GLOBAL RANK-1 APPROXIMATION FOR ORDER-3 TENSORS

MOODY T. CHU*

Abstract. Finding the absolute best rank-1 approximation for an order-3 tensor is an important first step as critical as the truncated singular value decomposition for a matrix. This note reports empirical results of two investigations. First, the feasibility of employing existent global optimization package to tackle the absolute best approximation problem is studied. Measuring the computational complexity in terms of CPU time, it is found that for tensors in $\mathbb{R}^{s\times s\times s}$ with $s \leq 100$, i.e., tensors with up to one million entries, the overhead is approximately $O(s^2)$. Second, taking advantage of its quadratic convergence, this note also experiments a modified Newton method with many starting points. It is found that the rank-1 approximation problem can easily have many local solutions. Though it has been generally recognized that most of the lower rank approximation algorithms available in the literature can find a local solution only, this experiment raises the flag that these methods might have severely missed the target.

Key words. rank-1 approximation, order-3 tensors, global optimization, projected gradient, inexact Newton method

AMS subject classifications. 65F10, 15A24, 65H10, 15A72, 58D19

1. Introduction. In addition to finding its own applications in areas such as chemometrics [18, 20, 30, 31], arithmetic complexity analysis [24], image processing, [29], principal component analysis [19, 23], psychometrics [25, 33] and other fields, the class of real-valued order-3 tensors

$$T = [\tau_{\ell,j,k}] \in \mathbb{R}^{I_1\times I_2\times I_3}, \tag{1.1}$$

has been of particular interest to researchers because it serves as the watershed that separates what we already know about the classical matrix theory and what we might not acquainted with for high dimensional arrays [24]. Some properties of matrices, such as representing linear transformations, can be generalized in an obvious way to tensors, while others, such as ranks or eigenvalues, are fundamentally different, despite of their resemblance in names. Likewise, some techniques well developed for matrices, such as the low rank approximation, cannot be immediately extended to tensors. This paper revisits the classical problem of rank-1 approximation of an order-3 tensor with some new numerical insights.

Given a real-valued order-3 tensor T, we are interested in finding unit vectors $\mathbf{u}^{(d)} \in \mathbb{R}^{I_d}$, $d = 1, \ldots 3$, and a scalar $\lambda \in \mathbb{R}$ such that the function

$$f(\lambda, \mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)}) := \|T - \lambda \mathbf{u}^{(1)} \circ \mathbf{u}^{(2)} \circ \mathbf{u}^{(3)}\|_F^2, \tag{1.2}$$

where $\circ$ stands for the tensor product, is "absolutely" minimized [24]. The problem is equivalent to maximizing

$$\lambda = \lambda(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)}) := \mathrm{vec}(T)^\top (\mathbf{u}^{(3)} \otimes \mathbf{u}^{(2)} \otimes \mathbf{u}^{(1)}), \tag{1.3}$$

subject to $\mathbf{u}^{(j)} \in S^{I_d}$, the unit sphere in $\mathbb{R}^{I_d}$, $d = 1, 2, 3$, with $\otimes$ standing for the Kronecker product. The challenge is at finding the "absolute" best approximation. Finding the absolute best rank-1 approximation for an order-3 tensor is as important as the truncated singular value decomposition (TSVD) for a matrix [6, 21]. The difficulty is that there is no direct generalization of TSVD from order-2 tensors to order-3 tensors. Solving (1.2) is a basic building block with many potential applications. We outline two scenarios below to demonstrate why this building block might be useful.

First, recall that the outer product $P = F \circ G$ between an order-m tensor $F \in \mathbb{R}^{I_1\times\ldots\times I_m}$ and an order-n tensor $G \in \mathbb{R}^{J_1\times\ldots\times J_n}$ is an order-$(m+n)$ tensor defined by

$$p_{i_1,\ldots,i_m,j_1,\ldots,j_n} := f_{i_1,\ldots,i_m} g_{j_1,\ldots,j_n}. \tag{1.4}$$

Regarding (1.4) as a type of tensor factorization of P, we ask the inverse question. That is, given an order-$(m+n)$ tensor $P \in \mathbb{R}^{I_1\times\ldots\times I_m\times J_1\times\ldots\times J_n}$, when can P be factorized as the tensor product of two lower order

*Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205. (chu@math.ncsu.edu) This research was supported in part by the National Science Foundation under grant DMS-1316779.

tensors F and G of fixed m and n? There is no easy answer. If the factorization is not possible, we then ask for its best possible approximation in the sense of finding the global solution to the problem

$$\min_{F\in\mathbb{R}^{I_1\times\ldots\times I_m},G\in\mathbb{R}^{J_1\times\ldots\times J_n}} \|P - F\circ G\|_F. \tag{1.5}$$

For matrices, this is the classical problem of the best rank-1 approximation. The answer is known precisely by the Eckart-Young-Mirsky theorem. For tensors, this is an interesting nonlinear approximation with fixed facets, which can be solved via proper matricization.

More specifically, an order-k tensor $T \in \mathbb{R}^{I_1\times I_2\times\ldots\times I_k}$ can be recognized as a linear array where the entry $\tau_{i_1,\ldots,i_k}$ of T is saved at the location

$$(i_k - 1)I_{k-1}I_{k-2}\ldots I_1 + (i_{k-1} - 1)I_{k-2}\ldots I_1 + \ldots + (i_2 - 1)I_1 + i_1. \tag{1.6}$$

Let $\mathrm{c}(F) := \prod_{i=1}^m I_i$ and $\mathrm{c}(G) := \prod_{j=1}^n J_j$ denote the total cardinalities of elements expected in F and G, respectively. We matricize the given P into a matrix $\mathbb{P} \in \mathbb{R}^{\mathrm{c}(F)\times\mathrm{c}(G)}$ whose vectorization is precisely the same linear array of P. Let $(s_i, \mathbf{f}_i, \mathbf{g}_i) \in \mathbb{R}\times\mathbb{R}^{\mathrm{c}(F)}\times\mathbb{R}^{\mathrm{c}(G)}$ denote the triplets of the left and right singular vectors of $\mathbb{P}$ corresponding to the ith largest singular value s_i, $i = 1,\ldots,\ell$, of $\mathbb{P}$. The following result, which is more general than (1.5), can be proved by using the Eckart-Young-Mirsky theorem [36].

LEMMA 1.1. *Suppose $\ell \geq 1$ is fixed. Define F_i and G_i by reshaping the columns $s_i\mathbf{f}_i$ and $\mathbf{g}_i$ (or, any combination of $a_i\mathbf{f}_i$ and $b_i\mathbf{g}_i$ with $a_ib_i = s_i$) into an order-m tensor in $\mathbb{R}^{I_1\times\ldots\times I_m}$ and an order-n tensor in $\mathbb{R}^{J_1\times\ldots\times J_n}$, respectively. Then*

$$\|P - \sum_{i=1}^{\ell} F_i\circ G_i\|_F. \tag{1.7}$$

is minimized among all possible $F_i \in \mathbb{R}^{I_1\times\ldots\times I_m}$, $G_i \in \mathbb{R}^{J_1\times\ldots\times J_n}$.

