

An adaptive gradient method for computing generalized tensor eigenpairs

Gaohang Yu ^{*} Zefeng Yu [†] Yi Xu [‡] Yisheng Song [§]

January 8, 2016

Abstract

High order tensor arises more and more often in signal processing, data analysis, higher-order statistics, as well as imaging sciences. In this paper, an adaptive gradient (AG) method is presented for generalized tensor eigenpairs. Global convergence and linear convergence rate are established under some suitable conditions. Numerical results are reported to illustrate the efficiency of the proposed method. Comparing with the GEAP method, an adaptive shifted power method proposed by Tamara G. Kolda and Jackson R. Mayo [SIAM J. Matrix Anal. Appl., 35 (2014), pp. 1563-1581], the AG method is much faster and could reach the largest eigenpair with a higher probability.

Keywords: Higher order Tensor, Eigenvalue, Eigenvector, Gradient method, Power method.

1 Introduction

A m th-order n -dimensional real tensor \mathcal{A} consists of n^m entries in real numbers:

$$\mathcal{A} = (a_{i_1 i_2 \dots i_m}), \quad a_{i_1 i_2 \dots i_m} \in \mathbb{R}, \quad 1 \leq i_1, i_2, \dots, i_m \leq n.$$

^{*}School of Mathematics and Computer Sciences, Gannan Normal University, Ganzhou, 341000, China. E-mail: maghyu@163.com

[†]School of Mathematics and Computer Sciences, Gannan Normal University, Ganzhou, 341000, China. E-mail: yzf_2000@sina.com

[‡]Department of Mathematics, Southeast University, China. E-mail: yi.xu1983@gmail.com

[§]School of Mathematics and Information Science and Henan Engineering Laboratory for Big Data Statistical Analysis and Optimal Control, Henan Normal University, China. E-mail: songyisheng1@gmail.com

\mathcal{A} is called *symmetric* if the value of $a_{i_1 i_2 \dots i_m}$ is invariant under any permutation of its indices i_1, i_2, \dots, i_m . Recall the definition of tensor product, $\mathcal{A}x^{m-1}$ is a vector in \mathbb{R}^n with its i th component as

$$(\mathcal{A}x^{m-1})_i = \sum_{i_2, \dots, i_m=1}^n a_{i i_2 \dots i_m} x_{i_2} \cdots x_{i_m}. \quad (1)$$

A real symmetric tensor \mathcal{A} of order m dimension n uniquely defines a m th degree homogeneous polynomial function h with real coefficient by

$$h(x) := \mathcal{A}x^m = x^T (\mathcal{A}x^{m-1}) = \sum_{i_1, \dots, i_m=1}^n a_{i_1 \dots i_m} x_{i_1} \cdots x_{i_m}. \quad (2)$$

We call that the tensor \mathcal{A} is positive definite if $\mathcal{A}x^m > 0$ for all $x \neq 0$.

In 2005, Qi [34] and Lim [27] proposed the definition of eigenvalues and eigenvectors for higher order tensors, independently. Furthermore, in [5], these definitions were unified by Chang, Person and Zhang. Let \mathcal{A} and \mathcal{B} be real-valued, m th-order n -dimensional symmetric tensors. Assume further that m is even and \mathcal{B} is positive definite. we call $(\lambda, x) \in \mathbb{R} \times \mathbb{R}^n \setminus \{0\}$ is a **generalized eigenpair** of $(\mathcal{A}, \mathcal{B})$ if

$$\mathcal{A}x^{m-1} = \lambda \mathcal{B}x^{m-1}. \quad (3)$$

When the tensor \mathcal{B} is an identity tensor ε such that $\varepsilon x^{m-1} = \|x\|^{m-2} x$ for all $x \in \mathbb{R}^n$ [5], the eigenpair reduces to Z -eigenpair [34, 27] which is defined as a pair $(\lambda, x) \in \mathbb{R} \times \mathbb{R}^n \setminus \{0\}$ satisfying

$$\mathcal{A}x^{m-1} = \lambda x \quad \text{and} \quad \|x\|^2 = 1. \quad (4)$$

Another special case is that when $\mathcal{B} = \mathcal{I}$ with $(\mathcal{I})_{i_1 i_2 \dots i_m} = \delta_{i_1 i_2 \dots i_m}$ [5], the real scalar λ is called an H -eigenvalue and the real vector x is the associated H -eigenvector of the tensor \mathcal{A} [34].

In the last decade, tensor eigenproblem has received much attention in the literature [7, 13, 25, 42, 43, 45, 46, 47], which has numerous applications in magnetic resonance imaging [41, 2, 38, 36, 39, 40, 9], image analysis [50], data fitting [32, 33], quantum information [31], automatic control [29], higher order Markov chains [26, 8], spectral graph theory [24, 48], multi-label learning [44], and so on. In [39], a positive semidefinite diffusion tensor (PSDT) model was proposed to approximate the apparent diffusion coefficient (ADC) profile for high-order diffusion tensor imaging, where the smallest Z -eigenvalue need to be nonnegative to guarantee the positive definiteness of the diffusivity function. Based on all of the Z -eigenvalues, a generalized fractional anisotropy (GFA) was proposed to characterize the anisotropic diffusion profile for PSDT. GFA is rotationally invariant and independent from the choice of the laboratory coordinate system. In automatic control [29], the smallest eigenvalue of tensors could reflect the stability of a nonlinear autonomous system. In [40], the principal Z -eigenvectors can depict the

orientations of nerve fibers in the voxel of white matter of human brain. Recently, a higher order tensor vessel tractography was proposed for segmentation of vascular structures, in which the principal directions of a 4-dimensional tensor were used in vessel tractography approach [3].

In general, it is NP-hard to compute eigenvalues of a tensor [17]. In [37], a direct method to calculate all of Z-eigenvalues was proposed for two and three dimensional symmetric tensors. For general symmetric tensors, a shifted higher order power method was proposed for computing Z-eigenpairs in [22]. Recently, in [21], an adaptive version of higher order power method was presented for generalized eigenpairs of symmetric tensor. In order to guarantee the convergence of power method, they need a shift to force the objective to be (locally) concave/convex. In this case, the power method is a monotone gradient method with unit-stepsizes. By using fixed-point analysis, linear convergence rate is established for the shifted higher order power method [22]. However, similarly to the case of Matrix, when the largest eigenvalue is close to the second dominant eigenvalue, the convergence of power method will be very slow [11].

In the recent years, there are various optimization approaches were proposed for tensor eigenvalue problem [14, 18, 15, 16, 30, 49]. In [14], Han proposed an unconstrained optimization model for computing generalized eigenpair of symmetric tensors. By using BFGS method to solve the unconstrained optimization, the sequence will be convergent superlinearly. A subspace projection method was proposed in [15] for Z-eigenvalues of symmetric tensors. Recently, in [16], Hao, Cui and Dai proposed a trust region method for Z-eigenvalues of symmetric tensor and the sequence enjoys a locally quadratic convergence rate. In [49], Ni and Qi employed Newton method for the KKT system of optimization problem, and obtained a quadratically convergent algorithm for finding the largest eigenvalue of a nonnegative homogeneous polynomial map. In [10], an inexact steepest descent method was proposed for computing eigenvalues of large scale Hankel tensors. Since nonlinear optimization methods may stop at a local optimum, a sequential semi-definite programming method was proposed by Hu et al. [19] for finding the extremal Z-eigenvalues of tensors. Moreover, in [12], a Jacobian semi-definite relaxation approach was presented to compute all of the real eigenvalues of symmetric tensors.

