Applied Mathematics and Computation 216 (2010) 1859–1867

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

A modified Newton's method for best rank-one approximation to tensors

Jingya Chang [a], Wenyu Sun [a,*], Yannan Chen [b]

[a] School of Mathematical Science, Nanjing Normal University, Nanjing 210097, China
[b] College of Science, Nanjing Forestry University, Nanjing 210037, China

ARTICLE INFO

Keywords:
Numerical multilinear algebra
Tensor computation
Rank-one approximation
Modified Newton's method
Projective direction

ABSTRACT

In this paper, a modified Newton's method for the best rank-one approximation problem to tensor is proposed. We combine the iterative matrix of Jacobi–Gauss–Newton (JGN) algorithm or Alternating Least Squares (ALS) algorithm with the iterative matrix of GRQ-Newton method, and present a modified version of GRQ-Newton algorithm. A line search along the projective direction is employed to obtain the global convergence. Preliminary numerical experiments and numerical comparison show that our algorithm is efficient.

1. Introduction

Numerical multilinear algebra, or tensor computation, which includes the studies of tensor computation, tensor approximations [9], tensor decompositions [10], rank of tensor [14], eigenvalues of tensor [13,15] and so on, received more attentions of scholars. Tensors are used in many fields, such as chemometrics [3], psychometrics, economics and high-order statistics [6,5,8,11]. A comprehensive survey about tensor computation can be found in [12]. Low rank approximation for higher order tensor produces a compact representation from a huge amount of redundant data and reduces the dimensionality. In this paper, we focus on the best rank-one approximation of tensors.

For notational simplicity, we only consider the approximation of a third order tensor. The generalizations to higher order cases are straightforward.

For a 3rd-order tensor $\mathscr{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, whose elements are a_{ijk}, where $i = 1, \ldots, I_1, j = 1, \ldots, I_2, k = 1, \ldots, I_3$, the best rank-one approximation problem is searching for vectors $x = (x_1, \ldots, x_{I_1})^T, y = (y_1, \ldots, y_{I_2})^T$ and $z = (z_1, \ldots, z_{I_3})^T$ which minimize the following problem

$$\min \quad \sum_{i,j,k} (a_{ijk} - x_i y_j z_k)^2. \tag{1.1}$$

Note that in (1.1), each x, y or z is determined up to a scaling factor. Therefore, we can introduce a parameter λ and constraints $\|x\| = \|y\| = \|z\| = 1$, where the norm denotes the Euclidean norm. So we can write (1.1) as the following constrained optimization problem

$$\begin{aligned} \min \quad & \sum_{i,j,k} (a_{ijk} - \lambda x_i y_j z_k)^2, \\ \text{s.t.} \quad & \|x\| = \|y\| = \|z\| = 1. \end{aligned} \tag{1.2}$$

An equivalent dual constrained optimization problem of (1.2) is as follows (see [17]).

* Corresponding author.
E-mail address: wysun@njnu.edu.cn (W. Sun).

0096-3003/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2009.12.019

Theorem 1.1. *For a real 3rd-order tensor $\mathscr{A}$, the minimization problem (1.2) is equivalent to the maximization problem*

$$\begin{aligned} \max \quad & \sum_{i,j,k} a_{ijk} x_i y_j z_k \\ \text{s.t.} \quad & \|x\| = \|y\| = \|z\| = 1. \end{aligned} \tag{1.3}$$

For the above problem, its first-order necessary condition is the following equations (see [17]):

$$\begin{cases} \lambda x_i = \sum_{j,k} a_{ijk} y_j z_k, & i = 1, \ldots, I_1, \\ \lambda y_j = \sum_{i,k} a_{ijk} x_i z_k, & j = 1, \ldots, I_2, \\ \lambda z_k = \sum_{i,j} a_{ijk} x_i y_j, & k = 1, \ldots, I_3, \\ \lambda = \sum_{i,j,k} a_{ijk} x_i y_j z_k. \end{cases} \tag{1.4}$$

So the best rank-one approximation problem to tensors is transformed to Eqs. (1.4). There are several algorithms for solving (1.4). The most commonly used method is the Alternating Least Squares (ALS) method (see [1,4,7]). Kroonenberg and Leeuw [7] used this method for Tucker models [11], and showed that ALS method is robust. Subsequently, Zhang and Golub [17] analyzed the locally linear convergence of ALS method.

