$ fixed. We then continue as in 2AMM method.

14.6.3 Complexity Analysis of AMM for Best \mathbf{r} -Approximation

Let $\underline{U} = (\mathbf{U}_1, \dots, \mathbf{U}_d)$. Assume that

$$\begin{aligned} \mathbf{U}_i &= \text{span}(\mathbf{u}_{1,i}, \dots, \mathbf{u}_{r_i, n_i}), & \mathbf{U}_i^\perp &= \text{span}(\mathbf{u}_{r_i+1}, \dots, \mathbf{u}_{n_i, i}), \\ \mathbf{u}_{j,i}^\top \mathbf{u}_{k,i} &= \delta_{j,k}, \quad j, k \in [n_i], & i &\in [d]. \end{aligned} \quad (14.33)$$

For each $i \in [d]$ compute the symmetric positive semi-definite matrix $A_i(\underline{U}_i)$ given by (14.27). For simplicity of exposition we give the complexity analysis for $d = 3$. To compute $A_1(\underline{U}_1)$ we need first to compute the vectors $\mathcal{T} \times (\mathbf{u}_{j_2, 2} \otimes \mathbf{u}_{j_3, 3})$ for $j_2 \in [r_2]$ and $j_3 \in [r_3]$. Each computation of such a vector has complexity $\mathcal{O}(N)$, where

$N = n_1 n_2 n_3$. The number of such vectors is $r_2 r_3$. To form the matrix $A_i(\underline{U}_i)$ we need $\mathcal{O}(r_2 r_3 n_1^2)$ flops. To find the first r_1 eigenvectors of $A_1(\underline{U}_1)$ we need $\mathcal{O}(r_1 n_1^2)$ flops. Assuming that $n_1, n_2, n_3 \approx n$ and $r_1, r_2, r_3 \approx r$ we deduce that we need $\mathcal{O}(r^2 n^3)$ flops to find the first r_1 orthonormal eigenvectors of $A_1(\underline{U}_1)$ which span $\mathbf{U}_1(\underline{U}_1)$. Hence the complexity of finding orthonormal bases of $\mathbf{U}_1(\underline{U}_1)$ is $\mathcal{O}(r^2 n^3)$, which is the complexity of computing $A_1(\underline{U}_1)$. Hence the complexity of each step of AMM for best \mathbf{r} -approximation, i.e. computing $\psi^{(l)}$, is $\mathcal{O}(r^2 n^3)$.

It is possible to reduce the complexity of AMM for best \mathbf{r} -approximation to $\mathcal{O}(r n^3)$ if we compute and store the matrices $\mathcal{T} \times \mathbf{u}_{j_1,1}, \mathcal{T} \times \mathbf{u}_{j_2,2}, \mathcal{T} \times \mathbf{u}_{j_3,3}$. See Sect. 14.10.

We now analyze the complexity of AMM for rank one approximation. In this case we need only to compute the vector of the form $\mathbf{v}_i := \mathcal{T} \times (\otimes_{j \in [d] \setminus \{i\}} \mathbf{u}_j)$ for each $i \in [d]$, where $\mathbf{U}_j = \text{span}(\mathbf{u}_j)$ for $j \in [i]$. The computation of each \mathbf{v}_i needs $\mathcal{O}((d-2)N)$ flops, where $N = \prod_{j \in [d]} n_j$. Hence each step of AMM for best rank one approximation is $\mathcal{O}(d(d-2)N)$. So for $d = 3$ and $n_1 \approx n_2 \approx n_3$ the complexity is $\mathcal{O}(n^3)$, which is the same complexity as above with $r = 1$.

14.7 Fixed Points of AMM and Newton Method

Consider the AMM as described in Sect. 14.6.1. Assume that the sequence $\psi^{(l)}, l \in \mathbb{N}$ converges to a point $\phi \in \Psi$. Then ϕ is a fixed point of the map:

$$\tilde{\mathbf{F}} : \Psi \rightarrow \Psi, \quad \tilde{\mathbf{F}} = (\tilde{F}_1, \dots, \tilde{F}_d), \quad \tilde{F}_i : \Psi \rightarrow \Psi_i, \quad \tilde{F}_i(\psi) = \psi_i^*(\hat{\psi}_i), \quad \psi = (\psi_i, \hat{\psi}_i), \quad i \in [d]. \quad (14.34)$$

In general, the map $\tilde{\mathbf{F}}$ is a multivalued map, since the maximum given in (14.30) may be achieved at a number of points denoted by $\psi_i^*(\hat{\psi}_i)$. In what follows we assume:

Assumption 1 *The AMM converges to a fixed point ϕ of $\tilde{\mathbf{F}}$ i.e. $\tilde{\mathbf{F}}(\phi) = \phi$, such that the following conditions hold:*

1. *There is a connected open neighborhood $O \subset \Psi$ such that $\tilde{\mathbf{F}} : O \rightarrow O$ is one valued map.*
2. *$\tilde{\mathbf{F}}$ is a contraction on O with respect to some norm on O .*
3. *$\tilde{\mathbf{F}} \in \mathcal{C}^2(O)$, i.e. $\tilde{\mathbf{F}}$ has two continuous partial derivatives in O .*
4. *O is diffeomorphic to an open subset in \mathbb{R}^L . That is, there exists a smooth one-to-one map $H : O \rightarrow \mathbb{R}^L$ such that the Jacobian $D(H)$ is invertible at each point $\psi \in O$.*

Assume that the conditions of Assumption 1 hold. Then the map $\tilde{\mathbf{F}} : O \rightarrow O$ can be represented as

$$\mathbf{F} : O_1 \rightarrow O_1, \quad \mathbf{F} = H \circ \tilde{\mathbf{F}} \circ H^{-1}, \quad O_1 = H(O).$$

Hence to find a fixed point of $\tilde{\mathbf{F}}$ in O it is enough to find a fixed point of \mathbf{F} in O_1 . A fixed point of \mathbf{F} is a zero point of the system

$$\mathbf{G}(\mathbf{x}) = \mathbf{0}, \quad G(\mathbf{x}) := \mathbf{x} - \mathbf{F}(\mathbf{x}). \tag{14.35}$$

To find a zero of \mathbf{G} we use the standard Newton method.

In this paper we propose new Newton methods. We make a few iterations of AMM and switch to a Newton method assuming that the conditions of Assumption 1 hold as explained above. A fixed point of the map $\tilde{\mathbf{F}}$ for best rank one approximation induces a fixed point of map $\mathbf{F} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ [15, Lemma 2]. Then the corresponding Newton method to find a zero of \mathbf{G} is straightforward to state and implement, as explained in the next subsection. This Newton method was given in [14, §5]. See also Zhang-Golub [43] for a different Newton method for best (1, 1, 1) approximation.

Let $\tilde{\mathbf{F}} : \text{Gr}(\mathbf{r}, \mathbf{n}) \rightarrow \text{Gr}(\mathbf{r}, \mathbf{n})$ be the induced map AMM. Each $\text{Gr}(r, \mathbb{R}^n)$ can be decomposed as a compact manifold to a finite number of charts as explained in Sect. 14.8. These charts induce standard charts of $\text{Gr}(\mathbf{r}, \mathbf{n})$. After a few AMM iterations we assume that the neighborhood O of the fixed point of $\tilde{\mathbf{F}}$ lies in one the charts of $\text{Gr}(\mathbf{r}, \mathbf{n})$. We then construct the corresponding map \mathbf{F} in this chart. Next we apply the standard Newton method to \mathbf{G} . The papers by Eldén-Savas [8] and Savas-Lim [39] discuss Newton and quasi-Newton methods for (r_1, r_2, r_3) approximation of 3-tensors using the concepts of differential geometry.

14.7.1 Newton Method for Best Rank One Approximation

Let $\mathcal{T} \in \mathbb{R}^n \setminus \{0\}$. Define:

$$\begin{aligned} \Psi_i &= \mathbb{R}^{n_i}, \quad i \in [d], \quad \Psi = \mathbb{R}^{n_1} \times \dots \times \mathbb{R}^{n_d}, \quad \psi = (\mathbf{x}_1, \dots, \mathbf{x}_d) \in \Psi, \\ f_{\mathcal{T}} : \Psi &\rightarrow \mathbb{R}, \quad f_{\mathcal{T}}(\psi) = \mathcal{T} \times (\otimes_{j \in [d]} \mathbf{x}_j), \end{aligned} \tag{14.36}$$

$$\mathbf{F} = (F_1, \dots, F_d) : \Psi \rightarrow \Psi, \quad F_i(\psi) = \mathcal{T} \times (\otimes_{j \in [d] \setminus \{i\}} \mathbf{x}_j), \quad i \in [d]. \tag{14.37}$$

Recall the results of Sect. 14.5.2: Any critical point of $f_{\mathcal{T}}|_{\mathbf{S}(\mathbf{n})}$ satisfies (14.23). Suppose we start the AMM with $\psi^{(0)} = (\mathbf{x}_1^{(0)}, \dots, \mathbf{x}_d^{(0)}) \in \mathbf{S}(\mathbf{n})$ such that $f_{\mathcal{T}}(\psi^{(0)}) \neq 0$. Then it is straightforward to see that $f_{\mathcal{T}}(\psi^{(l)}) > 0$ for $l \in \mathbb{N}$. Assume that $\lim_{l \rightarrow \infty} \psi^{(l)} = \omega = (\mathbf{u}_1, \dots, \mathbf{u}_d) \in \mathbf{S}(\mathbf{n})$. Then ω is the singular tuple of \mathcal{T} satisfying (14.23). Clearly, $\lambda = f_{\mathcal{T}}(\omega) > 0$. Let

$$\phi = (\mathbf{y}_1, \dots, \mathbf{y}_d) := \lambda^{-\frac{1}{d-2}} \omega = \lambda^{-\frac{1}{d-2}} (\mathbf{u}_1, \dots, \mathbf{u}_d). \tag{14.38}$$

Then ϕ is a fixed point of \mathbf{F} .