Each term in the approximation in (1.7) involves only two factors of lower order tensors, so the matrix technique is still applicable. Had it been possible to continue this procedure to three or more factors, we would have resolved the general low rank tensor approximation problem. It was pointed out that the Eckart-Young-Mirsky theorem could not be generalized to tensors [21]. We must develop some new approach for three factors. The case of order 3 considered in this paper is the very first step toward that direction.

Second, in contrast to the conventional approach by the so called alternating least squares (ALS) method that works to adjust one factor a time for the best rank-1 approximation of a generic tensor, it has been suggested recently that the SVD-based algorithms improving two factors simultaneously might have a better limiting behavior leading to better approximations [14, 16, 17, 38]. A randomized SVD updating scheme is typified in Algorithm 1, where $\times_d$ denotes the standard mode-d multiplication of a tensor with a vector [2]. The principal mechanism of the algorithm is the svds applied to the intermediate matrix C_t defined at Line 7 to obtain its globally best rank-1 approximation. Convergence theory of the algorithm has been established and numerical experiments do suggest that the SVD-based methods have the advantage of saving the computational time on large scale problems. The very same idea can be generalized to improve three or more factors simultaneously. For instance, replacing $\boldsymbol{\beta}_t$ at Line 6 by a triad of three random indices, then the product C_t at Line 7 becomes an order-3 tensor. Thus the svds should be replaced by a mechanism that finds the globally best rank-1 approximation of C_t, which is exactly the problem we are dealing with in this paper.

The conventional alternating least squares (ALS) methods [4, 9, 26], as well as most nonlinear programming techniques [34], can find only one local best rank-1 approximation to T a time. How often does this happen and how far are we off from the true best solution? In contrast to many previous studies in the literature [1, 5, 7, 13, 22, 32] for low rank approximation of general tensors by optimization techniques, we employ the notion of global optimization to search for the absolute best rank-1 approximation for order-3 tensors. This note investigates how to carry out the global optimization effectively and makes some overhead comparison. As a by-product, this investigation also implements an inexact Newton method and uncovers surprisingly many

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

Require: An order-k generic tensor T and k starting unit vectors $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(k)} \in \mathbb{R}^n$
Ensure: A local best rank-1 approximation to T

$t \leftarrow 0$
$\lambda_0 \leftarrow \langle T, \bigotimes_{\ell=1}^{k} \mathbf{u}^{(\ell)} \rangle$
repeat
  $t \leftarrow t + 1$
  $\sigma \leftarrow$ random permutation of $\{1, \ldots, k\}$
  $\boldsymbol{\beta}_t \leftarrow (\sigma_{k-1}, \sigma_k)$
  $C_t \leftarrow T \times_{\sigma_1} \mathbf{u}^{(\sigma_1)} \times_{\sigma_2} \cdots \times_{\sigma_{k-2}} \mathbf{u}^{(\sigma_{k-2})}$
  $[\mathbf{u}_t, s_t, \mathbf{v}_t] = \mathsf{svds}(C_t, 1)$ {Dominant singular value triplet via Matlab routine svds, assume uniqueness}
  if $(\mathbf{u}_t)_1 < 0$ **then**
    $\mathbf{u} = -\mathbf{u}_t, \mathbf{v} = -\mathbf{v}_t$ {Assume the general case that $(\mathbf{u}_t)_1 \neq 0$; otherwise, use another entry}
  end if
  $\lambda_t \leftarrow s_t$
  $\mathbf{u}^{(\sigma_{k-1})} \leftarrow \mathbf{u}_t, \mathbf{u}^{(\sigma_k)} \leftarrow \mathbf{v}_t$
until λ_t meets convergence criteria

local solutions, strongly implying that most locally convergent methods will be inadvertently trapped at local solutions which might be far off the absolute best approximation.

This paper is organized as follows. In Section 2 we begin with some basic information. In Section 3 we explain how the global optimization is set up and experiment on its application to randomly generated order-3 tensors with variable sizes. The information of overhead is gathered in terms of CPU time and is analyzed via regression models. In Section 4 we introduce a Newton iterative scheme that uses only the projected gradient and projected Hessian information. It is proved that, despite of the truncation by projection, the method converges quadratically. The rapid speed of convergence make it possible for us to perform large amount of repeated experiments with random starting points. It is discovered that even for a $5 \times 5 \times 5$ target tensor, there are hundreds of local solutions for the rank-1 approximation. In theory, the entire idea can generalized to higher order tensors, but the computation will be exceedingly more expensive.

2. Basic preparation. Throughout the discussion, we shall work with (1.3) as the objective function which is a cubic polynomial in the variables. Obviously, if the triplets $(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)})$ form a maximizer, then so do the triplets $(-\mathbf{u}^{(1)}, -\mathbf{u}^{(2)}, \mathbf{u}^{(3)})$, $(-\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, -\mathbf{u}^{(3)})$ and $(\mathbf{u}^{(1)}, -\mathbf{u}^{(2)}, -\mathbf{u}^{(3)})$, all of which lead to the same objective value. So the number of solutions in general should be a multiple of four. It suffices to limit our search to the case where $\mathbf{u}^{(1)}$ resides in the "northern" hemisphere of S^{I_1} by fixed the sign of its last entry. We use the Global Optimization Toolbox available in Matlab as the platform for computation.

Many solvers offer for convenience to estimate the gradients via finite differences and use an approximate Hessian which can be far from the true Hessian. The effect is the increase of overhead such as the CPU time, because more iterations and many more function evaluations are needed for convergence. Providing gradient or Hessian information of the objective or constraint functions can yield a solution in fewer iterations with more accuracy. This is especially useful since most global optimization techniques require to test out sufficiently many basins by solving the optimization problem repeatedly with different starting points. For our rank-1 approximation problem, we can supply both gradient and Hessian information in analytic form. This section explains how these particulars can be accomplished.

2.1. Projected gradient. The first-order optimality condition for a equality constrained optimization problem is that the projected gradient should be zero. However, it is not always possible to have a projected gradient in the analytic form, so general-purpose optimization approaches do not calculate the projected gradi-

ent explicitly. Instead, the theory of Lagrange multiplier is mathematically equivalent to having zero projected gradient [8]. Most codes estimate the multipliers as part of the calculation and adjust the iteration by taking into account the second order derivative information. For our rank-1 approximation problem, the following derivative information is readily available.

LEMMA 2.1. *Let $\times_d$ denote the mode-d vector product [2]. Then the gradient $\nabla\lambda$ of (1.3) is given by*

$$\nabla\lambda(\mathbf{u}^{(1)},\mathbf{u}^{(2)},\mathbf{u}^{(3)}) = \begin{bmatrix} T\times_2\mathbf{u}^{(2)}\times_3\mathbf{u}^{(3)} \\ T\times_1\mathbf{u}^{(1)}\times_3\mathbf{u}^{(3)} \\ T\times_1\mathbf{u}^{(1)}\times_2\mathbf{u}^{(2)} \end{bmatrix}, \tag{2.1}$$

where $\times_d$ denotes the mode-d tensor-vector multiplication.