Algorithm 1.1 (*ALS method*). Given initial point $\omega^0 = (x^0, y^0, z^0)^T$ such that $\|x^0\| = \|y^0\| = \|z^0\| = 1$.

```
for $p = 0, 1, 2, \ldots,$
    $x_i^{p+1} = \sum_{j,k} a_{ijk} y_j^p z_k^p, \quad i = 1, \ldots, I_1,$
    $y_j^{p+1} = \sum_{i,k} a_{ijk} x_i^{p+1} z_k^p, \quad j = 1, \ldots, I_2,$          (1.5)
    $z_k^{p+1} = \sum_{i,j} a_{ijk} x_i^{p+1} y_j^{p+1}, \quad k = 1, \ldots, I_3.$
    Normalize $\omega^{p+1}$ so that $\|x^{p+1}\| = \|y^{p+1}\| = \|z^{p+1}\| = 1.$
end for
```

ALS method is a generalization of power method for eigenvalue problems. It also can be regarded as a nonlinear version of the Gauss–Seidel method for solving (1.4). Subsequently, a combination of Jacobi version and Gauss–Newton iteration, i.e., Jacobi–Gauss–Newton (JGN) method, was presented by Cardoso and Souloumiac [2].

Algorithm 1.2 (*JGN method*). Given initial point $\omega^0 = (x^0, y^0, z^0)^T$ such that $\|x^0\| = \|y^0\| = \|z^0\| = 1$.

```
for $p = 0, 1, 2, \ldots,$
    $x_i^{p+1} = \sum_{j,k} a_{ijk} y_j^p z_k^p, \quad i = 1, \ldots, I_1,$
    $y_j^{p+1} = \sum_{i,k} a_{ijk} x_i^p z_k^p, \quad j = 1, \ldots, I_2,$          (1.6)
    $z_k^{p+1} = \sum_{i,j} a_{ijk} x_i^p y_j^p, \quad k = 1, \ldots, I_3.$
    Normalize $\omega^{p+1}$ so that $\|x^{p+1}\| = \|y^{p+1}\| = \|z^{p+1}\| = 1.$
end for
```

ALS method and JGN method are not as fast as Newton method in convergence rate. Based on the property of Generalized Rayleigh Quotient (GRQ), Zhang and Golub [17] proposed GRQ-Newton algorithm as follows.

Algorithm 1.3 (*GRQ-Newton method*). Given initial point $\omega^0 = (x^0, y^0, z^0)^T$ such that $\|x^0\| = \|y^0\| = \|z^0\| = 1$.

```
for $p = 0, 1, 2, \ldots$
    • Compute $\lambda^p = GRQ(\omega^p) \overset{\text{def}}{=} \sum_{i,j,k} a_{ijk} x_i^p y_j^p z_k^p.$
    • Solve the following linear equations for $\omega^{p+1}$,
    $J(\lambda^p, \omega^p)\omega^{p+1} = b(\omega^p),$          (1.7)
```

where

$$J(\lambda^p, \omega^p) = \begin{pmatrix} -\lambda I & A_3 & A_2 \\ A_3^T & -\lambda I & A_1 \\ A_2^T & A_1^T & -\lambda I \end{pmatrix}, \quad b(\omega^p) = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}, \tag{1.8}$$

and

$$A_1(j,k) = \sum_i a_{ijk} x_i^p \quad b_1(i) = \sum_{j,k} a_{ijk} y_j^p z_k^p,$$
$$A_2(i,k) = \sum_j a_{ijk} y_j^p \quad b_2(j) = \sum_{i,k} a_{ijk} x_i^p z_k^p,$$
$$A_3(i,j) = \sum_k a_{ijk} z_k^p \quad b_3(k) = \sum_{i,j} a_{ijk} x_i^p y_j^p.$$

- Normalize ω^{p+1} so that $\|x^{p+1}\| = \|y^{p+1}\| = \|z^{p+1}\| = 1$.

end for

Although this algorithm has fast convergence rate, it may break down at some iterate away from the solution, because the iterative matrix $J(\lambda^p, \omega^p)$ may be ill-conditioned or even singular.