Our Newton algorithm for finding the fixed point ϕ of \mathbf{F} corresponding to a fixed point ω of AMM is as follows. We do a number of iterations of AMM to obtain $\psi^{(m)}$. Then we renormalize $\psi^{(m)}$ according to (14.38):

$$\phi_0 := (f_{\mathcal{T}}(\psi^{(m)}))^{-\frac{1}{d-2}} \psi^{(m)}. \quad (14.39)$$

Let $D\mathbf{F}(\psi)$ denote the Jacobian of \mathbf{F} at ψ , i.e. the matrix of partial derivatives of \mathbf{F} at ψ . Then we perform Newton iterations of the form:

$$\phi^{(l)} = \phi^{(l-1)} - (I - D\mathbf{F}(\phi^{(l-1)}))^{-1}(\phi^{(l-1)} - \mathbf{F}(\phi^{(l-1)})), \quad l \in \mathbb{N}. \quad (14.40)$$

After performing a number of Newton iterations we obtain $\phi^{(m')} = (\mathbf{z}_1, \dots, \mathbf{z}_d)$ which is an approximation of ϕ . We then renormalize each \mathbf{z}_i to obtain $\omega^{(m')} := (\frac{1}{\|\mathbf{z}_1\|}\mathbf{z}_1, \dots, \frac{1}{\|\mathbf{z}_d\|}\mathbf{z}_d)$ which is an approximation to the fixed point ω . We call this Newton method *Newton-1*.

We now give the explicit formulas for 3-tensors, where $n_1 = m, n_2 = n, n_3 = l$. First

$$\mathbf{F}(\mathbf{u}, \mathbf{v}, \mathbf{w}) := (\mathcal{T} \times (\mathbf{v} \otimes \mathbf{w}), \mathcal{T} \times (\mathbf{u} \otimes \mathbf{w}), \mathcal{T} \times (\mathbf{u} \otimes \mathbf{v})), \quad \mathbf{G} := (\mathbf{u}, \mathbf{v}, \mathbf{w}) - \mathbf{F}(\mathbf{u}, \mathbf{v}, \mathbf{w}). \quad (14.41)$$

Then

$$D\mathbf{G}(\mathbf{u}, \mathbf{v}, \mathbf{w}) = \begin{bmatrix} I_m & -\mathcal{T} \times \mathbf{w} & -\mathcal{T} \times \mathbf{v} \\ -(\mathcal{T} \times \mathbf{w})^\top & I_n & -\mathcal{T} \times \mathbf{u} \\ -(\mathcal{T} \times \mathbf{v})^\top & -(\mathcal{T} \times \mathbf{u})^\top & I_l \end{bmatrix}. \quad (14.42)$$

Hence Newton-1 iteration is given by the formula

$$(\mathbf{u}_{i+1}, \mathbf{v}_{i+1}, \mathbf{w}_{i+1}) = (\mathbf{u}_i, \mathbf{v}_i, \mathbf{w}_i) - (D\mathbf{G}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{w}_i))^{-1} \mathbf{G}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{w}_i),$$

for $i = 0, 1, \dots$. Here we abuse notation by viewing $(\mathbf{u}, \mathbf{v}, \mathbf{w})$ as a column vector $(\mathbf{u}^\top, \mathbf{v}^\top, \mathbf{w}^\top)^\top \in \mathbb{C}^{m+n+l}$.

Numerically, to find $(D\mathbf{G}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{w}_i))^{-1} \mathbf{G}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{w}_i)$ one solves the linear system

$$(D\mathbf{G}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{w}_i))(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \mathbf{G}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{w}_i).$$

The final vector $(\mathbf{u}_j, \mathbf{v}_j, \mathbf{w}_j)$ of Newton-1 iterations is followed by a scaling to vectors of unit length $\mathbf{x}_j = \frac{1}{\|\mathbf{u}_j\|} \mathbf{u}_j, \mathbf{y}_j = \frac{1}{\|\mathbf{v}_j\|} \mathbf{v}_j, \mathbf{z}_j = \frac{1}{\|\mathbf{w}_j\|} \mathbf{w}_j$.

We now discuss the complexity of Newton-1 method for $d = 3$. Assuming that $m \approx n \approx l$ we deduce that the computation of the matrix $D\mathbf{G}$ is $\mathcal{O}(n^3)$. As the dimension of $D\mathbf{G}$ is $m + n + l$ it follows that the complexity of each iteration of Newton-1 method is $\mathcal{O}(n^3)$.

14.8 Newton Method for Best r -Approximation

Recall that an r -dimensional subspace $\mathbf{U} \in \text{Gr}(r, \mathbb{R}^n)$ is given by a matrix $U = [u_{ij}]_{i,j=1}^{n,r} \in \mathbb{R}^{n \times r}$ of rank r . In particular there is a subset $\alpha \subset [n]$ of cardinality r so that $\det U[\alpha, [r]] \neq 0$. Here $\alpha = (\alpha_1, \dots, \alpha_r), 1 \leq \alpha_1 < \dots < \alpha_r \leq n$. So $U[\alpha, [r]] := [u_{\alpha_i, j}]_{i,j=1}^r \in \text{GL}(r, \mathbb{R})$, (the group of invertible matrices). Clearly, $V := UU[\alpha, [r]]^{-1}$ represents another basis in \mathbf{U} . Note that $V[\alpha, [r]] = I_r$. Hence the set of all $V \in \mathbb{R}^{n \times r}$ with the condition: $V[\alpha, [r]] = I_r$ represent an open cell in $\text{Gr}(r, n)$ of dimension $r(n - r)$ denoted by $\text{Gr}(r, \mathbb{R}^n)(\alpha)$. (The number of free parameters in all such V 's is $(n - r)r$.) Assume for simplicity of exposition that $\alpha = [r]$. Note that $V_0 = \begin{bmatrix} I_r \\ 0 \end{bmatrix} \in \text{Gr}(r, \mathbb{R}^n)([r])$. Let $\mathbf{e}_i = (\delta_{1i}, \dots, \delta_{ni})^\top \in \mathbb{R}^n, i = 1, \dots, n$ be the standard basis in \mathbb{R}^n . So $\mathbf{U}_0 = \text{span}(\mathbf{e}_1, \dots, \mathbf{e}_r) \in \text{Gr}(r, \mathbb{R}^n)([r])$, and V_0 is the unique representative of \mathbf{U}_0 . Note that \mathbf{U}_0^\perp , the orthogonal complement of \mathbf{U}_0 , is $\text{span}(\mathbf{e}_{r+1}, \dots, \mathbf{e}_n)$. It is straightforward to see that $\mathbf{V} \in \text{Gr}(r, \mathbb{R}^n)([r])$ if and only if $\mathbf{V} \cap \mathbf{U}_0^\perp = \{\mathbf{0}\}$.