In practice, one just need to compute extremal eigenvalues or all of its local maximal eigenvalues, for example in MRI [39, 40]. On the other hand, when the order or the dimension of a tensor grows larger, the optimization problem will become large-scale or huge-scale. Therefore, we would like to investigate one simple and low-complexity method for finding tensor eigenpairs. In this paper, we consider an adaptive gradient method for solving the following nonlinear programming problem:

$$\max f(x) = \frac{\mathcal{A}x^m}{\mathcal{B}x^m} \text{ subject to } x \in \mathbb{S}^{n-1}. \quad (5)$$

Where \mathbb{S}^{n-1} denote the unit sphere, i.e., $\mathbb{S}^{n-1} = \{x \in \mathbb{R}^n \mid \|x\|^2 = 1\}$, $\|\cdot\|$ denotes the Euclidean norm. By some simple calculations, we can get its gradient and Hessian, as

follows [21, 10]:

$$g(x) \equiv \nabla f(x) = \frac{m}{\mathcal{B}x^m} (\mathcal{A}x^{m-1} - \frac{\mathcal{A}x^m}{\mathcal{B}x^m} \mathcal{B}x^{m-1}), \quad (6)$$

and its Hessian is

$$\begin{aligned} H(x) &\equiv \nabla^2 f(x) \\ &= \frac{m(m-1)\mathcal{A}x^{m-2}}{\mathcal{B}x^m} - \frac{m(m-1)\mathcal{A}x^m \mathcal{B}x^{m-2} + m^2(\mathcal{A}x^{m-1} \odot \mathcal{B}x^{m-1})}{(\mathcal{B}x^m)^2} \\ &\quad + \frac{m^2 \mathcal{A}x^m (\mathcal{B}x^{m-1} \odot \mathcal{B}x^{m-1})}{(\mathcal{B}x^m)^3}, \end{aligned}$$

where $x \odot y = xy^T + yx^T$, and $\mathcal{A}x^{m-2}$ is a matrix with its component as

$$(\mathcal{A}x^{m-2})_{i_1 i_2} = \sum_{i_3, \dots, i_m=1}^n a_{i_1 i_2 i_3 \dots i_m} x_{i_3} \cdots x_{i_m} \quad \text{for all } i_1, i_2 = 1, \dots, n.$$

According to (6), we can derive an important property for the nonlinear programming problem (5) that the gradient $g(x)$ is located in the tangent plane of \mathbb{S}^{n-1} at x [10], since

$$x^T g(x) = \frac{m}{\mathcal{B}x^m} (x^T \mathcal{A}x^{m-1} - \frac{\mathcal{A}x^m}{\mathcal{B}x^m} x^T \mathcal{B}x^{m-1}) = 0. \quad (7)$$

Let \bar{x} is a constrained stationary point of (5), i.e., that $\langle g(\bar{x}), x - \bar{x} \rangle \leq 0$ for all $x \in \mathbb{S}^{n-1}$. Then we can claim that **every constrained stationary point of (5) must be a stationary point** of $f(x)$ since $\langle g(\bar{x}), x \rangle \leq 0$ should be hold for all $x \in \mathbb{S}^{n-1}$. Otherwise, if $\|g(\bar{x})\| \neq 0$, we could choose $x = \frac{g(\bar{x})}{\|g(\bar{x})\|}$, and then $\|g(\bar{x})\| \leq 0$.

Suppose $x \in \mathbb{S}^{n-1}$ and denote $\lambda = \frac{\mathcal{A}x^m}{\mathcal{B}x^m}$. By $g(x) = 0$, we know that any KKT point of (5) will be a solution of the system of equations (3). Before end of this section, we would like to state the following theorem and its proof is omitted.

Theorem 1 *If the gradient $g(x)$ at x vanishes, then $\lambda = f(x)$ is a generalized eigenvalue and the vector x is the associated generalized eigenvector.*

The rest of this paper is organized as follows. In the next section, we introduce some existed gradient methods for tensor eigenvalue problems. In Section 3, based on a curvilinear search scheme, we present a inexact gradient method. Then, we establish its global convergence and linear convergence results under some suitable assumptions. Section 4 provides numerical experiments to show the efficiency of our gradient method. Finally, we have a conclusion section.

2 Some exist gradient methods for tensor eigenpairs

2.1 Gradient method with fixed stepsize–power method

The symmetric higher-order power method (S-HOPM) was introduced by De Lathauwer, De Moor, and Vandewalle [23] for solving the following optimization problem:

$$\max f(x) = \mathcal{A}x^m \quad \text{subject to } x \in \mathbb{S}^{n-1}. \quad (8)$$

This problem is equivalent to finding the largest Z-eigenvalue of \mathcal{A} [34] and is related to finding the best symmetric rank-1 approximation of a symmetric tensor $\mathcal{A} \in \mathbb{S}^{[m,n]}$ [23].

Algorithm 1: Symmetric higher order power Method (S-HOPM) [23]

Given a tensor $\mathcal{A} \in \mathbb{S}^{[m,n]}$, an initial unit iterate x_0 . Let $\lambda_0 = \mathcal{A}x_0^m$.

for $k = 0, 1, \dots$ **do**

1: $\hat{x}_{k+1} \leftarrow \mathcal{A}x_k^{m-1}$

2: $x_{k+1} \leftarrow \hat{x}_{k+1} / \|\hat{x}_{k+1}\|$

3: $\lambda_{k+1} = \mathcal{A}x_{k+1}^m$

End for

The cost per iteration of power method is $O(mn^m)$, mainly for computing $\mathcal{A}x_k^{m-1}$. Let $g(x_k) = \nabla f(x_k) = \frac{1}{m}\mathcal{A}x_k^{m-1}$. Set $d_k = g(x_k) - x_k$, $\hat{x}_{k+1} = x_k + d_k$, then the main iteration could be reformulated as $x_{k+1} = \frac{x_k + d_k}{\|x_k + d_k\|}$, which is a projected gradient method with unit-stepsize. Kofidis and Regalia [20] pointed out that S-HOPM method can not guarantee to converge. By using convexity theory, they show that S-HOPM method could be convergent for even-order tensors under the convexity assumption on $\mathcal{A}x^m$. For general symmetric tensors, a shifted S-HOPM (SS-HOPM) method was proposed by Kolda and Mayo [22] for computing Z-eigenpairs. One shortcoming of SS-HOPM is that its performance depended on choosing an appropriate shift. Recently, Kolda and Mayo extended SS-HOPM for computing generalized tensor eigenpairs, called GEAP method which is an adaptive, monotonically convergent, shifted power method for generalized tensor eigenpairs (3). They showed that GEAP method is much faster than the SS-HOPM method due to its adaptive shift choice.