In MATLAB we can use the commands reshape and shiftdim to conveniently carry out the above calculation. The partial gradient $\frac{\partial\lambda}{\partial\mathbf{u}^{(1)}} \in \mathbb{R}^{I_1}$, for example, can be obtained literally by a one-line command:

```
(reshape(u2*u3',I2*I3,1)'*reshape(shiftdim(T,1),I2*I3,[]))';
```

Using the product topology, the projected gradient is obtained by projecting each partial gradient $\frac{\partial\lambda}{\partial\mathbf{u}^{(d)}}$ onto the the tangent space of the corresponding unit sphere $S^{I_d} \subset \mathbb{R}^{I_d}$.

LEMMA 2.2. *The projected gradient of the objective function λ onto the product $S^{I_1}\times S^{I_2}\times S^{I_3}$ of unit spheres is given by*

$$g(\mathbf{u}^{(1)},\mathbf{u}^{(2)},\mathbf{u}^{(3)}) = \begin{bmatrix} T\times_2\mathbf{u}^{(2)}\times_3\mathbf{u}^{(3)} - \lambda(\mathbf{u}^{(1)},\mathbf{u}^{(2)},\mathbf{u}^{(3)})\mathbf{u}^{(1)} \\ T\times_1\mathbf{u}^{(1)}\times_3\mathbf{u}^{(3)} - \lambda(\mathbf{u}^{(1)},\mathbf{u}^{(2)},\mathbf{u}^{(3)})\mathbf{u}^{(2)} \\ T\times_1\mathbf{u}^{(1)}\times_2\mathbf{u}^{(2)} - \lambda(\mathbf{u}^{(1)},\mathbf{u}^{(2)},\mathbf{u}^{(3)})\mathbf{u}^{(3)} \end{bmatrix}. \tag{2.2}$$

A typical way to solve $g(\mathbf{u}^{(1)},\mathbf{u}^{(2)},\mathbf{u}^{(3)}) = 0$ is the so called the so called alternating least squares (ALS) approach. The basic scheme is that, given $\mathbf{u}^{(2)}_{[p]}$ and $\mathbf{u}^{(3)}_{[p]}$, we update $\mathbf{u}^{(1)}_{[p+1]}$ via

$$(\text{The ALS method}) \begin{cases} \lambda^{(1)}_{[p]} & := & \|T\times_2\mathbf{u}^{(2)}_{[p]}\times_3\mathbf{u}^{(3)}_{[p]}\|_2, \\ \mathbf{u}^{(1)}_{[p+1]} & := & \frac{T\times_2\mathbf{u}^{(2)}_{[p]}\times_3\mathbf{u}^{(3)}_{[p]}}{\lambda^{(1)}_{[p]}}, \end{cases} \tag{2.3}$$

continue on in a similar way to update $\mathbf{u}^{(2)}_{[p+1]}$ and $\mathbf{u}^{(3)}_{[p+1]}$ in turn, and repeat the cycles. It can be shown that the sequence $\{(\mathbf{u}^{(1)}_{[p]},\mathbf{u}^{(2)}_{[p]},\mathbf{u}^{(3)}_{[p]})\}$ converges [35, 37]. Being just a first order method, the ALS iteration converges only linearly.

It is easy to see that (2.2) is precisely the derivative of the classical Lagrangian with all multipliers equal to $\lambda(\mathbf{u}^{(1)},\mathbf{u}^{(2)},\mathbf{u}^{(3)})$. We certainly can go beyond the conventional ALS method with much more advanced and efficient techniques to find the critical points. We could employ, for example, a Newton-type iteration with the availability of the analytic projected Hessian or any proven optimization software, e.g., those available in [15, 28]. Using these techniques has the danger of finding merely a critical point or a local solution, which may not be the best rank-1 approximation. To achieve our goal, some extra efforts must be taken.

2.2. Projected Hessian. We can calculate the projected Hessian without presupposing the Lagrange multipliers. The justification that the following two steps are equivalent to the action of the projected Hessian can be found in [8].

Step 1. Extend $g(\mathbf{u}^{(1)},\mathbf{u}^{(2)},\mathbf{u}^{(3)})$ formally to the entire space $\mathbb{R}^{I_1}\times\mathbb{R}^{I_2}\times\mathbb{R}^{I_3}$ and calculate its Fréchet derivative. Because we know the projected gradient analytically, the extension is easy. Using the same

notation for the extended function, the action of $g'(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)})$ on an arbitrary $(\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)}) \in \mathbb{R}^{I_1} \times \mathbb{R}^{I_2} \times \mathbb{R}^{I_3}$ is given by the vector

$$g'(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)}).(\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)}) =$$
$$\begin{bmatrix} -\frac{\partial \lambda}{\partial \mathbf{u}^{(1)}}.\mathbf{h}^{(1)}\mathbf{u}^{(1)} - \lambda \mathbf{h}^{(1)} + T \times_2 \mathbf{h}^{(2)} \times_3 \mathbf{u}^{(3)} - \frac{\partial \lambda}{\partial \mathbf{u}^{(2)}}.\mathbf{h}^{(2)}\mathbf{u}^{(1)} + T \times_2 \mathbf{u}^{(2)} \times_3 \mathbf{h}^{(3)} - \frac{\partial \lambda}{\partial \mathbf{u}^{(3)}}.\mathbf{h}^{(3)}\mathbf{u}^{(1)} \\ T \times_1 \mathbf{h}^{(1)} \times_3 \mathbf{u}^{(3)} - \frac{\partial \lambda}{\partial \mathbf{u}^{(1)}}.\mathbf{h}^{(1)}\mathbf{u}^{(2)} - \frac{\partial \lambda}{\partial \mathbf{u}^{(2)}}.\mathbf{h}^{(2)}\mathbf{u}^{(2)} - \lambda \mathbf{h}^{(2)} + T \times_1 \mathbf{u}^{(1)} \times_3 \mathbf{h}^{(3)} - \frac{\partial \lambda}{\partial \mathbf{u}^{(3)}}.\mathbf{h}^{(3)}\mathbf{u}^{(2)} \\ T \times_1 \mathbf{h}^{(1)} \times_2 \mathbf{u}^{(2)} - \frac{\partial \lambda}{\partial \mathbf{u}^{(1)}}.\mathbf{h}^{(1)}\mathbf{u}^{(3)} + T \times_1 \mathbf{u}^{(1)} \times_2 \mathbf{h}^{(2)} - \frac{\partial \lambda}{\partial \mathbf{u}^{(2)}}.\mathbf{h}^{(2)}\mathbf{u}^{(3)} - \frac{\partial \lambda}{\partial \mathbf{u}^{(3)}}.\mathbf{h}^{(3)}\mathbf{u}^{(3)} - \lambda \mathbf{h}^{(3)} \end{bmatrix} \quad (2.4)$$

Step 2. Limit the action of $g'(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)})$ to tangent vectors. The tangent space of $S^{I_1} \times S^{I_2} \times S^{I_3}$ is the product of tangent spaces of S^{I_d}, $d = 1, 2, 3$. Therefore, apply the action only to those $(\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)})$ satisfying

$$< \mathbf{h}^{(d)}, \mathbf{u}^{(d)} >= 0, \quad d = 1, 2, 3. \quad (2.5)$$

The negative definiteness of the resulting action on the tangent space can serve as the second-order optimality condition for our problem of maximizing (1.3).