The following definition is a geometric generalization of $\text{Gr}(r, \mathbb{R}^n)(\alpha)$:

$$\text{Gr}(r, \mathbb{R}^n)(\mathbf{U}) := \{\mathbf{V} \in \text{Gr}(r, \mathbb{R}^n), \mathbf{V} \cap \mathbf{U}^\perp = \{\mathbf{0}\}\} \text{ for } \mathbf{U} \in \text{Gr}(r, \mathbb{R}^n). \tag{14.43}$$

A basis for $\text{Gr}(r, \mathbb{R}^n)(\mathbf{U})$, which can be identified the tangent hyperplane $T_{\mathbf{U}} \text{Gr}(r, \mathbb{R}^n)$, can be represented as $\oplus^r \mathbf{U}^\perp$: Let $\mathbf{u}_1, \dots, \mathbf{u}_r$ and $\mathbf{u}_{r+1}, \dots, \mathbf{u}_n$ be orthonormal bases of \mathbf{U} and \mathbf{U}^\perp respectively. Then each subspace $\mathbf{V} \in \text{Gr}(r, \mathbb{R}^n)(\mathbf{U})$ has a unique basis of the form $\mathbf{u}_1 + \mathbf{x}_1, \dots, \mathbf{u}_r + \mathbf{x}_r$ for unique $\mathbf{x}_1, \dots, \mathbf{x}_r \in \mathbf{U}^\perp$. Equivalently, every matrix $X \in \mathbb{R}^{(n-r) \times r}$ induces a unique subspace \mathbf{V} using the equality

$$[\mathbf{x}_1 \ \dots \ \mathbf{x}_r] = [\mathbf{u}_1 \ \dots \ \mathbf{u}_{n-r}]X \text{ for each } X \in \mathbb{R}^{(n-r) \times r}. \tag{14.44}$$

Recall the results of Sect. 14.5.3. Let $\underline{U} = (U_1, \dots, U_d) \in \text{Gr}(\mathbf{r}, \mathbf{n})$. Then

$$\tilde{\mathbf{F}} = (\tilde{F}_1, \dots, \tilde{F}_d) : \text{Gr}(\mathbf{r}, \mathbf{n}) \rightarrow \text{Gr}(\mathbf{r}, \mathbf{n}), \quad \tilde{F}_i(\underline{U}) = \mathbf{U}_i(\underline{U}_i), \quad i \in [d], \tag{14.45}$$

where $\mathbf{U}_i(\underline{U}_i)$ a subspace spanned by the first r_i eigenvectors of $A_i(\underline{U}_i)$. Assume that $\tilde{\mathbf{F}}$ is one valued at \underline{U} , i.e. (14.28) holds. Then it is straightforward to show that $\tilde{\mathbf{F}}$ is smooth (real analytic) in neighborhood of \underline{U} . Assume next that there exists a neighborhood O of \underline{U} such that

$$O \subset \text{Gr}(\mathbf{r}, \mathbf{n})(\underline{U}) := \text{Gr}(r_1, \mathbb{R}^{n_1})(\mathbf{U}_1) \times \dots \times \text{Gr}(r_d, \mathbb{R}^{n_d})(\mathbf{U}_d), \quad \underline{U} = (\mathbf{U}_1, \dots, \mathbf{U}_d), \tag{14.46}$$

such that the conditions I–3 of Assumption 1 hold. Observe next that $\text{Gr}(\mathbf{r}, \mathbf{n})(\underline{U})$ is diffeomorphic to

$$\mathbb{R}^L := \mathbb{R}^{(n_1-r_1) \times r_1} \dots \times \mathbb{R}^{(n_d-r_d) \times r_d}, \quad L = \sum_{i \in [d]} (n_i - r_i)r_i$$

We say that $\tilde{\mathbf{F}}$ is *regular* at \underline{U} if in addition to the above condition the matrix $I - D\tilde{\mathbf{F}}(\underline{U})$ is invertible. We can view $X = [X_1 \dots X_d] \in \mathbb{R}^{(n_1-r_1) \times r_1} \dots \times \mathbb{R}^{(n_d-r_d) \times r_d}$. Then $\tilde{\mathbf{F}}$ on O can be viewed as

$$\mathbf{F}: O_1 \rightarrow O_1, \quad O_1 \subset \mathbb{R}^L, \quad \mathbf{F}(X) = [F_1(X), \dots, F_d(X)], \quad X = [X_1 \dots X_d] \in \mathbb{R}^L. \quad (14.47)$$

Note that $F_i(X)$ does not depend on X_i for each $i \in [d]$. In our numerical simulations we first do a small number of AMM and then switch to Newton method given by (14.40). Observe that \underline{U} corresponds to $X(\underline{U}) = [X_1(\underline{U}), \dots, X_d(\underline{U})]$. When referring to (14.40) we identify $X = [X_1, \dots, X_d]$ with $\phi = (\phi_1, \dots, \phi_d)$ and no ambiguity will arise.

Note that the case $\mathbf{r} = \mathbf{1}_d$ corresponds to best rank one approximation. The above Newton method in this case is different from Newton method given in Sect. 14.7.1.

14.9 A Closed Formula for $D\mathbf{F}(X(\underline{U}))$

Recall the definitions and results of Sect. 14.5.3. Given \underline{U} we compute $\tilde{F}_i(\underline{U}) = \mathbf{U}_i(\underline{U}_i)$, which is the subspace spanned by the first r_i eigenvectors of $A_i(\underline{U}_i)$, which is given by (14.27), for $i \in [d]$. Assume that (14.33) holds. Let

$$\begin{aligned} \mathbf{U}_i(\underline{U}_i) &= \text{span}(\mathbf{v}_{1,i}, \dots, \mathbf{v}_{r_i,n_i}), \quad \mathbf{U}_i(\underline{U}_i)^\perp = \text{span}(\mathbf{v}_{r_i+1,i}, \dots, \mathbf{v}_{n_i,i}), \\ \mathbf{v}_{j,i}^\top \mathbf{v}_{k,i} &= \delta_{jk}, \quad j, k \in [n_i], \quad i \in [d]. \end{aligned} \quad (14.48)$$

With each $X = [X_1, \dots, X_d] \in \mathbb{R}^L$ we associate the following point $(\mathbf{W}_1, \dots, \mathbf{W}_d) \in \text{Gr}(\mathbf{r}, \mathbf{n})(\underline{U})$. Suppose that $X_i = [x_{pq,i}] \in \mathbb{R}^{(n_i-r_i) \times r_i}$. Then \mathbf{W}_i has a basis of the form

$$\mathbf{u}_{j_i,i} + \sum_{k_i \in [n_i-r_i]} x_{k_i j_i,i} \mathbf{u}_{r_i+k_i,i}, \quad j_i \in [r_i].$$

One can use the following notation for a basis $\mathbf{w}_{1,i}, \dots, \mathbf{w}_{r_i,i}$, written as a vector with vector coordinates $[\mathbf{w}_{1,i} \cdots \mathbf{w}_{r_i,i}]$:

$$[\mathbf{w}_{1,i} \cdots \mathbf{w}_{r_i,i}] = [\mathbf{u}_{1,i} \cdots \mathbf{u}_{r_i,i}] + [\mathbf{u}_{r_i+1,i} \cdots \mathbf{u}_{n_i,i}] X_i, \quad i \in [d]. \quad (14.49)$$

Note that to the point $\underline{U} \in \text{Gr}(\mathbf{r}, \mathbf{n})(U)$ corresponds the point $X = 0$. Since $\mathbf{u}_{1,i}, \dots, \mathbf{u}_{n_i,i}$ is a basis in \mathbb{R}^{n_i} it follows that

$$\begin{aligned} [\mathbf{v}_{1,i} \cdots \mathbf{v}_{r_i,i}] &= [\mathbf{u}_{1,i} \cdots \mathbf{u}_{r_i,i}]Y_{i,0} + [\mathbf{u}_{r_i+1,i}, \dots, \mathbf{u}_{n_i,i}]X_{i,0} = [\mathbf{u}_{1,i} \cdots \mathbf{u}_{n_i,i}]Z_{i,0}, \\ Y_{i,0} \in \mathbb{R}^{r_i \times r_i}, X_{i,0} \in \mathbb{R}^{(n_i-r_i) \times r_i}, Z_{i,0} &= \begin{bmatrix} Y_{i,0} \\ X_{i,0} \end{bmatrix} \in \mathbb{R}^{n_i \times r_i}, \text{ for } i \in [d]. \end{aligned} \quad (14.50)$$

View $[\mathbf{v}_{1,i} \cdots \mathbf{v}_{r_i,i}]$, $[\mathbf{u}_{1,i} \cdots \mathbf{u}_{n_i,i}]$ as $n_i \times r_i$ and $n_i \times n_i$ matrices with orthonormal columns. Then

$$Z_{i,0} = [\mathbf{u}_{1,i} \cdots \mathbf{u}_{n_i,i}]^\top [\mathbf{v}_{1,i} \cdots \mathbf{v}_{r_i,i}], \quad i \in [d]. \quad (14.51)$$

The assumption that $\tilde{\mathbf{F}} : O \rightarrow O$ implies that $Y_{i,0}$ is an invertible matrix. Hence $[\mathbf{v}_{1,i} \cdots \mathbf{v}_{r_i,i}]Y_{i,0}^{-1}$ is also a basis in $\mathbf{U}_i(\underline{U}_i)$. Clearly,

$$[\mathbf{v}_{1,i} \cdots \mathbf{v}_{r_i,i}]Y_{i,0}^{-1} = [\mathbf{u}_{1,i} \cdots \mathbf{u}_{r_i,i}] + [\mathbf{u}_{r_i+1,i}, \dots, \mathbf{u}_{n_i,i}]X_{i,0}Y_{i,0}^{-1}, \quad i \in [d].$$

Hence $\tilde{\mathbf{F}}(\underline{U})$ corresponds to $\mathbf{F}(0)$ where

$$F_i(0) = X_{i,0}Y_{i,0}^{-1}, \quad i \in [d], \quad \mathbf{F}(0) = (F_1(0), \dots, F_d(0)). \quad (14.52)$$

We now find the matrix of derivatives. So $D_i F_j \in \mathbb{R}^{((n_i-r_i)r_i \times ((n_j-r_j)r_j))}$ is the partial derivative matrix of $(n_j - r_j)r_j$ coordinates of F_j with respect to $(n_i - r_i)r_i$ the coordinates of \mathbf{U}_i viewed as the matrix $\begin{bmatrix} I_{r_i} \\ G_i \end{bmatrix}$. So $G_i \in \mathbb{R}^{(n_i-r_i) \times r_i}$ are the variables representing the subspace \mathbf{U}_i . Observe first that $D_i F_i = 0$ just as in Newton method for best rank one approximation in Sect. 14.7.1.