Algorithm 2: GEAP Method [21]

Given tensors $\mathcal{A} \in \mathbb{S}^{[m,n]}$ and $\mathcal{B} \in \mathbb{S}_+^{[m,n]}$, and an initial guess \hat{x}_0 . Let $\beta = 1$ if we want to find the local maxima; otherwise, let $\beta = -1$ for seeking local minima. Let $\tau > 0$ be the tolerance on being positive/negative definite.

for $k = 0, 1, \dots$ **do**

1: Precompute $\mathcal{A}x_k^{m-2}, \mathcal{B}x_k^{m-2}, \mathcal{A}x_k^{m-1}, \mathcal{B}x_k^{m-1}, \mathcal{A}x_k^m, \mathcal{B}x_k^m$

2: $\lambda_k = \mathcal{A}x_k^m / \mathcal{B}x_k^m$

3: $H_k \leftarrow H(x_k) = \nabla^2 f(x_k)$

- 4:** $\alpha_k \leftarrow \beta \max\{0, (\tau - \lambda_{\min}(\beta H_k))/m\}$
5: $\hat{x}_{k+1} \leftarrow \beta(\mathcal{A}x_k^{m-1} - \lambda_k \mathcal{B}x_k^{m-1} + (\alpha_k + \lambda_k)\mathcal{B}x_k^m x_k)$
6: $x_{k+1} \leftarrow \hat{x}_{k+1}/\|\hat{x}_{k+1}\|$
End for

In [28], Ng, Qi and Zhou proposed a power method for finding the largest H-eigenvalue of irreducible nonnegative tensors. It is proved in [6] that NQZ's power method is convergent for primitive nonnegative tensors. Further, Zhang et. al [51, 52] established its linear convergence result and presented some updated version for essentially positive tensors and weakly positive tensors, respectively. However, similarly to the case of Matrix, when the largest eigenvalue is close to the second dominant eigenvalue, the convergence of power method will be very slow [11].

2.2 Gradient method with optimal stepsize

In [15], Hao, Cui and Dai proposed a sequential subspace projection method (SSPM) for Z-eigenvalue of symmetric tensors. In each iteration of SSPM method, one need to solve the following 2-dimensional subproblem:

$$\max_{x \in \text{span}\{x_k, \mathcal{A}x_k^{m-1}\}} f(x) = \mathcal{A}x^m \quad \text{subject to } x \in \mathbb{S}^{n-1}. \quad (9)$$

Let $g_k \triangleq \mathcal{A}x_k^{m-1}$. The point in $\mathbb{S}^{n-1} \cap \text{span}\{x_k, g_k\}$ can be expressed as

$$x(\alpha) = \sqrt{1 - \alpha^2 \|g_k\|^2} x_k + \alpha g_k, \quad -\frac{1}{\|g_k\|} \leq \alpha \leq \frac{1}{\|g_k\|}.$$

If $\alpha_k \equiv \frac{1}{\|g_k\|}$, then SSPM method will reduce to the power method. For simplicity, if $\alpha_k \neq \frac{1}{\|g_k\|}$, the iterate can be expressed as $x(\alpha) = x_k + \sigma g_k$ with $\sigma = \frac{\alpha}{\sqrt{1 - \alpha^2 \|g_k\|^2}}$. In order to solve (9), one need to solve a equation like $\nabla f(x_k + \sigma g_k)^T g_k = (\mathcal{A}(x_k + \sigma g_k)^{m-1})^T g_k = 0$. For each iteration, the computational cost of SSPM method is m times than that of power method. As shown in [15], the main computational cost of SSPM is the tensor-vector multiplications $\mathcal{A}x_k^{m-1}$ and \mathcal{A}_k (defined in [15]), which requires $O(mn^m)$ operations and $O(m^2 n^m)$ operations, respectively.

3 Inexact gradient method

Indicated by the idea in [15], we can present a gradient method with optimal stepsize for computing the generalized tensor eigenpairs problem (5). But we don't want to present it here, since the computational cost per iterate is more expensive than Power method. In this section, we firstly present the following inexact gradient method, and then establish its global convergence and linear convergence results under some suitable assumptions.

Algorithm 3: Adaptive Gradient (AG) method for (5)

Given tensors $\mathcal{A} \in \mathbb{S}^{[m,n]}$ and $\mathcal{B} \in \mathbb{S}_+^{[m,n]}$, an initial unit iterate x_0 , parameter $\rho \in (0, 1)$. Let $\epsilon > 0$ be the tolerance. Set $k=0$; Calculate gradient $g(x_k)$.

While $\|g(x_k)\| > \epsilon$ **do**

1: Generate a stepsize $\frac{1}{\|g_k\|} \geq \alpha_k > 0$ such that $x_k(\alpha) = \sqrt{1 - \alpha^2 \|g_k\|^2} x_k + \alpha g_k$ satisfying

$$f(x_k(\alpha)) \geq f(x_k) + \rho \alpha \|g(x_k)\|^2 \quad (10)$$

2: Update the iterate $x_{k+1} = x_k(\alpha_k)$, calculate $g(x_{k+1})$.

End while

It is clear that $x_{k+1} \in \mathbb{S}^{n-1} \cap \text{span}\{x_k, g_k\}$. Moreover, by using (7), we can show the first-order gain per iterate is $g_k^T(x_{k+1} - x_k) = \alpha_k \|g_k\|^2$. Since the spherical feasible region \mathbb{S}^{n-1} is compact, $\mathcal{B}x^m$ is positive and bounds away from zero, we can get that all the functions and gradients of the objective (5) at feasible points are bounded [10], i.e., there exists a constant $M > 0$ such that for all $x \in \mathbb{S}^{n-1}$,

$$|f(x)| \leq M, \quad \text{and} \quad \|g(x)\| \leq M. \quad (11)$$

The following theorem indicates that the Algorithm 3 is convergent to the KKT point of the problem (5). The constructive proof is motivated by the idea in [15].

Theorem 2 *Suppose that the gradient $g(x)$ is Lipschitz continuous on the unit sphere. Let $\{x_k\}$ is generated by Algorithm 3. Then the inexact curvilinear search condition defined in (10) is well-defined and there exists a positive constant $c > 0$ such that*

$$f(x_{k+1}) - f(x_k) \geq c \|g_k\|^2. \quad (12)$$

Furthermore,

$$\lim_{k \rightarrow \infty} \|g_k\| = 0.$$

Proof. Firstly, we have

$$x'_k(\alpha) = \frac{-\alpha \|g_k\|^2}{\sqrt{1 - \alpha^2 \|g_k\|^2}} x_k + g_k. \quad (13)$$

Furthermore, we can obtain

$$\frac{df(x_k(\alpha))}{d\alpha} \Big|_{\alpha=0} = g(x_k(\alpha))^T x'_k(\alpha) \Big|_{\alpha=0} = \|g_k\|^2. \quad (14)$$

Let $d_k(\alpha) = x_k(\alpha) - x_k$, using (7), we can derived that for any constant $\rho \in (0, 1)$, there exists a positive scalar $\bar{\alpha} \leq \frac{1}{\|g_k\|}$ such that for all $\alpha \in (0, \bar{\alpha}]$,

$$f(x_k(\alpha)) - f(x_k) \geq \rho \langle g_k, d_k(\alpha) \rangle = \rho \alpha \|g(x_k)\|^2.$$

Considering the gap between $f(x_k(\alpha))$ and $f(x_k)$, similarly to the proof of Lemma 4.2 in [15], we can get

$$\begin{aligned}
f(x_k(\alpha)) - f(x_k) &= \int_0^\alpha g(x_k(t))^T x'_k(t) dt \\
&= \alpha g(x_k(0))^T x'_k(0) + \int_0^\alpha g(x_k(t))^T [x'_k(t) - x'_k(0)] dt \\
&\quad + \int_0^\alpha [g(x_k(t)) - g(x_k(0))]^T x'_k(0) dt \\
&\quad \text{(Using (13), (14), and Lipschitz condition)} \\
&\geq \alpha \|g_k\|^2 - M \int_0^\alpha \|x'_k(t) - x'_k(0)\| dt - L \|g_k\| \int_0^\alpha \|x_k(t) - x_k(0)\| dt \\
&\geq \alpha \|g_k\|^2 - M \|g_k\|^2 \int_0^\alpha \frac{t}{\sqrt{1 - t^2 \|g_k\|^2}} dt - L \|g_k\| \int_0^\alpha \sqrt{2 - 2\sqrt{1 - t^2 \|g_k\|^2}} dt
\end{aligned}$$