The restricted map, the projected Hessian, can be interpreted as follows.

LEMMA 2.3. *The projected Hessian of the objective function λ over the product of tangent spaces of S^{I_d}, $d = 1, 2, 3$, can be represented by the matrix*

$$H(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)}) := \begin{bmatrix} -\lambda \mathrm{I}_{I_1} & T \times_3 \mathbf{u}^{(3)} & T \times_2 \mathbf{u}^{(2)} \\ (T \times_3 \mathbf{u}^{(3)})^\top & -\lambda \mathrm{I}_{I_2} & T \times_1 \mathbf{u}^{(1)} \\ (T \times_2 \mathbf{u}^{(2)})^\top & (T \times_1 \mathbf{u}^{(1)})^\top & -\lambda \mathrm{I}_{I_3} \end{bmatrix}, \quad (2.6)$$

where I_{I_d} stands for the identity matrix in $\mathbb{R}^{I_d \times I_d}$, $d = 1, 2, 3$.

Proof. Consider the bilinear form $\langle\!\langle (\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)}), g'(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)}).(\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)}) \rangle\!\rangle$ where $\mathbf{h}^{(d)}$ is a tangent vector to the sphere S^{I_d}. Using (2.5), the bilinear form applied to the tangent vectors can be simplified to

$$\langle\!\langle (\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)}), g'(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)}).(\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)}) \rangle\!\rangle =$$
$$-\lambda(\|\mathbf{h}^{(1)}\|^2 + \|\mathbf{h}^{(2)}\|^2 + \|\mathbf{h}^{(3)}\|^2) + 2(\langle T \times_3 \mathbf{u}^{(3)}, \mathbf{h}^{(1)} \circ \mathbf{h}^{(2)} \rangle + \langle T \times_2 \mathbf{u}^{(2)}, \mathbf{h}^{(1)} \circ \mathbf{h}^{(3)} \rangle + \langle T \times_1 \mathbf{u}^{(1)}, \mathbf{h}^{(2)} \circ \mathbf{h}^{(3)} \rangle).$$

The expression (2.6) is the matrix representation of such a bilinear map. □

Note that the off-diagonal portion of $H(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)})$ is precisely the Hessian of the objective function λ, whereas the diagonal portion is that of the constraints of unit length. In other words, (2.6) is precisely the Hessian of the Lagrangian restricted to the tangent space of $S^{I_1} \times S^{I_2} \times S^{I_3}$, but we have avoided using the Lagrangian.

3. Global optimization. Finding the best rank-1 approximation for an order-3 tensor might seem analogous to finding the best rank-1 approximation for an order-2 matrix. Low rank approximation for matrices enjoys well developed theory, i.e., the Echard-Young-Mirsky theorem, and reliable algorithms, e.g., the Golub-Kahan SVD algorithm or the Lanzcos bidiagonalization process if only a few singular triplets are needed. In contrast, there is no such a theory nor an effective algorithm for the best rank-1 approximation for an order-3 tensor. The so called truncated higher-order SVD for tensors in [10] does not generalize even the concept of the TSVD for matrices at all [21]. Instead of a linear algebra approach, we propose using global optimization approach which, just like the SVD algorithm for matrices, is iterative in nature. We explain the setup and investigate the computational overhead in this section. Most importantly, we want to call to the attention that, without resorting to the global optimization mechanism, most low rank tensor approximation techniques will fail in finding the best solution.

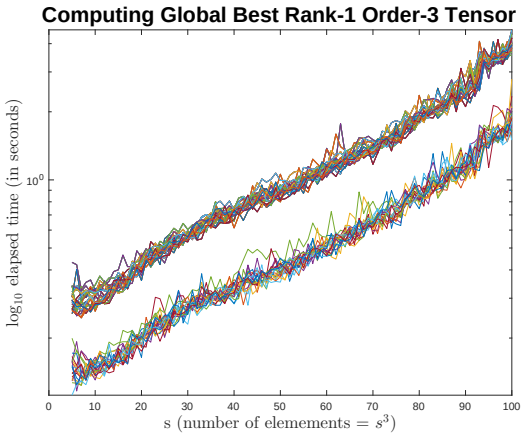

FIGURE 3.1. *Running time needed by global optimization for the best rank-1 approximation for order-3 tensors.*

3.1. Set-up. The basic idea behind global optimization is to use a local solver repeatedly on many starting points with the goal of choosing the best among the local optima found in the basins of attraction of the starting points. Because we have to compromise on the overhead with finitely many trials, there is no guarantee that a solution is a true global solution. Still, after sufficiently many trials and if the starting points are reasonably distributed throughout the bounded feasible set, the final solution has the potential of being the best one.

We propose to carry out numerical experiments as follows. We generate square test tensors of the form

$$T_s = \mathbf{a}_s \circ \mathbf{b}_s \circ \mathbf{c}_s$$

where $\mathbf{a}_s, \mathbf{b}_s, \mathbf{c}_s$ are random vectors in $\mathbb{R}^s$, $s = 4, 5, \ldots, 100$. Our idea is that if the global optimization has difficulty in recovering the exact decomposition for T_s, then there is little hope for general order-3 tensors. The interior point method is used as the local solver for the constrained optimization package fmincon. Variables $\mathbf{u}^{(d)}$, $d = 1, 2, 3$, are constrained to the unit sphere $S^{(s)}$. Analytic information of both gradient and Hessian is explicitly given. Starting points are chosen randomly from uniform distribution over the cube $[-1, 1]^s \subset \mathbb{R}^s$. For each given T_s, the global optimization solver MultiStart is called to run the procedure fmincon at 10 random starting points. Positive exit flag is checked to ascertain that the local solver does converge to a local optimum. Because the tensor T_s is exactly of rank one, the global objective value should be close to the theoretical value $\lambda = 1$, contingent upon the stopping criteria which are determined by the tolerance on the constraint violation TolCon $= 10^{-10}$, the termination tolerance on the function value TolFun $= 10^{-10}$, and the termination tolerance on iterates TolX $= 10^{-6}$.