Let us now find $D_i F_j(0)$. Recall that $D_i F_j(0)$ is a matrix of size $(n_i - r_i)r_i \times (n_j - r_j)r_j$. The entries of $D_i F_j(0)$ are indexed by $((p, q), (s, t))$ as follows: The entries of $G_i = [g_{pq,i}] \in \mathbb{R}^{(n_i-r_i) \times r_i}$ are viewed as $(n_i - r_i)r_i$ variables, and are indexed by (p, q) , where $p \in [n_i - r_i], q \in [r_i]$. F_j is viewed as a matrix $G_j \in \mathbb{R}^{(n_j-r_j) \times r_j}$. The entries of F_j are indexed by (s, t) , where $s \in [n_j - r_j]$ and $t \in [r_j]$. Since $\underline{U} \in \text{Gr}(\mathbf{r}, \mathbf{n})(U)$ corresponds to $0 \in \mathbb{R}^L$ we denote by $A_j(0)$ the matrix $A_j(\underline{U}_j)$ for $j \in [d]$. We now give the formula for $\frac{\partial A_j(0)}{\partial g_{pq,i}}$. This is done by noting that we vary \mathbf{U}_i by changing the orthonormal basis $\mathbf{u}_{1,i}, \dots, \mathbf{u}_{r_i,i}$ up to the first perturbation with respect to the real variable ε to

$$\hat{\mathbf{u}}_{1,i} = \mathbf{u}_{1,i}, \dots, \hat{\mathbf{u}}_{q-1,i} = \mathbf{u}_{q-1,i}, \hat{\mathbf{u}}_{q,i} = \mathbf{u}_{q,i} + \varepsilon \mathbf{u}_{r_i+p,i}, \hat{\mathbf{u}}_{q+1,i} = \mathbf{u}_{q+1,i}, \dots, \hat{\mathbf{u}}_{r_i,i} = \mathbf{u}_{r_i,i}$$

We denote the subspace spanned by these vectors as $\mathbf{U}_i(\varepsilon, p, q)$. That is, we change only the q orthonormal vector of the standard basis in \mathbf{U}_i , for $q = 1, \dots, r_i$. The new basis is an orthogonal basis, and up order ε , the vector $\mathbf{u}_{q,i} + \varepsilon \mathbf{u}_{r_i+p,i}$ is also of length 1. Let $\underline{U}(\varepsilon, i, p, q) = (\mathbf{U}_1, \dots, \mathbf{U}_{i-1}, \mathbf{U}_i(\varepsilon, p, q), \mathbf{U}_{i+1}, \dots, \mathbf{U}_d)$. Then

$\underline{U}(\varepsilon, i, p, q)_j$ is obtained by dropping the subspace \mathbf{U}_j from $\underline{U}(\varepsilon, i, p, q)$. We will show that

$$A_j(\underline{U}(\varepsilon, i, p, q)_j) = A_j(\underline{U}_j) + \varepsilon B_{j,i,p,q} + O(\varepsilon^2). \quad (14.53)$$

We now give a formula to compute $B_{j,i,p,q}$. Assume that $i, j \in [d]$ is a pair of different integers. Let J be a set of $d - 2$ pairs $\cup_{l \in [d] \setminus \{i,j\}} \{(k_l, l)\}$, where $k_l \in [r_l]$. Denote by \mathcal{J}_{ij} the set of all such J 's. Note that $\mathcal{J}_{ij} = \mathcal{J}_{ji}$. Furthermore, the number of elements in \mathcal{J}_{ij} is $R_{ij} = \prod_{l \in [d] \setminus \{i,j\}} r_l$. We now introduce the following matrices

$$C_{ij}(J) := \mathcal{T} \times (\otimes_{(k,l) \in J} \mathbf{u}_{k,l}) \in \mathbb{R}^{n_i \times n_j}, \quad J \in \mathcal{J}_{ij}. \quad (14.54)$$

Note that $C_{ij}(J) = C_{ji}(J)^\top$.

Lemma 1 *Let $i, j \in [d], i \neq j$. Assume that $p \in [n_i - r_i], q \in [r_i]$. Then (14.53) holds. Furthermore*

$$A_j(\underline{U}_j) = \sum_{k \in [r_i], J \in \mathcal{J}_{ji}} (C_{ji}(J) \mathbf{u}_{k,i})(C_{ji}(J) \mathbf{u}_{k,i})^\top, \quad (14.55)$$

$$B_{j,i,p,q} = \sum_{J \in \mathcal{J}_{ji}} (C_{ji}(J) \mathbf{u}_{k_i+p,i})(C_{ji}(J) \mathbf{u}_{q,i})^\top + (C_{ji}(J) \mathbf{u}_{q,i})(C_{ji}(J) \mathbf{u}_{k_i+p,i})^\top \quad (14.56)$$

Proof The identity of (14.55) is just a restatement of (14.27). To compute $A_j(\underline{U}(\varepsilon, i, p, q)_j)$ use (14.54) by replacing $u_{k,i}$ with $\hat{u}_{k,i}$ for $k \in [n_i]$. Deduce first (14.53) and then (14.56). \square

Recall that $\mathbf{v}_{1,j}, \dots, \mathbf{v}_{r_j,j}$ is an orthonormal basis of $\mathbf{U}_j(\underline{U}_j)$, and these vectors are the eigenvectors $A_j(\underline{U}_j)$ corresponding its first r_j eigenvalues. Let $\mathbf{v}_{r_j+1,j}, \dots, \mathbf{v}_{n_j,i}$ be the last $n_j - r_j$ orthonormal eigenvectors of $A_j(\underline{U}_j)$. We now find the first perturbation of the first r_i eigenvectors for the matrix $A_j(\underline{U}_j) + \varepsilon B_{j,i,p,q}$. Assume first, for simplicity of exposition, that each $\lambda_k(A_j(\underline{U}_j))$ is simple for $k \in [r_j]$: Then it is known, e.g. [10, Chapter 4, §19, (4.19.2)]:

$$\mathbf{v}_{k,j}(\varepsilon, i, p, q) = \mathbf{v}_{k,j} + \varepsilon (\lambda_k(A_j(\underline{U}_j)) I_{n_j} - A_j(\underline{U}_j))^\dagger B_{j,i,p,q} \mathbf{v}_{k,j} + O(\varepsilon^2), \quad k \in [r_j]. \quad (14.57)$$

The assumption that $\lambda_k(A_j(\underline{U}_j))$ is a simple eigenvalue for $k \in [r_j]$ yields

$$(\lambda_k(A_j(\underline{U}_j)) I_{n_j} - A_j(\underline{U}_j))^\dagger \mathbf{y} = \sum_{l \in [n_j] \setminus \{k\}} \frac{1}{\lambda_k(A_j(\underline{U}_j)) - \lambda_l(A_j(\underline{U}_j))} (\mathbf{v}_{l,j}^\top \mathbf{y}) \mathbf{v}_{l,j},$$

for $\mathbf{y} \in \mathbb{R}^{n_j}$.

Since we are interested in a basis of $U_j(\underline{U}(\varepsilon, i, p, q)_j)$ up to the order of ε we can assume that this basis is of the form

$$\tilde{\mathbf{v}}_{k,j}(\varepsilon, i, p, q) = \mathbf{v}_{k,j} + \varepsilon \mathbf{w}_{k,j}(i, p, q), \quad \mathbf{w}_{k,j}(i, p, q) \in \text{span}(\mathbf{v}_{r_j+1,j}, \dots, \mathbf{v}_{n_j,j}).$$

Hence

$$\begin{aligned} \mathbf{w}_{k,j}(i, p, q) &= \sum_{l \in [n_j] \setminus [r_j]} \frac{1}{\lambda_k(A_j(\underline{U}_j)) - \lambda_l(A_j(\underline{U}_j))} (\mathbf{v}_{l,j}^\top \mathbf{c}_{k,j,i,p,q}) \mathbf{v}_{l,j}, \\ \mathbf{c}_{k,j,i,p,q} &:= B_{j,i,p,q} \mathbf{v}_{k,j}. \end{aligned} \tag{14.58}$$

Note that the assumption (14.28) yields that $\mathbf{w}_{k,j}$ is well defined for $k \in [r_j]$. Let

$$\begin{aligned} W_j(i, p, q) &= [\mathbf{w}_{1,j}(i, p, q) \cdots \mathbf{w}_{r_j,j}(i, p, q)] = \begin{bmatrix} V_j(i, p, q) \\ U_j(i, p, q) \end{bmatrix}, \\ V_j(i, p, q) &\in \mathbb{R}^{r_j \times r_j}, \quad U_j(i, p, q) \in \mathbb{R}^{(n_j-r_j) \times r_j}. \end{aligned}$$