Without loss of generality, assume that $\alpha_k \leq \tilde{\alpha} \triangleq \frac{\sqrt{3}}{2\|g_k\|}$, then for $t \leq \alpha \leq \tilde{\alpha}$, we have

$$\frac{1}{\sqrt{1 - t^2 \|g_k\|^2}} \leq \frac{1}{\sqrt{1 - \tilde{\alpha}^2 \|g_k\|^2}} = 2,$$

and

$$\sqrt{2 - 2\sqrt{1 - t^2 \|g_k\|^2}} \leq 2t \|g_k\|.$$

So, we can obtain that

$$f(x_k(\alpha)) - f(x_k) \geq (\alpha - M\alpha^2 - L\alpha^2) \|g_k\|^2 = (1 - M\alpha - L\alpha) \alpha \|g_k\|^2. \quad (15)$$

Set $\bar{\alpha} = \frac{1}{2(M+L)}$, we have $1 - M\alpha - L\alpha \geq \frac{1}{2}$ for all $\alpha \in (0, \bar{\alpha}]$. It follows from (15) that (10) holds for all $\alpha \in (0, \bar{\alpha}]$ with $\rho = \frac{1}{2}$. So, by using a backward strategy in curvilinear search, one can claim that the stepsize α_k is bounded from below. That is to say, there exists a positive constant c (e.g. $c = \frac{1}{4(M+L)}$) such that (12) holds. As $f(x)$ is bounded on the unit sphere, by (12), it is easy to prove that $\sum_k \|g_k\|^2 < +\infty$. Therefore, $\lim_{x \rightarrow \infty} \|g_k\| = 0$; namely, the Algorithm 3 is globally convergent. \square

In the rest of this section, we would like to establish the linear convergence rate of the Algorithm 3 under the assumption of second order sufficient condition. For convenience, rewrite (5) as

$$\max f(x) = \frac{\mathcal{A}x^m}{\mathcal{B}x^m} \quad \text{subject to} \quad c(x) = x^T x - 1 = 0. \quad (16)$$

The Lagrangian function is $L(x, \mu) = f(x) - \mu c(x)$, and its gradient and Hessian are

$$\nabla_x L(x, \mu) = \nabla f(x) - \mu x, \quad \text{and} \quad \nabla_x^2 L(x, \mu) = \nabla^2 f(x) - \mu I_n.$$

At the KKT point (x^*, μ^*) , we have $\nabla_x L(x^*, \mu^*) = 0$. So, $\nabla f(x^*)^T x^* - \mu^* \|x^*\|^2 = 0$. By (7), we know $\mu^* = \nabla f(x^*)^T x^* = 0$. We can formulate the sufficient conditions for x^* being a strict local maximizer of (16) as: $g(x^*) = 0$ and

$$v^T H(x^*) v < 0, \forall v \in \nabla c(x^*)^\perp \cap \mathbb{S}^{n-1}, \quad (17)$$

where $g(x) = \nabla f(x)$, $H(x) = \nabla^2 f(x)$, and $x^\perp \equiv \{y \in \mathbb{R}^n | x \perp y\}$. By Theorem 2, the sequence $\{x_k\}$ generated by Algorithm 3 is convergent to a KKT point x^* with $g(x^*) = 0$. If we further assume that the assumption of second sufficient condition (17) holds at x^* , then the following linear convergence theorem could be established for Algorithm 3.

Theorem 3 *Let $\{x_k\}$ is generated by Algorithm 3. Suppose that the gradient $g(x)$ is Lipschitz continuous on the unit sphere and the second sufficient condition (17) holds at the KKT point x^* . Then $\{f(x_k)\}$ converges to $f(x^*)$ linearly.*

Proof. In order to show $\{f(x_k)\}$ converges to $f(x^*)$ linearly, we need to prove

$$0 < \lim_{k \rightarrow \infty} \frac{f(x^*) - f(x_{k+1})}{f(x^*) - f(x_k)} = 1 - \lim_{k \rightarrow \infty} \frac{f(x_{k+1}) - f(x_k)}{f(x^*) - f(x_k)} < 1. \quad (18)$$

To end of this, we firstly deduce that $f(x^*) - f(x_k) > 0$. Project $x_k - x^*$ on the orthogonal space of $\nabla c(x^*) = 2x^*$, we get

$$v_k = x_k - x^* - ((x_k - x^*)^T \nabla c(x^*)) \nabla c(x^*) = x_k - x^* - 4((x_k - x^*)^T x^*) x^*.$$

It is clear that $v_k \in \mathbb{S}^{n-1}$, since $\|x_k\|^2 = \|x^*\|^2 = 1$. Notice that $-(x_k - x^*)^T x^* = 1 - x_k^T x^* = \frac{1}{2}(x_k - x^*)^T (x_k - x^*)$, we can obtain that

$$x_k - x^* = v_k - 2\|x_k - x^*\|^2 x^* = v_k + O(\|x_k - x^*\|^2).$$

By the Taylor expansion, and the second order sufficient condition (17), we have

$$\begin{aligned} f(x^*) - f(x_k) &= -\frac{1}{2}(x_k - x^*)^T H(x^*)(x_k - x^*) + o(\|x_k - x^*\|^2) \\ &= -\frac{1}{2}(v_k + O(\|x_k - x^*\|^2))^T H(x^*)(v_k + O(\|x_k - x^*\|^2)) + o(\|x_k - x^*\|^2) \\ &= -\frac{1}{2}v_k^T H(x^*)v_k + o(\|x_k - x^*\|^2) > 0. \end{aligned}$$

Secondly, let us consider the following limitation:

$$\lim_{k \rightarrow \infty} \frac{\|g_k\|^2}{f(x^*) - f(x_k)} = \lim_{k \rightarrow \infty} \frac{\|H(x^*)(x_k - x^*)\|^2}{-\frac{1}{2}v_k^T H(x^*)v_k} = \lim_{k \rightarrow \infty} \frac{2\|H(x^*)v_k\|^2}{-v_k^T H(x^*)v_k}.$$

Let $U \in \mathfrak{R}^{n \times (n-1)}$ be the orthogonal complement of the vector $\nabla c(x^*)$. For any $v \in \nabla c(x^*)^\perp$, there exist $w \in \mathfrak{R}^{n-1}$ such that $v = Uw$. Suppose that the Cholesky decomposition of the positive definite matrix $-U^T H(x^*)U$ is $D^T D$. Denoting $y = Dw$. Notice that $\|U\|_2 = \sqrt{\|U^T U\|} = 1$, we can derive that:

$$\begin{aligned} \frac{2\|H(x^*)v\|^2}{-v^T H(x^*)v} &= \frac{2\|H(x^*)Uw\|^2}{-w^T U^T H(x^*)Uw} \\ &= \frac{2\|U\| \|H(x^*)Uw\|^2}{-w^T U^T H(x^*)Uw} \\ &\geq \frac{2\|U^T H(x^*)Uw\|^2}{-w^T U^T H(x^*)Uw} \\ &= \frac{-2y^T U^T H(x^*)Uy}{y^T y} \\ &\geq 2\lambda_{n-1}, \end{aligned}$$

where λ_{n-1} is the smallest eigenvalue of the matrix $-U^T H(x^*)U$. Therefore, we have

$$\lim_{k \rightarrow \infty} \frac{\|g_k\|^2}{f(x^*) - f(x_k)} \geq 2\lambda_{n-1}. \quad (19)$$