For the purpose of global convergence and overcoming ill-condition, we will present a modified Newton's method for the best rank-one approximation of tensor. The rest of this paper is organized as follows. Section 2 introduces the modified Newton's method, and the global convergence analysis is shown in Section 3. The results of numerical experiments are reported in Section 4. Finally, Section 5 gives some conclusions.

2. The modified Newton's method

The line search is the popular global strategy for nonlinear programming [16], which consists of two parts: finding direction and finding steplength. The algorithms mentioned in above section mixed the direction and steplength together. In order to establish our new algorithm, we will investigate how finding directions of these methods. For convenience, we first define two useful operators.

2.1. Two operators

First, we consider the constraint condition $\|x\|_2 = 1$, which is the surface of an unit super sphere. The tangent plane at some x on the surface of unit super sphere is

$$H_x \overset{\text{def}}{=} \{s | x^T s = 1\}.$$

We can define H_y and H_z in the same way. So, we can define a tangent superplane $H_x \times H_y \times H_z$.

For an arbitrary direction δx starting from the point x, we calculate its projection onto the tangent plane H_x by

$$P_x(\delta x) \overset{\text{def}}{=} \delta x - \delta x^T x x = (I - x x^T)\delta x,$$

which is a linear function (see Fig. 1), where $P_x \overset{\text{def}}{=} I - x x^T$. Similarly, we can define P_y and P_z.

If $\delta\omega = (\delta x, \delta y, \delta z)^T$, we can define projective operator $P = \text{diag}(P_x, P_y, P_z)$, then

$$P\delta\omega \overset{\text{def}}{=} \begin{pmatrix} P_x & & \\ & P_y & \\ & & P_z \end{pmatrix} \delta\omega = \begin{pmatrix} P_x(\delta x) \\ P_y(\delta y) \\ P_z(\delta z) \end{pmatrix}. \tag{2.1}$$

Second, in order to fit the constraint condition of (1.3), let us normalize ω. Define

$$U_x(x) \overset{\text{def}}{=} \frac{x}{\|x\|},$$

and define U_y and U_z similarly, then we have

$$U(\omega) \overset{\text{def}}{=} \begin{pmatrix} U_x(x) \\ U_y(y) \\ U_z(z) \end{pmatrix}. \tag{2.2}$$

So, we have the following property.

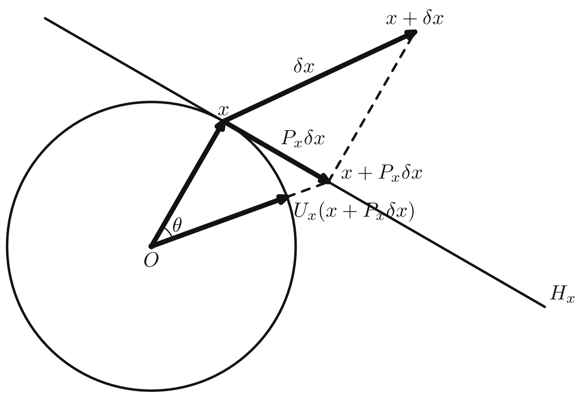

Fig. 1. Projection.

Lemma 2.1.

$$\|U(\omega+P\delta\omega)-U(\omega)\| \leqslant \|P\delta\omega\|.$$

Proof. We only need to prove

$$\|U_x(x+P_x\delta x)-U_x(x)\| \leqslant \|P_x\delta x\|.$$

Similarly, we can get the conclusion about y and z. Because $(P_x\delta x)^\top x = 0$ (see Fig. 1), define the angle between x and $x+P_x\delta x$ as θ. Then

$$\|U_x(x+P_x\delta x)-U_x(x)\| = 2\sin\frac{\theta}{2} \leqslant \theta \leqslant \tan\theta = \|P_x\delta x\|. \qquad \square$$

Therefore, we can use the norm of the projection $P\delta\omega$ as a stopping tolerance.