Up to the order of ε we have that a basis of $U_j(\underline{U}(\varepsilon, i, p, q)_j)$ is given by columns of matrix $Z_{j,0} + \varepsilon W_j(i, p, q) = \begin{bmatrix} Y_{j,0} + \varepsilon V_j(i, p, q) \\ X_{j,0} + \varepsilon U_j(i, p, q) \end{bmatrix}$. Note that

$$\begin{aligned} &(Z_{j,0} + \varepsilon W_j(i, p, q))(Y_{j,0} + \varepsilon V_j(i, p, q))^{-1} \\ &= \begin{bmatrix} I_{r_j} \\ (X_{j,0} + \varepsilon U_j(i, p, q))(Y_{j,0} + \varepsilon V_j(i, p, q))^{-1} \end{bmatrix}. \end{aligned}$$

Observe next

$$\begin{aligned} Y_{j,0} + \varepsilon V_j(i, p, q) &= Y_{j,0}(I_{r_j} + \varepsilon Y_{j,0}^{-1} V_j(i, p, q)), \\ (Y_{j,0} + \varepsilon V_j(i, p, q))^{-1} &= (I_{r_j} + \varepsilon Y_{j,0}^{-1} V_j(i, p, q))^{-1} Y_{j,0}^{-1} = \\ &Y_{j,0}^{-1} - \varepsilon Y_{j,0}^{-1} V_j(i, p, q) Y_{j,0}^{-1} + O(\varepsilon^2), \\ (X_{j,0} + \varepsilon U_j(i, p, q))(Y_{j,0} + \varepsilon V_j(i, p, q))^{-1} &= \\ &X_{j,0} Y_{j,0}^{-1} + \varepsilon (U_j(i, p, q) Y_{j,0}^{-1} - X_{j,0} Y_{j,0}^{-1} V_j(i, p, q) Y_{j,0}^{-1}) + O(\varepsilon^2). \end{aligned}$$

Hence

$$\frac{\partial F_j}{\partial \mathbf{g}_{pq,i}}(0) = U_j(i, p, q) Y_{j,0}^{-1} - X_{j,0} Y_{j,0}^{-1} V_j(i, p, q) Y_{j,0}^{-1}. \tag{14.59}$$

Thus $D\mathbf{F}(0) = [D_i F_j]_{i,j \in [d]} \in \mathbb{R}^{L \times L}$. We now make one iteration of Newton method given by (14.40) for $l = 1$, where $\phi^{(0)} = 0$:

$$\phi^{(1)} = -(I - D\mathbf{F}(0))^{-1} F(0), \quad \phi^{(1)} = [X_{1,1}, \dots, X_{d,1}] \in \mathbb{R}^L. \quad (14.60)$$

Let $\mathbf{U}_{i,1} \in \text{Gr}(r_i, \mathbb{R}^{n_i})$ be the subspace represented by the matrix $X_{i,1}$:

$$\mathbf{U}_{i,1} = \text{span}(\tilde{\mathbf{u}}_{1,i,1}, \dots, \tilde{\mathbf{u}}_{r_i,i,1}), \quad [\tilde{\mathbf{u}}_{1,i,1}, \dots, \tilde{\mathbf{u}}_{n_i,i,1}] = [\mathbf{u}_{1,i}, \dots, \mathbf{u}_{n_i,i}] \begin{bmatrix} I_{r_i} \\ X_{j,1} \end{bmatrix} \quad (14.61)$$

for $i \in [d]$. Perform the Gram-Schmidt process on $\tilde{\mathbf{u}}_{1,i,1}, \dots, \tilde{\mathbf{u}}_{r_i,i,1}$ to obtain an orthonormal basis $\mathbf{u}_{1,i,1}, \dots, \mathbf{u}_{r_i,i,1}$ of $\mathbf{U}_{i,1}$. Let $\underline{\mathbf{U}} := (\mathbf{U}_{1,1}, \dots, \mathbf{U}_{d,1})$ and repeat the algorithm which is described above. We call this Newton method *Newton-2*.

14.10 Complexity of Newton-2

In this section we assume for simplicity that $d = 3$, $r_1 = r_2 = r_3 = r$, $n_i \approx n$ for $i \in [3]$. We assume that executed a number of times the AMM for a given $\mathcal{T} \in \mathbb{R}^n$. So we are given $\underline{\mathbf{U}} = (\mathbf{U}_1, \dots, \mathbf{U}_d)$, and an orthonormal basis $\mathbf{u}_{1,i}, \dots, \mathbf{u}_{r,i}$ of \mathbf{U}_i for $i \in [d]$. First we complete each $\mathbf{u}_{1,i}, \dots, \mathbf{u}_{r,i}$ to an orthonormal basis $\mathbf{u}_{1,i}, \dots, \mathbf{u}_{n_i,i}$ of \mathbb{R}^{n_i} , which needs $\mathcal{O}(n^3)$ flops. Since $d = 3$ we still need only $\mathcal{O}(n^3)$ to carry out this completion for each $i \in [3]$.

Next we compute the matrices $C_{ij}(J)$. Since $d = 3$, we need n flops to compute each entry of $C_{ij}(J)$. Since we have roughly n^2 entries, the complexity of computing $C_{ij}(J)$ is $\mathcal{O}(n^3)$. As the cardinality of \mathcal{J}_{ij} is r we need $\mathcal{O}(rn^3)$ flops to compute all $C_{ij}(J)$ for $J \in \mathcal{J}_{ij}$. As the number of pairs in [3] is 3 it follows that the complexity of computing all $C_{ij}(J)$ is $\mathcal{O}(rn^3)$.

The identity (14.55) yields that the complexity of computing $A_j(\underline{\mathbf{U}}_j)$ is $\mathcal{O}(r^2n^2)$. Recall next that $A_j(\underline{\mathbf{U}}_j)$ is $n_j \times n_j$ symmetric positive semi-definite matrix. The complexity of computations of the eigenvalues and the orthonormal eigenvectors of $A_j(\underline{\mathbf{U}}_j)$ is $\mathcal{O}(n^3)$. Hence the complexity of computing $\underline{\mathbf{U}}$ is $\mathcal{O}(rn^3)$, as we pointed out at the end of Sect. 14.6.3.

The complexity of computing $B_{j,i,p,q}$ using (14.56) is $\mathcal{O}(rn^2)$. The complexity of computing $\mathbf{w}_{k,j}(i, p, q)$, given by (14.58) is $\mathcal{O}(n^2)$. Hence the complexity of computing $W_j(i, p, q)$ is $\mathcal{O}(rn^2)$. Therefore the complexity of computing $D_i F_j$ is $\mathcal{O}(r^2n^3)$. Since $d = 3$, the complexity of computing the matrix $D\mathbf{F}(0)$ is also $\mathcal{O}(r^2n^3)$.

As $D\mathbf{F}(0) \in \mathbb{R}^{L \times L}$, where $L \approx 3rn$, the complexity of computing $(I - D\mathbf{F}(0))^{-1}$ is $\mathcal{O}(r^3n^3)$. In summary, the complexity of one step in Newton-2 is $\mathcal{O}(r^3n^3)$.

14.11 Numerical Results

We have implemented a Matlab library tensor decomposition using Tensor Toolbox given by [30]. The performance was measured via the actual CPU-time (seconds) needed to compute. All performance tests have been carried out on a 2.8 GHz Quad-Core Intel Xeon Macintosh computer with 16 GB RAM. The performance results are discussed for real data sets of third-order tensors. We worked with a real computer tomography (CT) data set (the so-called MELANIX data set of OsiriX) [15].

Our simulation results are averaged over 10 different runs of the each algorithm. In each run, we changed the initial guess, that is, we generated new random start vectors. We always initialized the algorithms by random start vectors, because this is cheaper than the initialization via HOSVD. We note here that for Newton methods our initial guess is the subspaces returned by one iteration of AMM method.

All the alternating algorithms have the same stopping criterion where convergence is achieved if one of the two following conditions are met: $iterations > 10$; $fitchange < 0.0001$ is met. All the Newton algorithms have the same stopping criterion where convergence is achieved if one of the two following conditions are met: $iterations > 10$; $change < \exp(-10)$.

Our numerical simulations demonstrate the well known fact that for large size tensors Newton methods are not efficient. Though the Newton methods converge in fewer iterations than alternating methods, the computation associated with the matrix of derivatives (Jacobian) in each iteration is too expensive making alternating maximization methods much more cost effective. Our simulations also demonstrate that our Newton-1 for best rank one approximation is as fast as AMM methods. However our Newton-2 is much slower than alternating methods. We also give a comparison between our Newton-2 and the Newton method based on Grassmannian manifold by [8], abbreviated as Newton-ES.