It follows from (12) and (19) that

$$\begin{aligned} \lim_{k \rightarrow \infty} \frac{f(x^*) - f(x_{k+1})}{f(x^*) - f(x_k)} &= 1 - \lim_{k \rightarrow \infty} \frac{f(x_{k+1}) - f(x_k)}{f(x^*) - f(x_k)} \\ &\leq 1 - \lim_{k \rightarrow \infty} \frac{c\|g_k\|^2}{f(x^*) - f(x_k)} \\ &\leq 1 - 2c\lambda_{n-1} \\ &\leq 1. \end{aligned}$$

The proof is completed. \square

4 Numerical experiments

In this section, we present some numerical results to illustrate the effectiveness of the proposed adaptive gradient (AG) method, which was compared with the GEAP method—an adaptive shifted power method proposed by Tamara G. Kolda and Jackson R. Mayo [21]. The experiments were done on a laptop with Intel Core 2 Duo CPU with a 4GB RAM, using MATLAB R2014b, and the Tensor Toolbox [1]. We set the parameter $\rho = 0.001$ and initial guess of the stepsize α in (10) is generated by $\min(\frac{1}{\|g_k\|}, \frac{\|\Delta x_k\|}{\|\Delta g_k\|})$. If

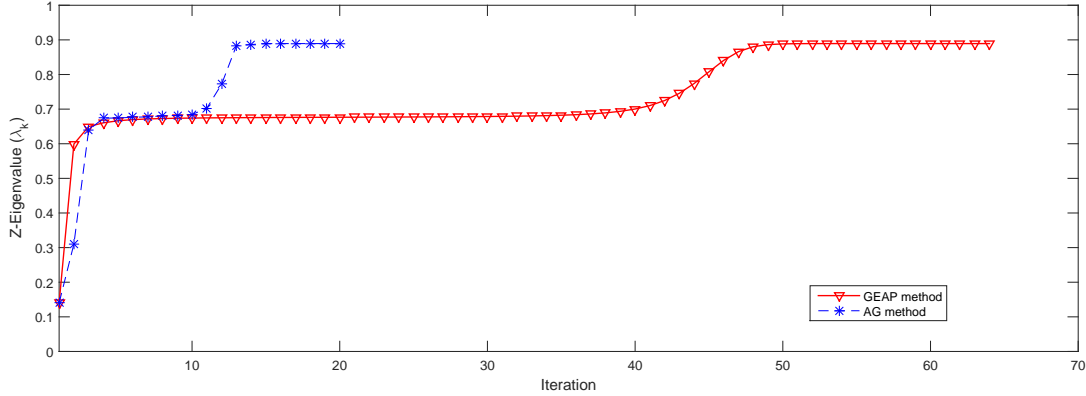


Figure 1: Comparison with GEAP algorithm for computing Z-eigenvalues of \mathcal{A} from *Example 1*, and the starting point is $x_0 = [0.0417 \ -0.5618 \ 0.6848]$

this initial guess can not satisfy the line search condition (10), then we truncate it as $\alpha = 0.5 * \alpha$, and try it again. Generally, once or twice is enough in our experiments.

In all numerical experiments, we stop the iterates once $|\lambda_{k+1} - \lambda_k| \leq 10^{-10}$. The maximum iterations is 500.

4.1 Comparison with GEAP for computing Z-eigenpairs

The following example is originally from [20] and was used in evaluating the SS-HOPM algorithm in [22] and the GEAP algorithm in [21] for computing Z-eigenpairs.

Example 1 (Kofidis and Regalia [20]). Let $\mathcal{A} \in \mathbb{S}^{[4,3]}$ be the symmetric tensor defined by

$$\begin{aligned}
 a_{1111} &= 0.2883, & a_{1112} &= -0.0031, & a_{1113} &= 0.1973, & a_{1122} &= -0.2485, \\
 a_{1223} &= 0.1862, & a_{1133} &= 0.3847, & a_{1222} &= 0.2972, & a_{1123} &= -0.2939, \\
 a_{1233} &= 0.0919, & a_{1333} &= -0.3619, & a_{2222} &= 0.1241, & a_{2223} &= -0.3420, \\
 & & a_{2233} &= 0.2127, & a_{2333} &= 0.2727, & a_{3333} &= -0.3054.
 \end{aligned}$$

To compare the convergence in terms of the number of iterations. Figure 1 shows the results for computing Z-eigenvalues of \mathcal{A} from *Example 1*, and the starting point is $x_0 = [0.0417 \ -0.5618 \ 0.6848]$. In this case, both of Adaptive Gradient (AG) method and GEAP method can find the largest Z-eigenvalue 0.8893. AG method just need run 19 iterations in 0.168521 seconds while GEAP method need run 63 iterations in 0.469648 seconds.

We used 1000 random starting guesses, each entry selected uniformly at random from the interval $[-1, 1]$. For each set of experiments, the same set of random starts was used.

For the largest eigenpair, we list the number of occurrences in the 1000 experiments. We also list the median number of iterations until convergence, the average error and the average run time in the 1000 experiments in Tables 1-4. As we can see from Tables 1-4, Adaptive Gradient (AG) method is much faster than GEAP method and could reach the largest eigenpair with a higher probability.

Table 1. Comparison results for computing Z-eigenvalues of \mathcal{A} from *Example 1*.

Alg.	Occ.	λ	Its.	Error	Time (sec.)
GEAP	49.9%	0.8893	27.06	5.69e-11	0.1632
AG	56.6%	0.8893	13.81	1.74e-11	0.1205

Example 2 (Nie and Wang [12]). Let $\mathcal{A} \in \mathbb{S}^{[4,n]}$ be the symmetric tensor defined by

$$a_{ijkl} = \sin(i + j + k + l) \quad (1 \leq i, j, k, l \leq n).$$

For the case of $n = 5$, there are five real Z-eigenvalues which are respectively

$$\lambda_1 = 7.2595, \lambda_2 = 4.6408, \lambda_3 = 0.0000, \lambda_4 = -3.9204, \lambda_5 = -8.8463.$$

Table 2. Comparison results for computing Z-eigenvalues of \mathcal{A} from *Example 2*.

Alg.	Occ.	λ	Its.	Error	Time (sec.)
GEAP	48.2%	7.2595	50.01	7.68e-11	0.3235
AG	54.6%	7.2595	24.85	4.72e-11	0.2286

Example 3 (Nie and Wang [12]). Let $\mathcal{A} \in \mathbb{S}^{[4,n]}$ be the symmetric tensor defined by

$$a_{ijkl} = \tan(i) + \tan(j) + \tan(k) + \tan(l) \quad (1 \leq i, j, k, l \leq n).$$

For the case of $n = 5$, there are five real Z-eigenvalues which are respectively

$$\lambda_1 = 34.5304, \lambda_2 = 0.0000, \lambda_3 = -101.1994.$$

Table 3. Comparison results for computing Z-eigenvalues of \mathcal{A} from *Example 3*.