2.2. Directions

In order to study the property of the directions, we denote

$$x_i^{p+1} = x_i^p + \delta x_i^p, \quad y_j^{p+1} = y_j^p + \delta y_j^p, \quad z_k^{p+1} = z_k^p + \delta z_k^p. \tag{2.3}$$

Based on (1.6) and normalization, we can add a constant λ in (1.6). So, (1.6) can be written as

$$\begin{aligned} \lambda x_i^{p+1} &= \sum_{j,k} a_{ijk} y_j^p z_k^p, \quad i = 1,\ldots,I_1,\\ \lambda y_j^{p+1} &= \sum_{i,k} a_{ijk} x_i^p z_k^p, \quad j = 1,\ldots,I_2,\\ \lambda z_k^{p+1} &= \sum_{i,j} a_{ijk} x_i^p y_j^p, \quad k = 1,\ldots,I_3. \end{aligned} \tag{2.4}$$

Substituting (2.3) into (2.4) yields

$$\begin{cases} -\lambda\delta x_i^p = \lambda x_i^p - \sum_{j,k} a_{ijk} y_j^p z_k^p, & i = 1,\ldots,I_1,\\ -\lambda\delta y_j^p = \lambda y_j^p - \sum_{i,k} a_{ijk} x_i^p z_k^p, & j = 1,\ldots,I_2,\\ -\lambda\delta z_k^p = \lambda z_k^p - \sum_{i,j} a_{ijk} x_i^p, y_j^p, & k = 1,\ldots,I_3, \end{cases} \tag{2.5}$$

or, in the matrix form, where the index 'p' is ignored,

$$\begin{pmatrix} -\lambda I & & \\ & -\lambda I & \\ & & -\lambda I \end{pmatrix} \begin{pmatrix} \delta x \\ \delta y \\ \delta z \end{pmatrix} = \lambda \begin{pmatrix} x \\ y \\ z \end{pmatrix} - \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}, \tag{2.6}$$

that is, $-\lambda I\delta\omega = \lambda\omega - b(\omega)$. So, we get the direction of Jacobi–Gauss–Newton method

$$\delta\omega^{\mathrm{JGN}} = \frac{1}{\lambda}b(\omega) - \omega. \tag{2.7}$$

The following theorem indicates that this direction is the projection of $b(\omega)$, which is the steepest ascent direction of the objective function of dual program (1.3), onto the tangent superplane $H_x \times H_y \times H_z$.

Theorem 2.1. *Denote $\delta\omega^{\mathrm{JGN}}$ as the direction of Jacobi–Gauss–Newton method, then*

$$Pb(\omega) = \lambda\delta\omega^{\mathrm{JGN}}. \tag{2.8}$$

Proof. We only need to prove $P_x b_1 = \lambda\delta x^{\mathrm{JGN}}$. In fact,

$$P_x b_1 = b_1 - b_1^T x x = b_1 - \lambda x = \lambda\delta x^{\mathrm{JGN}}.$$

Similarly, we can obtain the conclusions about y and z, and hence (2.8). □

Next, we consider the directions of the GRQ-Newton method. For certain λ, we rewrite the linearization of Eqs. (1.4) as

$$\begin{cases} \sum_{j,k} a_{ijk}(y_j\delta z_k + \delta y_j z_k) - \lambda\delta x_i = \lambda x_i - \sum_{j,k} a_{ijk}y_j z_k, \\ \sum_{i,k} a_{ijk}(x_i\delta z_k + \delta x_i z_k) - \lambda\delta y_j = \lambda y_j - \sum_{i,k} a_{ijk}x_i z_k, \\ \sum_{i,j} a_{ijk}(x_i\delta y_j + \delta x_i y_j) - \lambda\delta z_k = \lambda z_k - \sum_{i,j} a_{ijk}x_i y_j. \end{cases}$$

Then, the direction $\delta\omega^{Newton}$ of GRQ-Newton method is the solution of the following equation

$$\begin{pmatrix} -\lambda I & A_3 & A_2 \\ A_3^T & -\lambda I & A_1 \\ A_2^T & A_1^T & -\lambda I \end{pmatrix}\begin{pmatrix} \delta x \\ \delta y \\ \delta z \end{pmatrix} = \lambda\begin{pmatrix} x \\ y \\ z \end{pmatrix} - \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}. \tag{2.9}$$

When (λ,ω) is far away from the solution (λ^*,ω^*), the matrix $J(\lambda,\omega)$ may be ill-conditioned or even singular. So, we cannot insure the direction of GRQ-Newton method being an increasing direction.