We also observe that for large tensors and large rank approximation two alternating maximization methods, namely MAMM and 2AMMV, seem to outperform the other alternating maximization methods. We would recommend Newton-1 for rank one approximation in case of rank one approximation both for large and small sized tensors. For higher rank approximation we recommend 2AMMV in case of large size tensors and AMM or MAMM in case of small size tensors.

Our Newton-2 performs a bit slower than Newton-ES, however we would like to point couple of advantages. Our method can be easily extendable to higher dimensions (for $d > 3$ case) both analytically and numerically compared to Newton-ES. Our method is also highly parallelizable which can bring down the computation time drastically. Computation of $D_i F_j$ matrices in each iteration contributes to about 50 % of the total time, which however can be parallelizable. Finally the number of iterations in Newton-2 is at least 30 % less than in Newton-ES (Figs. 14.1–14.3).

It is not only important to check how fast the different algorithms perform but also what quality they achieve. This was measured by checking the Hilbert-Schmidt

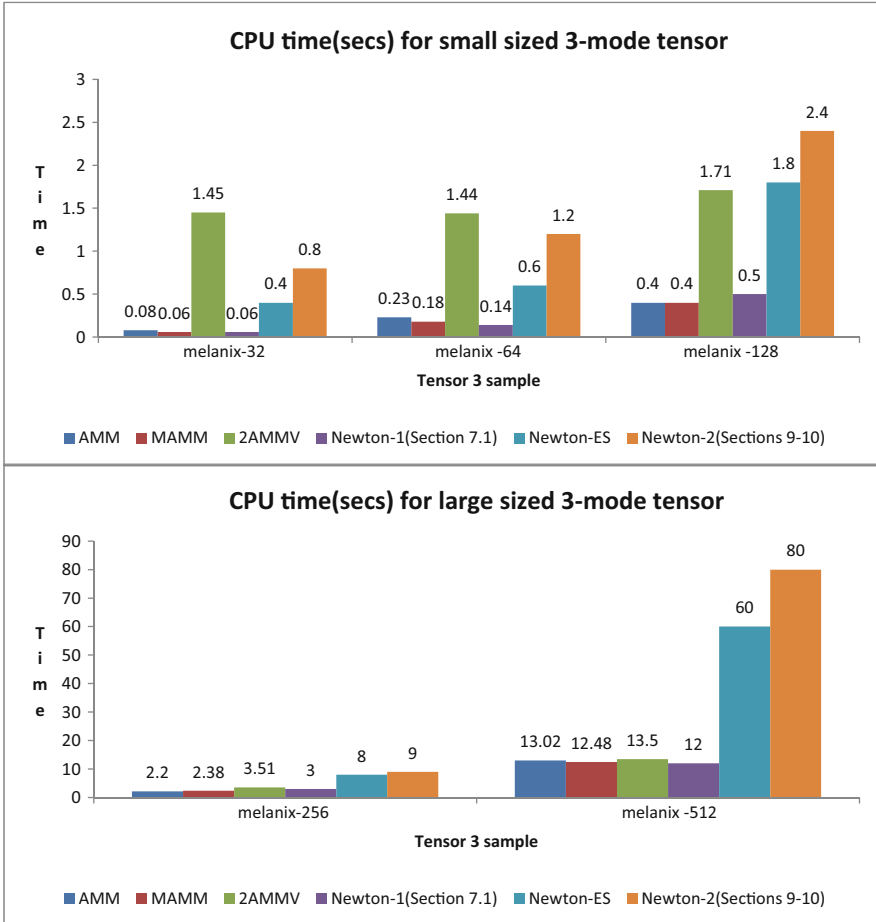


Fig. 14.1 Rank (1, 1, 1) average CPU times taken over 10 initial random guesses

norm, abbreviated as HS norm, of the resulting decompositions, which serves as a measure for the quality of the approximation. In general, we can say that the higher the HS norm, the more likely it is that we find a global maximum. Accordingly, we compared the HS norms to say whether the different algorithms converged to the same stationary point. In Figs. 14.4 and 14.5, we show the average HS norms achieved by different algorithms and compared them with the input norm. We observe all the algorithms seem to attain the same local maximum.

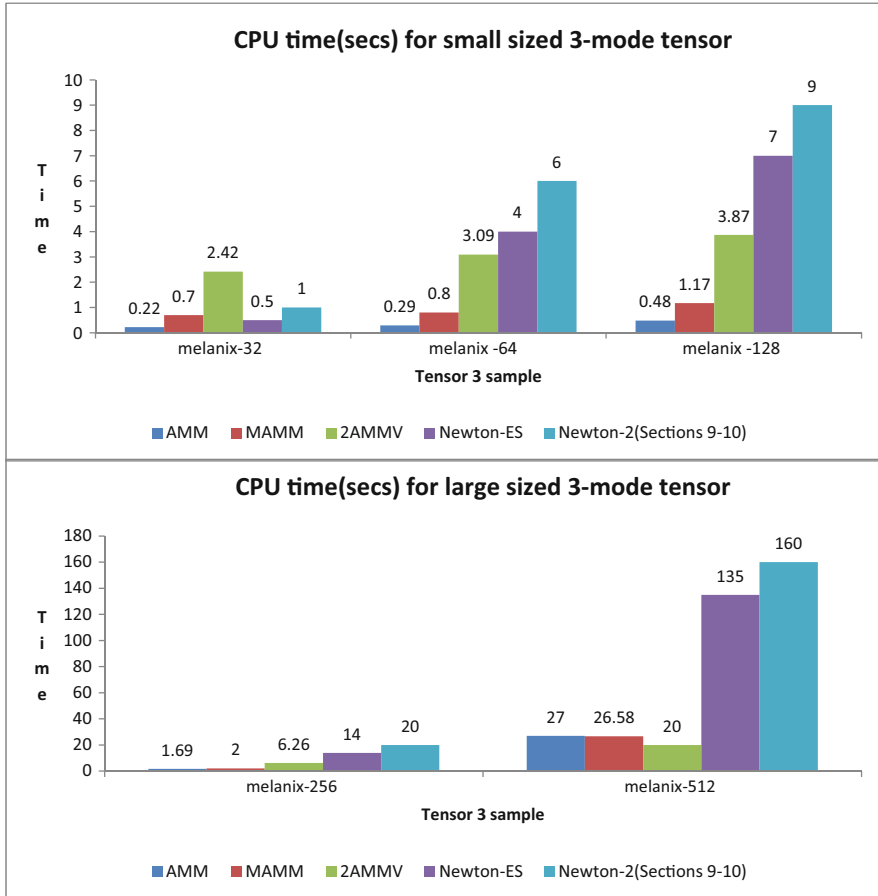


Fig. 14.2 Rank (2, 2, 2) average CPU times taken over 10 initial random guesses

14.11.1 Best (2,2,2) and Rank Two Approximations

Assume that \mathcal{T} is a 3-tensor of rank three at least and let \mathcal{S} be a best (2, 2, 2)-approximation to \mathcal{T} given by (14.1). It is easy to show that \mathcal{S} has at least rank 2. Let $\mathcal{S}' = [s_{j_1, j_2, j_3}] \in \mathbb{R}^{2 \times 2 \times 2}$ be the core tensor corresponding to \mathcal{S} . Clearly $\text{rank } \mathcal{S} = \text{rank } \mathcal{S}' \geq 2$. Recall that a real nonzero $2 \times 2 \times 2$ tensor has rank one, two or three [41]. So $\text{rank } \mathcal{S} \in \{2, 3\}$. Observe next that if $\text{rank } \mathcal{S} = \text{rank } \mathcal{S}' = 2$ then \mathcal{S} is also a best rank two approximation of \mathcal{T} . Recall that a best rank two approximation of \mathcal{T} may not always exist. In particular where $\text{rank } \mathcal{T} > 2$ and the border rank of \mathcal{T} is 2 [6]. In all our numerical simulations for best (2, 2, 2)-approximation we performed on random large tensors, the tensor \mathcal{S}' had rank two.