Alg.	Occ.	λ	Its.	Error	Time (sec.)
GEAP	64.0%	34.5304	28.07	5.84e-11	0.1701
AG	83.9%	34.5304	17.70	3.17e-11	0.1544

Example 4 (Nie and Wang [12]). Let $\mathcal{A} \in \mathbb{S}^{[4,n]}$ be the symmetric tensor defined by

$$a_{ijkl} = \arctan\left((-1)^i \frac{i}{n}\right) + \arctan\left((-1)^j \frac{j}{n}\right) + \arctan\left((-1)^k \frac{k}{n}\right) + \arctan\left((-1)^l \frac{l}{n}\right)$$

Table 4. Comparison results for computing Z-eigenvalues of \mathcal{A} from *Example 4* ($n=5$).

Alg.	Occ.	λ	Its.	Error	Time (sec.)
GEAP	65.2%	13.0779	22.43	5.64e-11	0.1453
AG	87.7%	13.0779	13.88	2.66e-11	0.1242

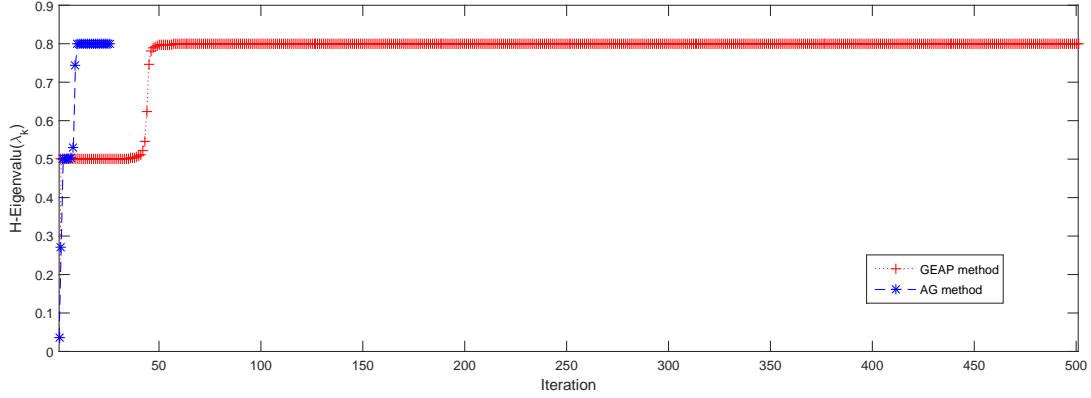


Figure 2: Comparison with GEAP algorithm for computing H-eigenvalues of \mathcal{A} from *Example 5* ($n=5$), and the starting point is $x_0 = [-0.8181 \ -0.4264 \ -0.0163 \ 0.1198 \ -0.1574]$

4.2 Comparison with GEAP for computing H-eigenpairs

In this subsection, we test the proposed AG method with comparison to GEAP method on finding H-eigenpairs.

To compare the convergence in terms of the number of iterations. Figure 2 shows the results for computing H-eigenvalues of \mathcal{A} from *Example 5* ($n=5$), and the starting point is $x_0 = [-0.8181 \ -0.4264 \ -0.0163 \ 0.1198 \ -0.1574]$. In this case, GEAP fails to stop in 500 iterations. But Adaptive Gradient (AG) method can find the largest H-eigenvalue 0.8 after running 25 iterations in 0.2183 seconds.

Example 5. Let $\mathcal{A} \in \mathbb{S}^{[4,n]}$ be the diagonal tensor defined by $a_{iiii} = \frac{i-1}{i}$.

Table 5. Comparison results for computing H-eigenvalues of \mathcal{A} from *Example 5* ($n=5$).

Alg.	Occ.	λ	Its.	Error	Time (sec.)
GEAP	81%	0.8	500	4.04e-05	2.6941
AG	94%	0.8	14.48	4.34e-11	0.0872

Example 6 (Nie and Wang [12]). Let $\mathcal{A} \in \mathbb{S}^{[4,n]}$ be the tensor defined by

$$a_{ijkl} = \frac{(-1)^i}{i} + \frac{(-1)^j}{j} + \frac{(-1)^k}{k} + \frac{(-1)^l}{l}, (1 \leq i, j, k, l \leq n).$$

Table 6. Comparison results for computing H-eigenvalues of \mathcal{A} from *Example 6* ($n=5$).

Alg.	Occ.	λ	Its.	Error	Time (sec.)
GEAP	61%	34.3676	20.94	5.32e-11	0.1209
AG	100%	34.3676	15.71	1.93e-11	0.2650

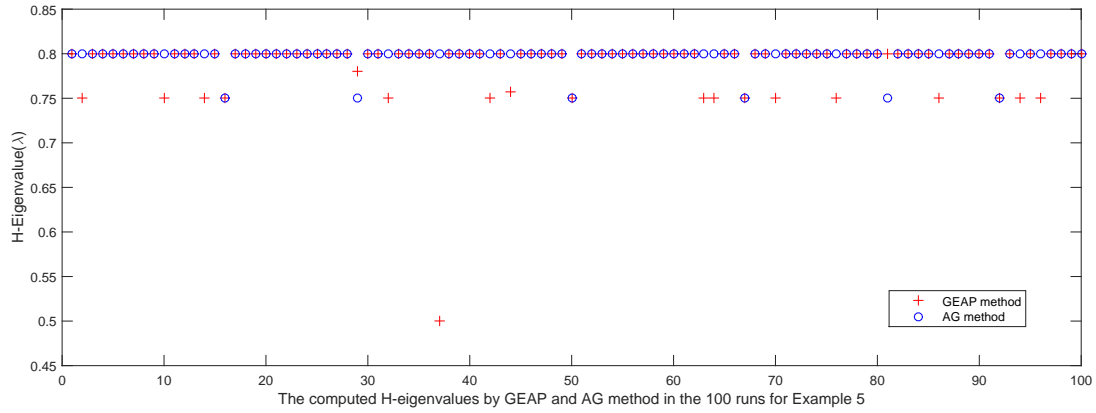


Figure 3: The computed H-eigenvalues by GEAP and AG method in the 100 runs on the \mathcal{A} from *Example 5* ($n=5$).

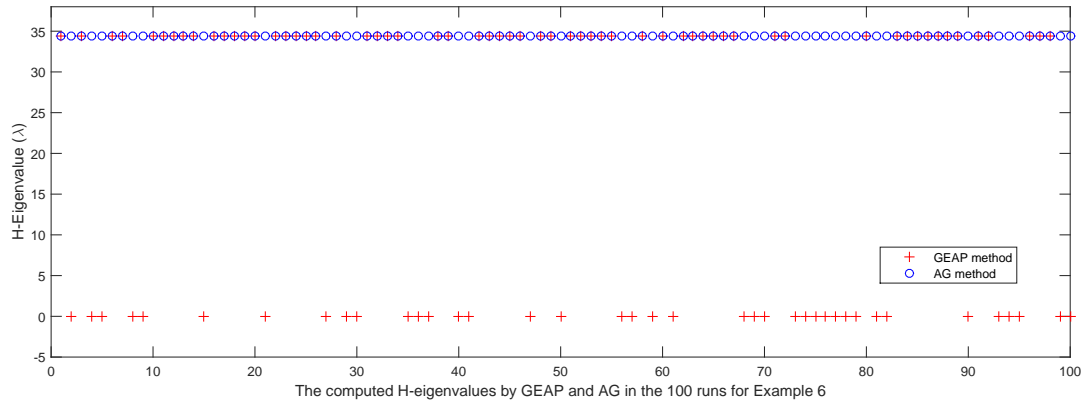


Figure 4: The computed H-eigenvalues by GEAP and AG method in the 100 runs on the \mathcal{A} from *Example 6* ($n=5$).