It used to be that the most objective means for gauging the computational complexity would be counting the theoretical number of floating-point operations (flops). However, given nowadays computing technologies where machines (even a desktop PC) are equipped with high performance processors (vector or parallel, hyper-threading or multi-cores), simple flop counts are no longer valid anymore for measuring the performance. A standard computer software library, such as Matlab or LAPACK, is often highly optimized, which also muddles the flop counts. Furthermore, given the high speed of today's CPU, the memory latency to fetch anything not in cache is much greater than the cost of a flop. For these reasons, Matlab has removed one of their most famous commands flops since Version 6. In order to compare the overhead, the elapsed time taken by MultiStart to finish the task, i.e., running through 10 random starting points to find the best solution, is recorded per given T_s. Depending on the loading of the CPU, the time measurement might fluctuate. So, for each of the dimension $s = 4, 5, \ldots, 100$, we repeat the experiment 20 times with randomly generated test tensors T_s and measure the individual CPU time. We understand that the face value of time measurement is machine dependent, but the trend should be generally indicative.

3.2. Overhead estimate. Plotted in the top streak of Figure 3.1 are the logarithmic graphs of running time needed by MultiStart to complete 10 random starting points to compute the global best rank-1 approximation

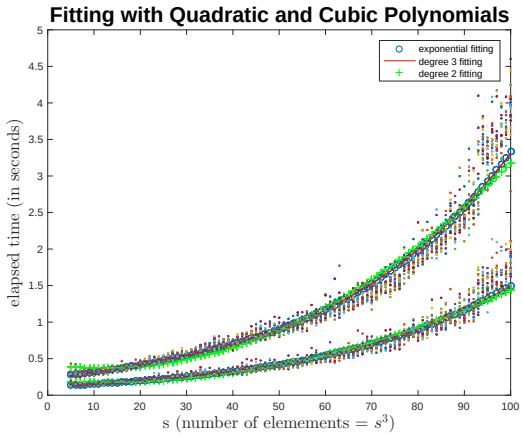

FIGURE 3.2. *Polynomial fitting of the running time needed for the best rank-1 approximation of order-3 tensors.*

for T_s. Each color represents one collection of time across the variation of dimension s from 4 to 100. There should be 20 colors to represent the 20 runs for each s. There is no need to tell the difference among the curves. Rather, the clustering indicates that the measurement of time for the 20 runs is consistent. What is most interesting is the nearly linear relationship in the logarithmic graphs. Suppose we average the running time for each s and perform a linear regression. We obtain an approximation of the computational time

$$t_{10,exponential}(s) \approx 10^{(-6.6065\times10^{-1})+(1.1301\times10^{-2})s}. \tag{3.1}$$

The boundedness of our feasible set helps, but there is no theoretical basis that an attempt by finitely many starting points will guarantee the finding of a global solution. Out of curiosity, we redo the experiment by limiting MultiStart to the task of completing merely 5 random starting points. The results are plotted in the bottom streak of Figure 3.1, which again shows a linear correlation that is almost parallel to that of the 10-starting-point case. The computational time is given approximately by

$$t_{5,exponential}(s) \approx 10^{(-9.2521\times10^{-1})+(1.1039\times10^{-2})s}, \tag{3.2}$$

which is about one half of $t_{10,exponential}(s)$. Such a ratio is understandable because in theory the 5-starting-point case should cost half. It is rather surprising that the global solution is found in almost 99.9% of the tests even with only 5 multi-starts. This behavior, of course, has something to do with the effective restart strategy built in the package, e.g., rejecting starting points falling in the basin of attraction that has already been computed. If some statistical significance of success rate relative to the number of trials can be established, then we might save tremendous overhead by using fewer starting points.

We can further simplify the estimate of overhead by polynomial regressions. Let each time measurement be represented by a suitably colored dot. The exponential fittings by models (3.1) and (3.2) are plotted by blue circles in Figure 3.2, respectively. Suppose we perform polynomial fittings of degree 2 and 3 to the average of the running time. We obtain the models

$$t_{10,cubic}(s) = (9.0381\times10^{-06})s^3 - (9.4997\times10^{-04})s^2 + (4.3106\times10^{-02})s - (6.1435\times10^{-02}), \tag{3.3}$$
$$t_{10,quadratic}(s) = (4.7353\times10^{-04})s^2 - (1.9137\times10^{-02})s + (5.9062\times10^{-1}), \tag{3.4}$$

and

$$t_{5,cubie}(s) = (4.1153\times10^{-06})s^3 - (4.2636\times10^{-04})s^2 + (1.9023\times10^{-02})s - (1.0537\times10^{-02}), \tag{3.5}$$
$$t_{5,quadratic}(s) = (2.2180\times10^{-04})s^2 - (9.3178\times10^{-03})s + (2.8636\times10^{-01}), \tag{3.6}$$

respectively. The plots of these polynomials in Figure 3.2 seem to suggest that even the quadratic fitting provides a reasonable estimate of overhead for low dimensional tensors. In that case, notice again that

$$t_{5,quadratic}(s) \approx \frac{1}{2} t_{10,quadratic}(s),$$

which suggests that the experiment is reasonably robust. It is interesting to expand the exponential function (3.2) in its Taylor series

$$0.2184489693 + 0.005684372943\, s + 0.00007395799551\, s^2 + 0.0000006414996749\, s^3 + O(s^4)$$

The small coefficient associated with s^3 suggests again that maybe a quadratic fitting is sufficient.

In all, the scalability of time measurement by two independent trials with 5 multi-starts and 10 multi-starts across the variation of dimensions $s = 4, 5, \ldots, 100$ indicate that the data we have collected can be considered as consistent. Our experiments thus lead to a rather surprising conclusion — the CPU time needed for finding the global best rank-1 approximation for an order-3 tensor in $\mathbb{R}^{s\times s\times s}$ by the global optimization package MultiStart is of the order $O(s^2)$. Such an overhead for finding the globally best rank-1 approximation is somewhat unexpectedly low because the tensor contains s^3 many entries.

4. Newton Method. The problem of maximizing (1.3) is simple enough that maybe we can apply the Newton method to the projected gradient directly and, similar to the MultiStart, we can test out multiple initial points for the iteration to search for a global solution.

4.1. Inexact Newton Iteration. Taking into account that not only the equation $g(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)}) = 0$ is to be satisfied, but also that $(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)}) \in S^{I_1} \times S^{I_2} \times S^{I_3}$, whereas the matrix representation $H(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)})$ is applicable only to the tangent space of $S^{I_1} \times S^{I_2} \times S^{I_3}$, we propose a modified iterative scheme that utilizes only the projection Hessian as follows.