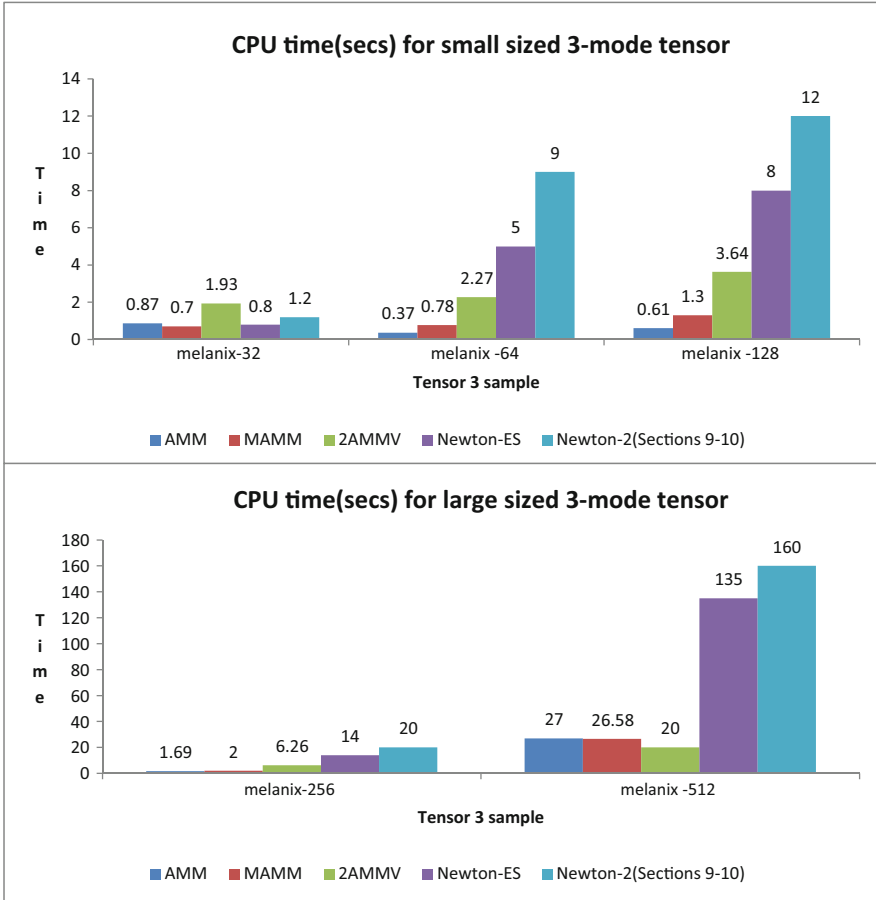


Fig. 14.3 Rank (3, 3, 3) average CPU times taken over 10 initial random guesses

Note that the probability of $2 \times 2 \times 2$ tensors, with entries normally distributed with mean 0 and variance 1, to have rank 2 is $\frac{\pi}{4}$ [2].

14.12 Conclusions

We have extended the alternating maximization method (AMM) and modified alternating maximization method (MAMM) given in [15] for the computation of best rank one approximation to best \mathbf{r} -approximations. We have also presented new algorithms such as 2-alternating maximization method variant (2AMMV) and Newton method for best \mathbf{r} -approximation (Newton-2). We have provided closed form solutions for computing the DF matrix in Newton-2. We implemented Newton-1

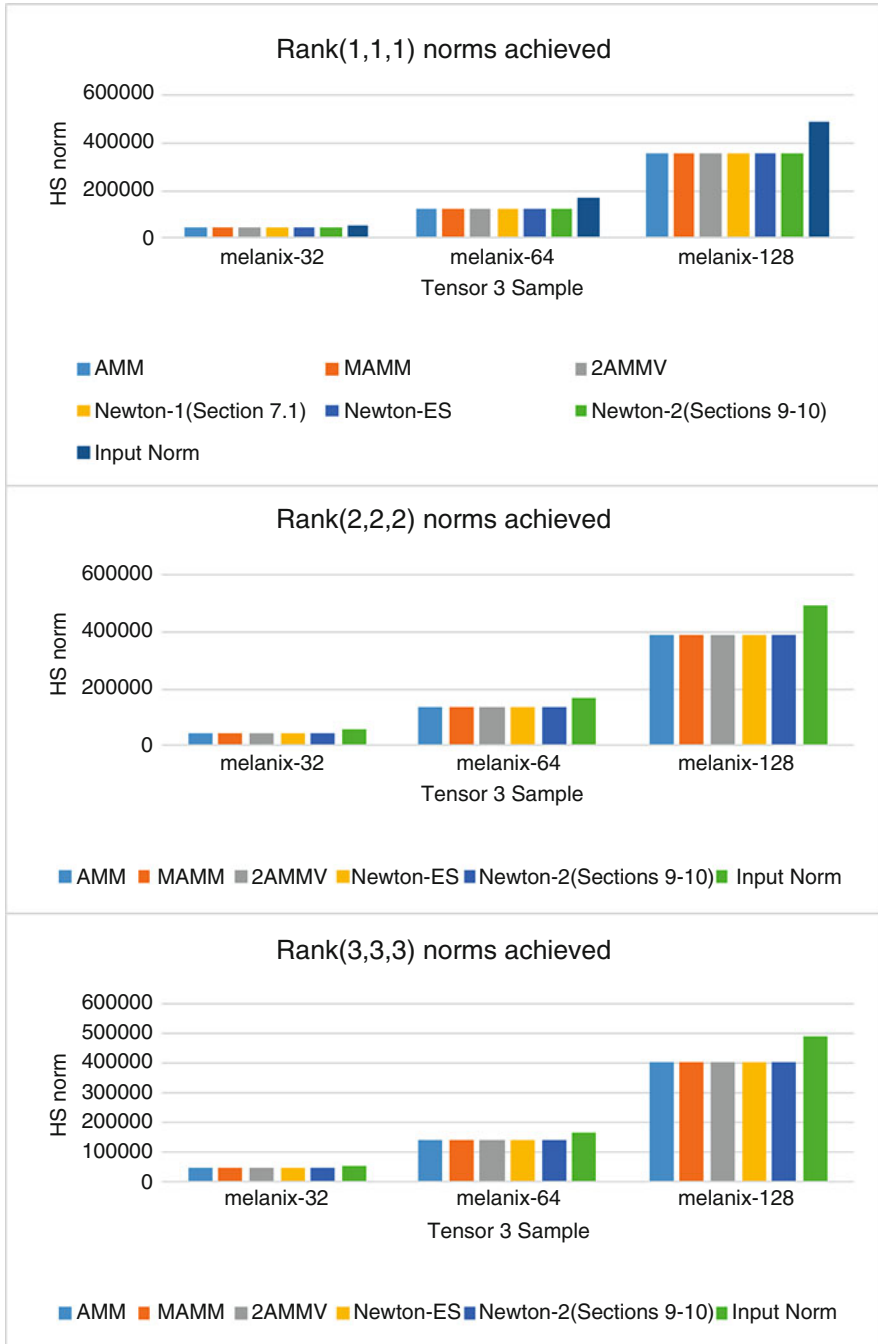


Fig. 14.4 Average Hilbert-Schmidt (HS) norms per algorithm and per data set taken over 10 different initial random guesses for small and medium size tensors

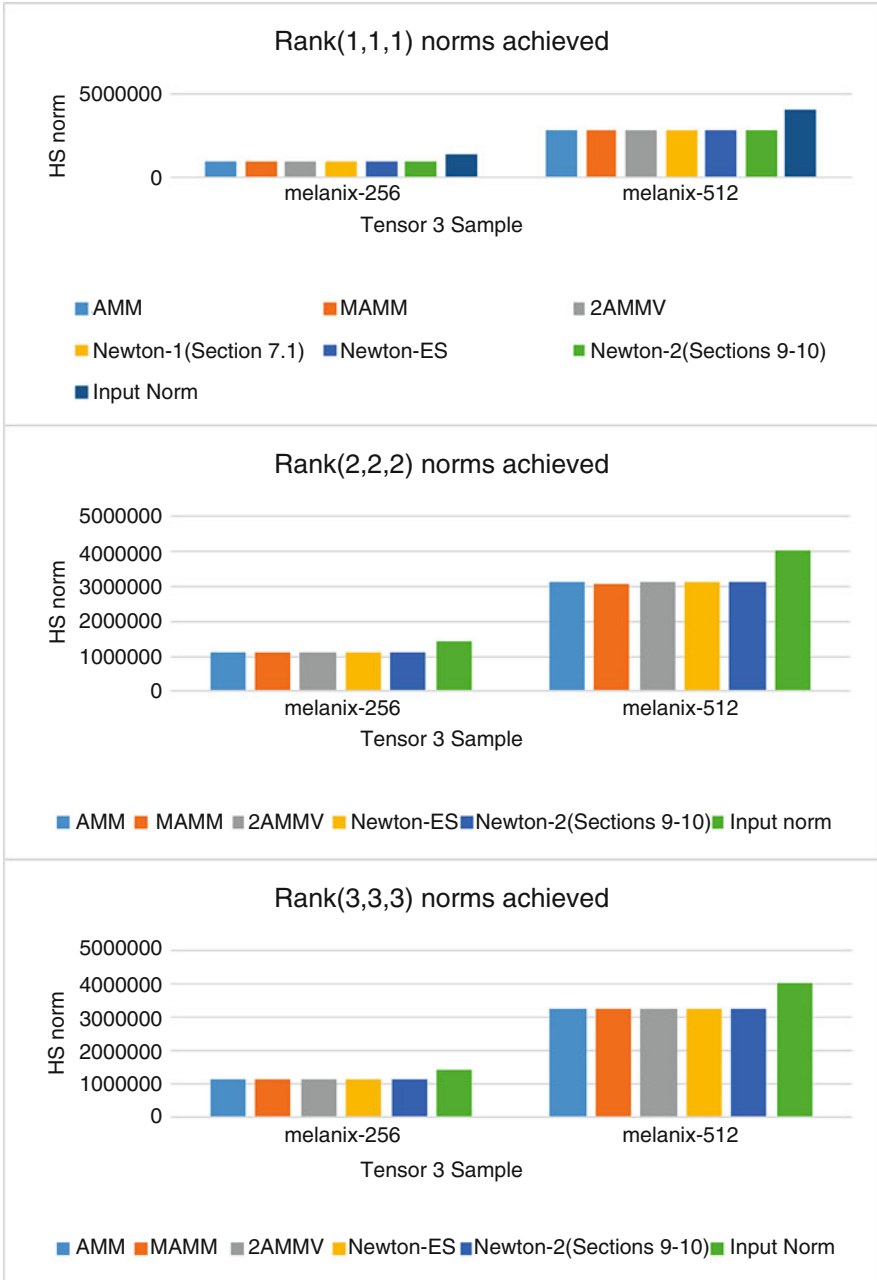


Fig. 14.5 Average Hilbert-Schmidt (HS) norms per algorithm and per data set taken over 10 different initial random guesses for large size tensors

for best rank one approximation [14] and Newton-2. From the simulations, we have found out that for rank one approximation of both large and small sized tensors, Newton-1 performed the best. For higher rank approximation, the best performers were 2AMMV in case of large size tensors and AMM or MAMM in case of small size tensors.