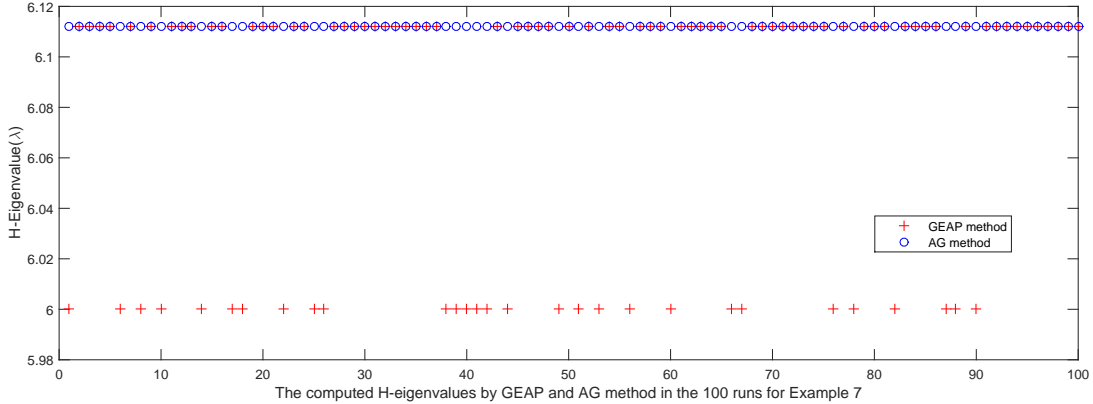


Figure 5: The computed H-eigenvalues by GEAP and AG method in the 100 runs on the \mathcal{A} from *Example 7* ($n=3, b=1$).

Example 7. Let $\mathcal{A} \in \mathbb{S}^{[4,n]}$ be the tensor defined by

$$a_{iii} = 2i, (1 \leq i \leq n) \text{ and } a_{1123} = 4b,$$

here b is a parameter. Then use the **symmetrize** function in the Matlab Tensor Toolbox to symmetrize it.

Table 7. Comparison results for computing H-eigenvalues of \mathcal{A} from *Example 7* ($n = 3, b = 1$).

Alg.	Occ.	λ	Its.	Error	Time (sec.)
GEAP	71%	6.112	179.01	1.99e-08	0.9536
AG	100%	6.112	50.52	5.94e-11	0.5198

We used 100 random starting guesses to test AG method and GEAP method for computing H-eigenvalues of \mathcal{A} from *Examples 5-7*. For each set of experiments, the same set of random starts was used. For the largest eigenpair, we list the number of occurrences in the 100 experiments. We also list the median number of iterations until convergence, the average error and the average run time in the 100 experiments in Tables 5-7. As we can see from Table 5, GEAP method fails to stop in 500 iterations for all of the 100 test experiments for Example 5. But GEAP can slowly approach to the largest H-eigenvalue in 81 test experiments as shown in Figure 3. Adaptive Gradient (AG) method much faster than GEAP method and could reach the largest eigenpair with a higher probability. Especially, for Examples 6 and 7, Adaptive Gradient (AG) method could find the largest H-eigenvalue in all of the 100 experiments.

5 Conclusion

In this paper, we introduced an adaptive gradient (AG) method for generalized tensor eigenpairs, which could be viewed as an inexact version of the gradient method with optimal stepsize for finding Z-eigenvalues of tensor in [15]. What we have done is to use an inexact curvilinear search condition to replace the constraint on optimal stepsize. So, the computational complexity of AG method is much cheaper than SSPM method in [15]. Global convergence and linear convergence rate are established for the AG method for computing generalized eigenpairs of symmetric tensor. Some numerical experiments illustrated that the AG method is faster than GEAP and could reach the largest eigenpair with a higher probability.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (No.61262026, 11571905, 11501100), NCET Programm of the Ministry of Education (NCET 13-0738), JGZX programm of Jiangxi Province (20112BCB23027), Natural Science Foundation of Jiangxi Province (20132BAB201026), science and technology programm of Jiangxi Education Committee (LDJH12088), Program for Innovative Research Team in University of Henan Province (14IRTSTHN023).

References

- [1] B. W. Bader, T. G. Kolda and others. MATLAB Tensor Toolbox Version 2.6, Available online, February 2015. <http://www.sandia.gov/~tgkolda/TensorToolbox/>
- [2] L. Bloy, R. Verma, “On Computing the Underlying Fiber Directions from the Diffusion Orientation Distribution Function”, In *Medical Image Computing and Computer-Assisted Intervention MICCAI*, Vol. 2008. Springer: Berlin / Heidelberg, 2008: 1-8.
- [3] S. Cetin and G. Unal, “A Higher-Order Tensor Vessel Tractography for Segmentation of Vascular Structures”, *IEEE Transactions on Medical Imaging* 34 (2015) 2172-2185.
- [4] K.C. Chang, K.J. Pearson and T. Zhang, “Perron-Frobenius Theorem for nonnegative tensors”, *Communications in Mathematical Sciences*, 6 (2008) 507-520.
- [5] K.C. Chang, K.J. Pearson and T. Zhang, “On eigenvalue problems of real symmetric tensors”, *J. Math. Anal. Appl.*, 350 (2009) 416-422.

- [6] K. Chang, K. Pearson, T. Zhang, “Primitivity, the convergence of the NQZ method, and the largest eigenvalue for nonnegative tensors”, *SIAM Journal on Matrix Analysis and Applications* 32 (2011) 806-819.
- [7] K. Chang, L. Qi, and T. Zhang, “A survey on the spectral theory of nonnegative tensors,” *Numerical Linear Algebra with Applications*, 20 (2013) 891-912.
- [8] K. Chang, T. Zhang, “On the uniqueness and nonuniqueness of the Z-eigenvector for transition probability tensors”, *Journal of Mathematical Analysis and Applications* 408 (2013) 525-540.
- [9] Y. Chen, Y. Dai, D. Han, and W. Sun, “Positive semidefinite generalized diffusion tensor imaging via quadratic semidefinite programming”, *SIAM J. Imaging Sci.*, 6 (2013) 1531-1552.
- [10] Y. Chen, L. Qi and Q. Wang, “Computing eigenvalues of large scale Hankel tensors”, *arXiv: 1504.07413v2*, May 2015.
- [11] M. Chu, S. Wu, “On the second dominant eigenvalue affecting the Power method for transition probability tensors”, Manuscript, 2014.
- [12] C. Cui, Y. Dai and J. Nie, “All real eigenvalues of symmetric tensors”, *SIAM J. Matrix Anal. Appl.*, 35 (2014) 1582-1601.
- [13] W. Ding and Y. Wei, “Generalized tensor eigenvalue problems”, *SIAM J. Matrix Anal. Appl.*, 36 (2015) 1073-1099.
- [14] L. Han, “An unconstrained optimization approach for finding real eigenvalues of even order symmetric tensors”, *Numer. Algebr. Control Optim.* 3 (2013) 583-599.
- [15] C. Hao, C. Cui and Y. Dai, “A sequential subspace projection method for extreme Z-eigenvalues of supersymmetric tensors”, *Numer. Linear Algebr. Appl.*, 22 (2015) 283-298.
- [16] C. Hao, C. Cui and Y. Dai, “A feasible trust-region method for calculating extreme Z-eigenvalues of symmetric tensors”, *Pacific J. Optim.*, 11 (2015) 291-307.
- [17] C. Hillar and L. Lim, “Most tensor problems are NP-hard”, *J. ACM* 60 (2013) article No. 45: 1-39.
- [18] S. Hu, G. Li, L. Qi, Y. Song, “Finding the maximum eigenvalue of essentially non-negative symmetric tensors via sum of squares programming on the largest eigenvalue of a symmetric nonnegative tensor”, *Journal of Optimization Theory and Applications* 2013.