LEMMA 4.1. *Assuming that the current iterate* $(\mathbf{u}^{(1)}_{[p]}, \mathbf{u}^{(2)}_{[p]}, \mathbf{u}^{(3)}_{[p]})$ *is feasible, then the update obtained from*

$$H(\mathbf{u}^{(1)}_{[p]}, \mathbf{u}^{(2)}_{[p]}, \mathbf{u}^{(3)}_{[p]}) \begin{bmatrix} \Delta\mathbf{u}^{(1)}_{[p]} \\ \Delta\mathbf{u}^{(2)}_{[p]} \\ \Delta\mathbf{u}^{(3)}_{[p]} \end{bmatrix} = -g(\mathbf{u}^{(1)}_{[p]}, \mathbf{u}^{(2)}_{[p]}, \mathbf{u}^{(3)}_{[p]}) \tag{4.1}$$

$$\mathbf{u}^{(d)}_{[p+1]} := \frac{\mathbf{u}^{(d)}_{[p]} + \rho_{[p]} \Delta\mathbf{u}^{(d)}_{[p]}}{\|\mathbf{u}^{(d)}_{[p]} + \rho_{[p]} \Delta\mathbf{u}^{(d)}_{[p]}\|_2}, \quad d = 1, 2, 3, \tag{4.2}$$

where the step size $\rho_{[p]}$ *is selected by following the Armijo rule, converges quadratically.*

Proof. We start with the conventional Newton iteration applied to the Lagrangian of optimization problem (1.3). To save from making repeated reference to the subscript $_{[p]}$, we assume that the current iterates are denoted as $(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)}, \lambda_1, \lambda_2, \lambda_3)$. One Newton step amounts to solving the linear system

$$\begin{bmatrix} -\lambda_1 \mathrm{I}_{I_1} & T \times_3 \mathbf{u}^{(3)} & T \times_2 \mathbf{u}^{(2)} & -\mathbf{u}^{(1)} & 0 & 0 \\ (T \times_3 \mathbf{u}^{(3)})^\top & -\lambda_2 \mathrm{I}_{I_2} & T \times_1 \mathbf{u}^{(1)} & 0 & -\mathbf{u}^{(2)} & 0 \\ (T \times_2 \mathbf{u}^{(2)})^\top & (T \times_1 \mathbf{u}^{(1)})^\top & -\lambda_3 \mathrm{I}_{I_3} & 0 & 0 & -\mathbf{u}^{(3)} \\ -\mathbf{u}^{(1)\top} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\mathbf{u}^{(2)\top} & 0 & 0 & 0 & 0 \\ 0 & 0 & -\mathbf{u}^{(3)\top} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{u}^{(1)} \\ \Delta\mathbf{u}^{(2)} \\ \Delta\mathbf{u}^{(3)} \\ \Delta\lambda_1 \\ \Delta\lambda_2 \\ \Delta\lambda_3 \end{bmatrix} = - \begin{bmatrix} T \times_2 \mathbf{u}^{(2)} \times_3 \mathbf{u}^{(3)} - \lambda_1 \mathbf{u}^{(1)} \\ T \times_1 \mathbf{u}^{(1)} \times_3 \mathbf{u}^{(3)} - \lambda_2 \mathbf{u}^{(2)} \\ T \times_1 \mathbf{u}^{(1)} \times_2 \mathbf{u}^{(2)} - \lambda_3 \mathbf{u}^{(3)} \\ \frac{\|\mathbf{u}^{(1)}\|^2 - 1}{2} \\ \frac{\|\mathbf{u}^{(2)}\|^2 - 1}{2} \\ \frac{\|\mathbf{u}^{(3)}\|^2 - 1}{2} \end{bmatrix}$$

for the increments $(\Delta\mathbf{u}^{(1)}, \Delta\mathbf{u}^{(2)}, \Delta\mathbf{u}^{(3)}, \Delta\lambda_1, \Delta\lambda_2, \Delta\lambda_3)$. The last three equations together with the feasibility of $\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)}$ imply that

$$\mathbf{u}^{(d)\top}\Delta\mathbf{u}^{(d)} = 0, \quad d = 1, 2, 3.$$

Pre-multiplying the d-th equation by $\mathbf{u}^{(d)\top}$, we obtain the equalities

$$\left\{\begin{array}{rcl} (T \times_1 \mathbf{u}^{(1)} \times_3 \mathbf{u}^{(3)})^\top \Delta\mathbf{u}^{(2)} + (T \times_1 \mathbf{u}^{(1)} \times_2 \mathbf{u}^{(2)})^\top \Delta\mathbf{u}^{(3)} + (\lambda - \lambda_1) &=& \Delta\lambda_1, \\ (T \times_2 \mathbf{u}^{(2)} \times_3 \mathbf{u}^{(3)})^\top \Delta\mathbf{u}^{(1)} + (T \times_1 \mathbf{u}^{(1)} \times_2 \mathbf{u}^{(2)})^\top \Delta\mathbf{u}^{(3)} + (\lambda - \lambda_2) &=& \Delta\lambda_2, \\ (T \times_2 \mathbf{u}^{(2)} \times_3 \mathbf{u}^{(3)})^\top \Delta\mathbf{u}^{(1)} + (T \times_1 \mathbf{u}^{(1)} \times_3 \mathbf{u}^{(3)})^\top \Delta\mathbf{u}^{(2)} + (\lambda - \lambda_3) &=& \Delta\lambda_3. \end{array}\right. \tag{4.3}$$

For the Lagrangian of general problems, the Lagrange multipliers are part of the unknowns to be found. In our application, however, we know $\lambda_d = \lambda$, $d = 1, 2, 3$. The exact Newton step therefore can be written as

$$H(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)}) \begin{bmatrix} \Delta\mathbf{u}^{(1)} \\ \Delta\mathbf{u}^{(2)} \\ \Delta\mathbf{u}^{(3)} \end{bmatrix} = -g(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)}) + \mathbf{r}, \tag{4.4}$$

with

$$\mathbf{r} = \mathbf{r}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)}, \Delta\mathbf{u}^{(1)}, \Delta\mathbf{u}^{(2)}, \Delta\mathbf{u}^{(3)}) := \begin{bmatrix} \mathbf{u}^{(1)} & 0 & 0 \\ 0 & \mathbf{u}^{(2)} & 0 \\ 0 & 0 & \mathbf{u}^{(3)} \end{bmatrix} \begin{bmatrix} \Delta\lambda_1 \\ \Delta\lambda_2 \\ \Delta\lambda_3 \end{bmatrix}. \tag{4.5}$$

In other words, the scheme (4.1) which ignores the term $\mathbf{r}$ in (4.4) constitutes an inexact Newton step.

By the theory of inexact Newton method [11, 12, 27], the magnitude of the residual $\mathbf{r}$ determines the rate of local convergence. Using (4.5), together with the fact that $\mathbf{u}^{(d)}$, $d = 1, 2, 3$, is of unit length, we obtain the first estimate that

$$\|\mathbf{r}\| = O(\| \begin{bmatrix} \Delta\lambda_1 \\ \Delta\lambda_2 \\ \Delta\lambda_3 \end{bmatrix} \|) = O(\| \begin{bmatrix} \Delta\mathbf{u}^{(1)} \\ \Delta\mathbf{u}^{(2)} \\ \Delta\mathbf{u}^{(3)} \end{bmatrix} \|) = O(\|g(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)})\|), \tag{4.6}$$

which is sufficient to assert that the iterates converge linearly [11, Theorem 2.3].