Note that $-\lambda I$ is the iterative matrix of the Jacobi–Gauss–Newton method and it is well-conditioned. Therefore, we can combine the two iterative equations and get new directions. Let $\gamma \in [0,1]$, then $(1-\gamma)\times$ (2.6) $+\gamma\times$ (2.9) gives

$$\begin{pmatrix} -\lambda I & \gamma A_3 & \gamma A_2 \\ \gamma A_3^T & -\lambda I & \gamma A_1 \\ \gamma A_2^T & \gamma A_1^T & -\lambda I \end{pmatrix}\delta\omega = \lambda\omega - b(\omega), \tag{2.10}$$

which is defined as

$$\bar{J}_\gamma(\lambda,\omega)\delta\omega = \lambda\omega - b(\omega). \tag{2.11}$$

The iterative matrix $\bar{J}_\gamma(\lambda,\omega)$ is symmetric. When γ tends to 1, the iterative matrix $\bar{J}_\gamma(\lambda,\omega)$ tends to $J(\lambda,\omega)$, and the direction $\delta\omega$ approaches to the direction of GRQ-Newton method which enjoys the fast local quadratic convergence rate, but the iterative matrix may be ill-conditioned or singular, or the direction may be not an ascent direction. When γ tends to 0, the iterative matrix $\bar{J}_\gamma(\lambda,\omega)$ tends to a well-conditioned negative definite matrix $-\lambda I$, and the direction approaches to the projective direction of the steepest ascent direction, but it loses the fast local convergence rate. Therefore, our idea is to balance these two methods by choosing a proper parameter γ such that the iterative equation is easy to be solved and the direction $\delta\omega$ is a good ascent direction.

2.3. Symmetric modified Newton's algorithm

Our idea is that, we try the (modified) Newton direction with steplength 1 at the beginning. If this step makes the objective value of the dual problem increasing, we accept this step. Otherwise, we perform a line search along the projective direction of the steepest ascent direction, i.e., the Jacobi–Gauss–Newton direction, to guarantee the algorithm converges globally.

Algorithm 2.1 (Symmetric Modified Newton's (SMN) algorithm).

Initialization. Given $c \in (0,\frac{1}{2}), \beta \in (0,1), \rho \in (0,1), M$ a positive integer. An initial point $\omega^0 = [x^0, y^0, z^0]^T$ is given such that $\omega^0 = U(\omega^0)$ and $\lambda^0 = GRQ(\omega^0) > 0$.

for $p = 0,1,2,\ldots,$

- *Try the (modified) Newton step.*Choose a largest $\gamma \in \{1,\rho,\rho^2,\ldots,\rho^M\}$ to ensure that not only the following equation

$$\bar{J}_\gamma(\lambda^p,\omega^p)\delta\omega = \lambda^p\omega^p - b(\omega^p) \tag{2.12}$$

is solvable, but also the following inequality

$$\lambda^p_{\text{New}} := \text{GRQ}(U(\omega^p + \delta\omega^p)) \geqslant \lambda^p + \frac{c}{\lambda^p}\|Pb(\omega^p)\|^2 \tag{2.13}$$

is satisfied. If this γ exists, then let $\alpha^p = 1\omega^{p+1} = U(\omega^p + \delta\omega^p), \lambda^{p+1} = \lambda^p_{\text{New}}$, and go to next loop. Otherwise, go to next line search step.

- *Line search step.*Let $\delta\omega^p = \delta\omega^{p^{\text{JGN}}} = \frac{1}{\lambda^p} b(\omega^p) - \omega^p$. Seek a steplength $\alpha^p = \beta^m$, where m is a smallest nonnegative integer such that

$$\lambda^p_{\text{New}} := \text{GRQ}(U(\omega^p + \alpha^p\delta\omega^p)) \geqslant \lambda^p + \frac{c\alpha^p}{\lambda^p}\|Pb(\omega^p)\|^2. \tag{2.14}$$

Let $\omega^{p+1} = U(\omega^p + \alpha_p\delta\omega^p), \lambda^{p+1} = \lambda^p_{New}$, go to next loop.

end for

In this algorithm, we need to prove that the process of the line search will terminate finitely.