Acknowledgements We thank Daniel Kressner for his remarks.

References

1. Achlioptas, D., McSherry, F.: Fast computation of low rank approximations. In: Proceedings of the 33rd Annual Symposium on Theory of Computing, Heraklion, pp. 1–18 (2001)
2. Bergqvist, G.: Exact probabilities for typical ranks of $2 \times 2 \times 2$ and $3 \times 3 \times 2$ tensors. *Linear Algebra Appl.* **438**, 663–667 (2013)
3. de Lathauwer, L., de Moor, B., Vandewalle, J.: A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.* **21**, 1253–1278 (2000)
4. de Lathauwer, L., de Moor, B., Vandewalle, J.: On the best rank-1 and rank- (R_1, R_2, \dots, R_N) approximation of higher-order tensors. *SIAM J. Matrix Anal. Appl.* **21**, 1324–1342 (2000)
5. Deshpande, A., Vempala, S.: Adaptive sampling and fast low-rank matrix approximation, Electronic Colloquium on Computational Complexity, Report No. 42 pp. 1–11 (2006)
6. de Silva, V., Lim, L.-H.: Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM J. Matrix Anal. Appl.* **30**, 1084–1127 (2008)
7. Drineas, P., Kannan, R., Mahoney, M.W.: Fast Monte Carlo algorithms for matrices I–III: computing a compressed approximate matrix decomposition. *SIAM J. Comput.* **36**, 132–206 (2006)
8. Eldén, L., Savas, B.: A Newton-Grassmann method for computing the best multilinear rank- (r_1, r_2, r_3) approximation of a tensor. *SIAM J. Matrix Anal. Appl.* **31**, 248–271 (2009)
9. Friedland, S.: Nonnegative definite hermitian matrices with increasing principal minors. *Spec. Matrices* **1**, 1–2 (2013)
10. Friedland, S.: MATRICES, a book draft in preparation. <http://homepages.math.uic.edu/~friedlan/bookm.pdf>, to be published by World Scientific
11. Friedland, S., Kaveh, M., Niknejad, A., Zare, H.: Fast Monte-Carlo low rank approximations for matrices. In: Proceedings of IEEE Conference SoSE, Los Angeles, pp. 218–223 (2006)
12. Friedland, S., Mehrmann, V.: Best subspace tensor approximations, arXiv:0805.4220v1
13. Friedland, S., Mehrmann, V., Miedlar, A., Nkengla, M.: Fast low rank approximations of matrices and tensors. *J. Electron. Linear Algebra* **22**, 1031–1048 (2011)
14. Friedland, S., Mehrmann, V., Pajarola, R., Suter, S.K.: On best rank one approximation of tensors. <http://arxiv.org/pdf/1112.5914v1.pdf>
15. Friedland, S., Mehrmann, V., Pajarola, R., Suter, S.K.: On best rank one approximation of tensors. *Numer. Linear Algebra Appl.* **20**, 942–955 (2013)
16. Friedland, S., Ottaviani, G.: The number of singular vector tuples and uniqueness of best rank one approximation of tensors. *Found. Comput. Math.* **14**, 1209–1242 (2014)
17. Friedland, S., Torokhti, A.: Generalized rank-constrained matrix approximations. *SIAM J. Matrix Anal. Appl.* **29**, 656–659 (2007)
18. Frieze, A., Kannan, R., Vempala, S.: Fast Monte-Carlo algorithms for finding low-rank approximations. *J. ACM* **51**, 1025–1041 (2004)
19. Golub, G.H., Van Loan, C.F.: *Matrix Computation*, 3rd edn. John Hopkins University Press, Baltimore (1996)

20. Goreinov, S.A., Oseledets, I.V., Savostyanov, D.V., Tyrtshnikov, E.E., Zamarashkin, N.L.: How to find a good submatrix. In: *Matrix Methods: Theory, Algorithms and Applications*, pp. 247–256. World Scientific Publishing, Hackensack (2010)
21. Goreinov, S.A., Tyrtshnikov, E.E., Zamarashkin, N.L.: A theory of pseudo-skeleton approximations of matrices. *Linear Algebra Appl.* **261**, 1–21 (1997)
22. Goreinov, S.A., Tyrtshnikov, E.E.: The maximum-volume concept in approximation by low-rank matrices. *Contemp. Math.* **280**, 47–51 (2001)
23. Grasedyck, L.: Hierarchical singular value decomposition of tensors. *SIAM J. Matrix Anal. Appl.* **31**, 2029–2054 (2010)
24. Grasedyck, L., Kressner, D., Tobler, C.: A literature survey of low-rank tensor approximation techniques. *GAMM-Mitteilungen* **36**, 53–78 (2013)
25. Hackbusch, W.: Tensorisation of vectors and their efficient convolution. *Numer. Math.* **119**, 465–488 (2011)
26. Hackbusch, W.: *Tensor Spaces and Numerical Tensor Calculus*. Springer, Heilderberg (2012)
27. Hillar, C.J., Lim, L.-H.: Most tensor problems are NP-hard. *J. ACM* **60**, Art. 45, 39 pp. (2013)
28. Horn, R.A., Johnson, C.R.: *Matrix Analysis*. Cambridge University Press, Cambridge/New York (1988)
29. Khoromskij, B.N.: *Methods of Tensor Approximation for Multidimensional Operators and Functions with Applications*, Lecture at the workshop, Schnelle Löser für partielle Differentialgleichungen, Oberwolfach, 18.-23.05 (2008)
30. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. *SIAM Rev.* **51**, 455–500 (2009)
31. Lehoucq, R.B., Sorensen, D.C., Yang, C.: *ARPACK User's Guide: Solution of Large-Scale Eigenvalue Problems With Implicitly Restarted Arnoldi Methods (Software, Environments, Tools)*. SIAM, Philadelphia (1998)
32. Lim, L.-H.: Singular values and eigenvalues of tensors: a variational approach. In: *Proceedings of IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP '05)*, Puerto Vallarta, vol. 1, pp. 129–132 (2005)
33. Mahoney, M.W., Maggioni, M., Drineas, P.: Tensor-CUR decompositions for tensor-based data. In: *Proceedings of the 12th Annual ACM SIGKDD Conference*, Philadelphia, pp. 327–336 (2006)
34. Oseledets, I.V.: On a new tensor decomposition. *Dokl. Math.* **80**, 495–496 (2009)
35. Oseledets, I.V.: Tensor-Train decompositions. *SIAM J. Sci. Comput.* **33**, 2295–2317 (2011)
36. Oseledets, I.V., Tyrtshnikov, E.E.: Breaking the curse of dimensionality, or how to use SVD in many dimensions. *SIAM J. Sci. Comput.* **31**, 3744–3759 (2009)
37. Rudelson, M., Vershynin, R.: Sampling from large matrices: an approach through geometric functional analysis. *J. ACM* **54**, Art. 21, 19 pp. (2007)
38. Sarlos, T.: Improved approximation algorithms for large matrices via random projections. In: *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, Berkeley, pp. 143–152 (2006)
39. Savas, B., Lim, L.-H.: Quasi-Newton methods on Grassmannians and multilinear approximations of tensors. *SIAM J. Sci. Comput.* **32**, 3352–3393 (2010)
40. Stewart, G.W.: On the early history of the singular value decomposition. *SIAM Rev.* **35**, 551–566 (1993)
41. ten Berge, J.M.F., Kiers, H.A.L.: Simplicity of core arrays in three-way principal component analysis and the typical rank of $p \times q \times 2$ arrays. *Linear Algebra Appl.* **294**, 169–179 (1999)
42. Tucker, L.R.: Some mathematical notes on three-mode factor analysis. *Psychometrika* **31**, 279–311 (1966)
43. Zhang, T., Golub, G.H.: Rank-one approximation to high order tensors. *SIAM J. Matrix Anal. Appl.* **23**, 534–550 (2001)