Indeed, if we denote the three components of $g(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)})$ separately as $g^{(d)}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)})$, $d = 1, 2, 3$, we can further simplify $\Delta\lambda_1$ to

$$\begin{aligned} \Delta\lambda_1 &= (g^{(2)}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)}) + \lambda\mathbf{u}^{(2)})^\top \Delta\mathbf{u}^{(2)} + (g^{(3)}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)}) + \lambda\mathbf{u}^{(3)})^\top \Delta\mathbf{u}^{(3)}, \\ &= g^{(2)\top}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)})\Delta\mathbf{u}^{(2)} + g^{(3)\top}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)})\Delta\mathbf{u}^{(3)}, \end{aligned} \tag{4.7}$$

and likewise for $\Delta\lambda_2$ and $\Delta\lambda_3$. Thus we actually know that

$$\Delta\lambda_d = O(\|g(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)})\|^2), \quad d = 1, 2, 3. \tag{4.8}$$

It follows by [11, Theorem 3.3] that the convergence of the scheme (4.1) is quadratic. □

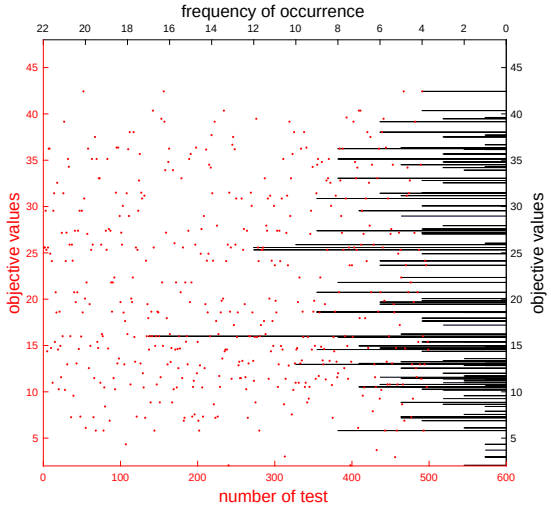

FIGURE 4.1. *An example of 117 optimal values from 500 runs (red dots) by the Newton iteration on a random tensor in* $\mathbb{R}^{5\times5\times5}$. *Histogram of optimal values is plotted in black frame.*

4.2. Multiple (many) local solutions. The Newton iteration converges fast, but is equally attracted to a local minimum and a local maximum. So, different from the global optimization, the Newton iteration alone cannot differentiate which direction the objective value λ is converging to. However, if $\lambda(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)})$ is a local minimum, then by the symmetry we see that $\lambda(-\mathbf{u}^{(1)}, -\mathbf{u}^{(2)}, -\mathbf{u}^{(3)}) = -\lambda(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)})$ is a local maximum, and vice versa. Any local minimum found by the iteration naturally provides a counter part of a local maximum by switching the sign. The Amijo rule therefore should be applied in both ways, i.e., if the condition of searching for a sufficient increase cannot be met by successful step size reductions, then we reverse to search for a sufficient decrease of the objective value.

Numerical experiments indicate that typically there are many solutions for the first order optimality condition $g(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)}) = 0$. These are critical points for the objective function λ and, with appropriate sign switches, these are the local maxima for (1.3). Demonstrated in Figure 4.1 are results from 500 runs of the Newton method applied to one randomly generated tensor of size $5 \times 5 \times 5$. The starting points are randomly selected from a uniform distribution over the unit sphere. Also plotted by horizontal bars on the right margin in Figure 4.1 is the histogram of the optimal values in this test. The frequency of occurrence is marked along the top axis from right to left. The graph is significant in twofold. First, it probably represents all possible critical values. In this particular example, there are 117 local optimal values some of which are close to each other. Recall that corresponding to each optimal value are 4 local solutions which differ by sign changes. So there are at least 468 local maximizers for this problem alone. Second, it strongly suggests that the first-order ALS method used extensively by practitioners or any locally convergent method can easily get trapped at any of the 468 local solutions, some of which occur more frequently than others, as is seen in the histogram. Because the starting points are distributed uniformly, a possible explanation for this frequency disparity is that some local solutions have larger basins of attraction. The largest objective value, attainable by only 4 successes out of 500 trials, is verified independently by a run of the global optimization mentioned in Section 3. Though it has been generally recognized that most lower rank approximation algorithms available in the literature can find a local solution only, this experiment raises the flag that these methods might have severely missed the target.

The Newton method is using the Hessian information, but it lacks the mechanism of steering the iterates to a global best rank-1 approximation. With its fast convergence, we have a chance to repeat many more multiple trials. Instead of just using brute force to perform many trials, we might improve the chance by bringing in some smart starting point strategy to avoid repeated basins of attraction. This requires a lot of fine tuning of the code, which we choose not to investigate in the preliminary note since we already have the global optimization in hand.

Also arises is the interesting problem of determining the number of solutions of the polynomial system $g(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)}) = 0$ on the $S^{I_1} \times S^{I_2} \times S^{I_3}$. This is a classical problem in the realm of real algebraic geometry. We do not think that there is an easy answer.

5. Generalization. Both ideas discussed above can readily be generalized to tensors of arbitrary order for finding the global best rank-1 approximation. In particular, the analytic forms of the gradient and the Hessian can be generalized mechanically from (2.2) and (2.6). The main concern, however, is the prevalent curse of dimensionality that will substantially increase the computational overhead.

For instance, to formulate a component such as

$$T \times_{\alpha_1} \mathbf{u}^{(\alpha_1)} \times_{\alpha_2} \mathbf{u}^{(\alpha_2)} \ldots \times_{\alpha_s} \mathbf{u}^{(\alpha_s)}$$

for an order-k square tensor over $\mathbb{R}^{I_1 \times \ldots I_k}$, where $s \leq k$, an efficient nested multiplication without relying on memory optimization requires [3]

$$\prod_{\ell=1}^{k} I_\ell + \prod_{\ell \neq \alpha_1} I_\ell + \prod_{\ell \neq \alpha_1, \alpha_2} I_\ell + \ldots + \prod_{\ell \neq \alpha_1, \alpha_2, \ldots \alpha_{s-1}} I_\ell$$

many entry-to-entry multiplications. In forming the projected gradient and the projected Hessian, the lengths s are $k-1$ and $k-2$, respectively, and there are k and $\frac{k(k-1)}{2}$ such components with varying $(\alpha_1, \ldots, \alpha_s)$. These counts reflect the cost needed for forming the derivatives only, which will be computed repeatedly many times during the optimization process. We need not calculate the total explicitly, but it should be clear that the computational cost will be high when k is large. Additionally, other internal tasks, such as solving linear system, convergence assessment, or even bookkeeping relevant to the optimization package, are also size dependent. While in theory the cost for high order tensors is scalable, it might be too expensive to be practical. If for problems of certain sizes the cost is tolerable, then the absolute best approximation might be used to bench the performance of other local methods.

6. Conclusion. The class of order-3 tensors might seem too limited to be of interest when comparing with general tensors. However, the class of order-2 tensors, namely, matrices, is even more special and has already proved critical in almost every branch of sciences. Just like matrices, order-3 tensors have their own roles in applications. Additionally, order-3 tensors serves as the gateway to and the testing ground for understanding higher order tensors.