Theorem 2.2. *If α is sufficiently small, then (2.14) must be satisfied.*

Proof. See Fig. 2, when $\alpha \to 0$, then $\theta \to 0$. Since $2\sin\frac{\theta}{2} = \tan\theta + o(\theta)$, then

$$U(\omega^p + \alpha\delta\omega^p) - \omega^p = 2\sin\frac{\theta}{2} = \tan\theta + o(\theta) = \alpha\delta\omega^p + o(\alpha).$$

Since $\delta\omega^p = \frac{1}{\lambda^p} b(\omega^p) - \omega^p = P\delta\omega^p$ and $b(\omega^p)$ is bounded, we have

$$\begin{aligned}\text{GRQ}(U(\omega^p + \alpha\delta\omega^p)) - \text{GRQ}(\omega^p) &= \text{GRQ}(\omega^p + \alpha\delta\omega^p + o(\alpha)) - \lambda^p = \alpha b(\omega^p)^T\delta\omega^p + o(\alpha) = \frac{\alpha}{\lambda^p} b(\omega^p)^T Pb(\omega^p) + o(\alpha)\\ &= \frac{\alpha}{\lambda^p}\|Pb(\omega^p)\|^2 + o(\alpha).\end{aligned}$$

When $0 < c < \frac{1}{2}$ and α is small enough, the line search rule (2.14) must hold. □

3. Global convergence analysis

In this section we give the global convergence analysis of the symmetric modified Newton's algorithm.

Because the feasible region of the dual problem (1.3) is a compact set $\{\omega = (x, y, z)^T, |\|x\| = \|y\| = \|z\| = 1\}$, its optimal solution must exist. We denote it by ω^* with objective value $\lambda^* = \text{GRQ}(\omega^*)$.

Lemma 3.1. *When the algorithm produces an infinite sequence, we have*

$$\lim_{p\to\infty} \alpha^p\|Pb(\omega^p)\|^2 = 0. \tag{3.1}$$

Proof. From the algorithm we know

$$\lambda^* - \lambda^0 \geqslant \sum_{p=0}^{\infty}(\lambda^{p+1} - \lambda^p) \geqslant \sum_{p=0}^{\infty} \frac{c\alpha^p}{\lambda^p}\|Pb(\omega^p)\|^2,$$

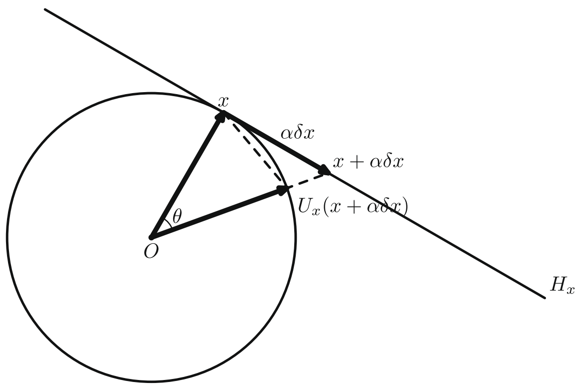

Fig. 2. New iteration.

thus the series on the right is convergent, that means

$$\lim_{p\to\infty} \frac{c\alpha^p}{\lambda^p}\|Pb(\omega^p)\|^2 = 0.$$

Since c is a constant and $\lambda^* \geqslant \lambda^p \geqslant \lambda^0 > 0$, the conclusion is obtained. □

Lemma 3.2. *Assume that $\|Pb(\omega^p)\| \geqslant \varepsilon > 0$. Assume the condition of Lemma 3.1 holds. Then the steplength α^p is bound below away from zero, that is,*

$$\liminf_{p\to\infty} \alpha^p > 0. \tag{3.2}$$

Proof. Suppose that the conclusion is not true, then there exit $\beta > 0$ and a subsequence $\{\alpha^p | \alpha^p \leqslant \beta, \alpha^p \to 0\}$. These α^p are stepsize in the algorithm, and these $\frac{\alpha^p}{\beta}$ do not satisfy (2.14), i.e.,

$$\mathrm{GRQ}(U(\omega^p + \frac{\alpha^p}{\beta}\delta\omega^p)) - \lambda^p < \frac{c\alpha^p}{\beta\lambda^p}\|Pb(\omega^p)\|^2.$$