- [19] S. Hu, Z. Huang and L. Qi, “Finding the extreme Z-eigenvalues of tensors via a sequential SDPs method”, *Numer. Linear Algebr. Appl.*, 20 (2013) 972-984.
- [20] E. Kofidis and P. Regalia, “On the best rank-1 approximation of higher-order supersymmetric tensors”, *SIAM J. Matrix Anal. Appl.*, 23 (2002) 863-884.
- [21] T. G. Kolda and J. R. Mayo, “An adaptive shifted power methods for computing generalized tensor eigenpairs”, *SIAM J. Matrix Anal. Appl.*, 35 (2014) 1563-1581.
- [22] T. G. Kolda and J. R. Mayo, “Shifted power method for computing tensor eigenpairs”, *SIAM J. Matrix Anal. Appl.*, 32 (2011) 1095-1124.
- [23] L. De Lathauwer, B. De Moor, and J. Vandewalle, “On the best rank-1 and rank-(R1,R2, . . . ,RN) approximation of higher-order tensors”, *SIAM J. Matrix Anal. Appl.*, 21 (2000) 1324-1342.
- [24] G. Li, L. Qi, G. Yu, “The Z-eigenvalues of a symmetric tensor and its application to spectral hypergraph theory”, *Numerical Linear Algebra with Applications*, 20 (2013) 1001-1029.
- [25] G. Li, L. Qi, G. Yu, “Semismoothness of the maximum eigenvalue function of a symmetric tensor and its application”, *Linear Algebra and its Applications*, 438 (2013) 813-833.
- [26] W. Li, M. Ng, “On the limiting probability distribution of a transition probability tensor”, *Linear and Multilinear Algebra*, 62 (2014) 362-385.
- [27] L.H. Lim, “ Singular values and eigenvalues of tensors, A variational approach”, Proc. 1st IEEE International workshop on computational advances of multi-tensor adaptive processing, 2005, 129-132.
- [28] M. Ng, L. Qi, G. Zhou, “Finding the largest eigenvalue of a nonnegative tensor”, *SIAM Journal on Matrix Analysis and Applications* 31 (2010) 1090-1099.
- [29] Q. Ni, L. Qi and F. Wang, “An eigenvalue method for testing positive definiteness of a multivariate form”, *IEEE T. Automat. Contr.*, 53 (2008) 1096-1107.
- [30] Q. Ni and L. Qi, “A quadratically convergent algorithm for finding the largest eigenvalue of a nonnegative homogeneous polynomial map”, *Journal of Global Optimization*, 61 (2015) 627-641.
- [31] G. Ni, L. Qi and M. Bai, “Geometric measure of entanglement and U-eigenvalues of tensors”, *SIAM J. Matrix Anal. Appl.*, 35 (2014) 73-87.

- [32] J.M. Papy, L. De Lathauwer and S. Van Huffel, “Exponential data fitting using multilinear algebra: the single-channel and multi-channel case”, *Numer. Linear Algebr. Appl.*, 12 (2005) 809-826.
- [33] J.M. Papy, L. De Lathauwer and S. Van Huffel, “Exponential data fitting using multilinear algebra: the decimative case”, *J. Chemometr.* 23 (2009) 341-351.
- [34] L. Qi, “Eigenvalues of a real supersymmetric tensor”, *J. Symbolic Computation*, 40 (2005) 1302-1324.
- [35] L. Qi, “Eigenvalues and invariants of tensor”, *J. Math. Anal. Appl.*, 325 (2007) 1363-1377.
- [36] L. Qi, D. Han and E.X. Wu, “Principal invariants and inherent parameters of diffusion kurtosis tensors”, *Journal of Mathematical Analysis & Applications*, 349 (2009) 165-180.
- [37] L. Qi, F. Wang and Y. Wang, “Z-eigenvalue methods for a global polynomial optimization problem”, *Mathematical Programming*, 118 (2009) 301-316.
- [38] L. Qi, Y. Wang and E.X. Wu, “D-Eigenvalues of diffusion kurtosis tensors”, *Journal of Computational and Applied Mathematics*, 221 (2008) 150-157.
- [39] L. Qi, G. Yu and E.X. Wu, “Higher order positive semi-definite diffusion tensor imaging”, *SIAM Journal on Imaging Sciences*, 3 (2010) 416-433.
- [40] L. Qi, G. Yu and Y. Xu, “Nonnegative diffusion orientation distribution function”, *Journal of Mathematical Imaging and Vision*, 45 (2013) 103-113.
- [41] T. Schultz and H.-P. Seidel, “Estimating crossing fibers: a tensor decomposition approach”, *IEEE T. Vis. Comput. Gr.*, 14 (2008) 1635-1642.
- [42] Y. Song and L. Qi, “Tensor Complementarity Problem and Semi-positive Tensors”, *Journal of Optimization Theory and Applications*, to appear.
- [43] Y. Song and G. Yu, “Properties of Solution Set of Tensor Complementarity Problem”, *arXiv:1508.00069v2*, Aug. 2015.
- [44] L. Sun, S. Ji and J. Ye, “Hypergraph spectral learning for multi-label classification”, in Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2008, pp. 668-676.
- [45] L. Yang, Q. Yang, X. Zhao, “Quadratic third-order tensor optimization problem with quadratic constraints”, *Statistics, Optimization and Information Computing*, 2 (2014) 130-146.

- [46] Q. Yang, Y. Yang, “Further results for Perron-Frobenius Theorem for nonnegative tensors II”, *SIAM Journal on Matrix Analysis and Applications* 32 (2011) 1236-1250.
- [47] Y. Yang, Q. Yang, “Further results for Perron-Frobenius Theorem for nonnegative tensors”, *SIAM Journal on Matrix Analysis and Applications* 31 (2010) 2517-2530.
- [48] J. Xie, A. Chang, “H-eigenvalues of the signless Laplacian tensor for an even uniform hypergraph”, *Frontiers of Mathematics in China*, 8 (2013) 107-127.
- [49] M. Zeng and Q. Ni, “Quasi-Newton method for computing Z-eigenpairs of a symmetric tensor”, *Pacific J. Optim.*, 11 (2015) 279-290.
- [50] F. Zhang, B. Zhou, L. Peng, “Gradient skewness tensors and local illumination detection for images”, *Journal of Computational and Applied Mathematics* 237 (2013) 663-671.
- [51] L. Zhang, L. Qi, “Linear convergence of an algorithm for computing the largest eigenvalue of a nonnegative tensor”, *Numerical Linear Algebra with Applications* 19 (2012) 830-841.
- [52] L. Zhang, L. Qi, Y. Xu, “Finding the largest eigenvalue of an irreducible nonnegative tensor and linear convergence for weakly positive tensors”, *J Comput Math*, 30 (2012) 24-33.