In this paper, we investigate the feasibility of employing the global optimization techniques to solve the absolute best rank-1 approximation problem for order-3 tensors. By furnishing the analytic projected gradient and projected Hessian information, we conclude that finding the absolute best solution can be achieved at reasonable computational cost.

We also experiment with an inexact Newton iteration. With its quadratic rate of convergence we are able to uncover many local solutions, so much so that most locally convergent methods probably are inadvertently trapped at local solutions which might be far off the absolute best approximation.

REFERENCES

[1] E. ACAR, D. M. DUNLAVY, AND T. G. KOLDA, *A scalable optimization approach for fitting canonical tensor decompositions*, Journal of Chemometrics, 25 (2011), pp. 67–86.

[2] B. W. BADER AND T. G. KOLDA, *Algorithm 862: MATLAB tensor classes for fast algorithm prototyping*, ACM Trans. Math. Software, 32 (2006), pp. 635–653.

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

[4] G. BEYLKIN AND M. J. MOHLENKAMP, *Algorithms for numerical analysis in high dimensions*, SIAM J. Sci. Comput., 26 (2005), pp. 2133–2159.

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

[6] J. CARROLL AND J.-J. CHANG, *Analysis of individual differences in multidimensional scaling via an n-way generalization of Eckart-Young decomposition*, Psychometrika, 35 (1970), pp. 283–319.

[7] B. CHEN, Z. LI, AND S. ZHANG, *On optimal low rank Tucker approximation for tensors: the case for an adjustable core size*, J. Global Optim., 62 (2015), pp. 811–832.

[8] M. T. CHU AND K. R. DRIESSEL, *The projected gradient method for least squares matrix approximations with spectral constraints*, SIAM J. Numer. Anal., 27 (1990), pp. 1050–1060.

[9] P. COMON, X. LUCIANI, AND A. L. F. DE ALMEIDA, *Tensor decompositions, alternating least squares and other tales*, J. Chemometrics, 23 (2009), pp. 393–405.

[10] L. DE LATHAUWER, B. DE MOOR, AND J. VANDEWALLE, *On the best rank-1 and rank-(r_1, r_2,..., r_n) approximation of higher-order tensors*, SIAM journal on Matrix Analysis and Applications, 21 (2000), pp. 1324–1342.

[11] R. S. DEMBO, S. C. EISENSTAT, AND T. STEIHAUG, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.

[12] S. C. EISENSTAT AND H. F. WALKER, *Globally convergent inexact Newton methods*, SIAM J. Optim., 4 (1994), pp. 393–422.

[13] L. ELDÉN AND B. SAVAS, *A Newton-Grassmann method for computing the best multilinear rank-(r_1, r_2, r_3) approximation of a tensor*, SIAM J. Matrix Anal. Appl., 31 (2009), pp. 248–271.

[14] S. FRIEDLAND, V. MEHRMANN, R. PAJAROLA, AND S. K. SUTER, *On best rank one approximation of tensors*, Numer. Linear Algebra Appl., 20 (2013), pp. 942–955.

[15] W. GROPP AND J. J. MORÉ, *Optimization environments and the neos server*, in Approximation Theory and Optimization, M. D. Buhman and A. Iserles, eds., Cambridge University Press, 1997, p. 167.

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

[17] ———, *SVD-based algorithms for the best rank-1 approximation of a symmetric tensor*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 1095–1115.

[18] R. A. HARSHMAN AND S. HONG, *"Stretch" vs "slice" methods for representing three-way structure via matrix notation*, Journal of Chemometrics, 16, pp. 198–205.

[19] R. HENRION, *Body diagonalization of core matrices in three-way principal components analysis: Theoretical bounds and simulation*, Journal of Chemometrics, 7 (1993), pp. 477–494.

[20] J.-H. JIANG, H.-L. WU, Y. LI, AND R.-Q. YU, *Three-way data resolution by alternating slice-wise diagonalization (asd) method*, Journal of Chemometrics, 14 (2000), pp. 15–36.

[21] T. G. KOLDA, *A counterexample to the possibility of an extension of the Eckart-Young low-rank approximation theorem for the orthogonal rank tensor decomposition*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 762–767.

[22] ———, *Numerical optimization for symmetric tensor decomposition*, Math. Program., 151 (2015), pp. 225–248.

[23] P. KROONENBERG, *Three-mode Principal Component Analysis: Theory and Applications*, M & T series, DSWO Press, 1983.

[24] J. B. KRUSKAL, *Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics*, Linear Algebra Appl., 18 (1977), pp. 95 – 138.

[25] J. LEVIN, *Three-mode Factor Analysis*, Project on techniques for investigation of structure of individual differences in psychological phenomena, University of Illinois at Urbana-Champaign, 1963.

[26] L.-H. LIM, *Foundations of numerical multilinear algebra: Decomposition and approximation of tensors*, PhD thesis, Stanford University, 2007.

[27] J. M. MARTÍNEZ, *Local convergence theory of inexact Newton methods based on structured least change updates*, Math. Comp., 55 (1990), pp. 143–167.

[28] *MATLAB Optimization Toolbox, Version 8.1*, R2018a.

[29] J. NAGY AND M. KILMER, *Kronecker product approximation for preconditioning in three-dimensional imaging applications*, Image Processing, IEEE Transactions on, 15 (2006), pp. 604 –613.

[30] E. SANCHEZ AND B. R. KOWALSKI, *Tensorial resolution: A direct trilinear decomposition*, Journal of Chemometrics, 4 (1990), pp. 29–45.

[31] A. K. SMILDE, Y. WANG, AND B. R. KOWALSKI, *Theory of medium-rank second-order calibration with restricted-tucker models*, Journal of Chemometrics, 8 (1994), pp. 21–36.

[32] L. SORBER, M. VAN BAREL, AND L. DE LATHAUWER, *Optimization-based algorithms for tensor decompositions: canonical polyadic decomposition, decomposition in rank-$(L_r, L_r, 1)$ terms, and a new generalization*, SIAM J. Optim., 23 (2013), pp. 695–720.

[33] A. STEGEMAN, *Degeneracy in candecomp/parafac and indscal explained for several three-sliced arrays with a two-valued typical rank*, Psychometrika, 72 (2007), pp. 601–619.

[34] G. TOMASI AND R. BRO, *A comparison of algorithms for fitting the parafac model*, Comput. Stat. Data Anal., 50 (2006), pp. 1700–1734.

[35] A. USCHMAJEW, *A new convergence proof for the higher-order power method and generalizations*, Pac. J. Optim., 11 (2015), pp. 309–321.

[36] C. F. VAN LOAN, *The ubiquitous Kronecker product*, J. Comput. Appl. Math., 123 (2000), pp. 85–100.

[37] L. WANG AND M. T. CHU, *On the global convergence of the alternating least squares method for rank-one approximation to generic tensors*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 1058–1072.

[38] Y. YANG, S. HU, L. DE LATHAUWER, AND J. A. SUYKENS, *Convergence study of block singular value maximization methods for rank-1 approximation to higher order tensors*, tech. report, Internal Report 16-149, ESAT-SISTA, KU Leuven, 2016.