When α^p is close enough to 0, the left-hand side of above inequality is

$$\mathrm{GRQ}(U(\omega^p + \frac{\alpha^p}{\beta}\delta\omega^p)) - \lambda^p = \mathrm{GRQ}(\omega^p + \frac{\alpha^p}{\beta}\delta\omega^p + o(\alpha^p)) - \lambda^p = \frac{\alpha^p}{\beta}b(\omega^p)^T\delta\omega^p + o(\alpha^p) = \frac{\alpha^p}{\beta\lambda^p}b(\omega^p)^T Pb(\omega^p) + o(\alpha^p)$$
$$= \frac{\alpha^p}{\beta\lambda^p}\|Pb(\omega^p)\|^2 + o(\alpha^p).$$

Therefore, we have

$$\frac{\alpha^p}{\beta\lambda^p}\|Pb(\omega^p)\|^2 + o(\alpha^p) < \frac{c\alpha^p}{\beta\lambda^p}\|Pb(\omega^p)\|^2.$$

Dividing $\frac{2\alpha^p}{\lambda^p}\|Pb(\omega^p)\|^2$on both sides, and letting $\alpha^p \to 0$ yield $1 \leqslant c$ which contradicts the assumption $0 < c < \frac{1}{2}$. Therefore the conclusion is obtained. □

Theorem 3.1. *Under the assumptions of Lemma 3.2, we have*

$$\liminf_{p\to\infty} \|Pb(\omega^p)\| = 0. \tag{3.3}$$

Proof. It is a direct consequence of Lemmas 3.1 and 3.2. □

From the above analysis, there exists a limit point (λ^*, ω^*) of iterative sequence $\{(\lambda, \omega)\}$ such that

$$b(\omega^*) - \lambda^*\omega^* = Pb(\omega^*) = 0.$$

This means (λ^*, ω^*) is a solution of the nonlinear Eq. (1.4), which is a Karush–Kuhn–Tucker point of the dual program (1.3).

4. Unsymmetric modified Newton's algorithm

The skill of the symmetric modified Newton's algorithm can be used to get the unsymmetric one. The main idea is combining the iterative equations of the GRQ-Newton method with the ALS method. We introduce this method by only considering the direction, but algorithm depiction is omitted here, since it is similar to Algorithm 2.1.

We write the linearization form of iterative Eq. (1.5) of the ALS method in the following form:

$$\begin{cases} \sum_{j,k} a_{ijk}y_j^p z_k^p = \lambda(x_i^p + \delta x_i), \\ \sum_{i,k} a_{ijk}(x_i^p + \delta x_i)z_k^p = \lambda(y_j^p + \delta y_j), \\ \sum_{i,j} a_{ijk}(x_i^p y_j^p + x_i^p\delta y_j + \delta x_i y_j^p) = \lambda(z_k^p + \delta z_k). \end{cases} \tag{4.1}$$

Rewrite it in the matrix form:

$$\begin{pmatrix} -\lambda I & 0 & 0 \\ A_3^T & -\lambda I & 0 \\ A_2^T & A_1^T & -\lambda I \end{pmatrix}\begin{pmatrix} \delta x \\ \delta y \\ \delta z \end{pmatrix} = \lambda\begin{pmatrix} x \\ y \\ z \end{pmatrix} - \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}. \tag{4.2}$$

Table 1
Total iterations.

Algorithm	SMN	UMN	ALS	JGN
Number of Iterations	67,896	40,767	185,867	469,541
Ratios to ALS (%)	36.53	21.93	100	252.62

Table 2
Total CPU time (s).

Algorithm	SMN	UMN	ALS	JGN
CPU time	426.409	264.521	569.324	1458.180
Ratios to ALS (%)	74.90	46.46	100	256.12

Table 3
Steps.

Algorithm	Modified Newton Steps (γ)						Line search step
	1	0.25	0.25^2	0.25^3	0.25^4	0.25^5	
SMN	9696	57005	1051	100	15	20	9
	14.28%	83.96%	1.55%	0.15%	0.02%	0.03%	0.01%
UMN	9661	24930	4893	700	404	114	65
	23.70%	61.15%	12.00%	1.72%	0.99%	0.28%	0.16%

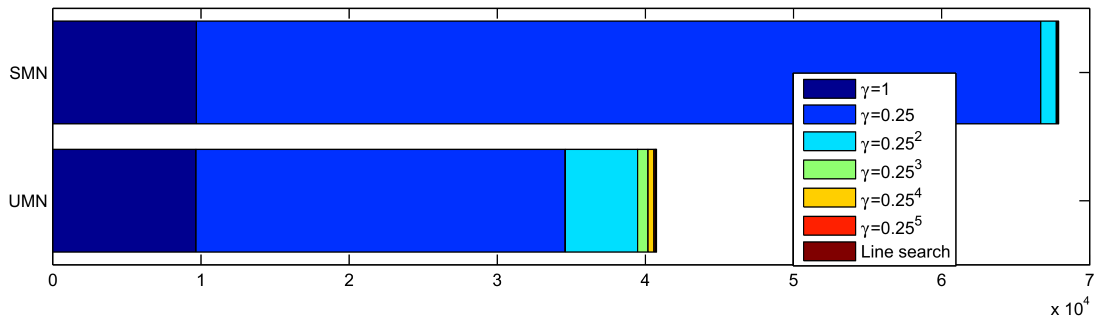

Fig. 3. Iterations.

With the same technique as above, we combine it with the iterative equation of the GRQ-Newton method, and $(1-\gamma)\times$ (4.2) $+\gamma\times$ (2.9) yields

$$\begin{pmatrix} -\lambda I & \gamma A_3 & \gamma A_2 \\ A_3^T & -\lambda I & \gamma A_1 \\ A_2^T & A_1^T & -\lambda I \end{pmatrix} \delta\omega = \lambda\omega - b(\omega). \tag{4.3}$$

Let us denote this iteration as

$$\widehat{J}_\gamma(\lambda,\omega)\delta\omega = \lambda\omega - b(\omega). \tag{4.4}$$

We can see that $\widehat{J}_\gamma$ is unsymmetric. When γ tends to $1, \widehat{J}_\gamma(\lambda,\omega)$ tends to $J(\lambda,\omega)$ and $\delta\omega$ tends to the direction of the GRQ-Newton method. When γ tends to $0, \delta\omega$ tends to the direction of the ALS method.

Note that an unsymmetric modified Newton's (UMN) method can be obtained by slightly changing the linear Eqs. (2.12) of the symmetric modified Newton's (SMN) method, i.e., by deleting the factor γ of the strict lower triangular part in (2.10). Obviously, the coefficient matrix in (4.3) is unsymmetric. Since the convergence analysis is also similar to that in Section 3, we omit it.

5. Numerical experiments

Now we describe the implementation of our algorithms in MATLAB codes. We set the following parameters

$$\rho = 0.25, \quad M = 5, \quad c = 0.0001, \quad \beta = 0.5.$$

We stop the algorithm if one of the following three criterions is satisfied,

$$\|Pb(\omega^p)\| \leqslant 10^{-6}, \quad \|\omega^p - \omega^{p-1}\| \leqslant 10^{-8}, \quad |\lambda^p - \lambda^{p-1}| \leqslant 10^{-10}.$$

If the number of iterations exceeds 1000, the algorithm stops too.

Four kinds of algorithms are compared.

- **SMN**: Symmetric Modified Newton's algorithm, see Algorithm 2.1;
- **UMN**: Unsymmetric Modified Newton's algorithm;
- **ALS**: Alternating Least Squares algorithm, see Algorithm 1.1;
- **JGN**: Jacobi–Gauss–Newton algorithm, see Algorithm 1.2.

Now, we report the numerical results. We consider a random $10 \times 20 \times 30$ tensor whose elements are generated by uniformly distributing in $[-100, 100]$. We test 20,000 problems with 3rd-order random tensor. There are 4848 problems which fail, 13,890 problems for that the four algorithms generate different optimal objective values, and 1262 problems for that the four algorithms generate the same objective values.

Tables 1 and 2 show the total number of iterations, CPU time of these algorithms, and the ratio to popular ALS algorithm. Obviously, our new modified algorithms SMN and UMN are faster and more efficient than algorithm JGN and ALS. The algorithm SMN saves about 25% CPU time than the algorithm ALS, while the UMN saves about 50%.

Why do our algorithms SMN and UMN have very good behavior?

In Table 3 and Fig. 3, we give the number of modified Newton's step and their percentage when the parameter γ takes different values, as well as the number of line search step. We find that the number of modified Newton's step is 9696 and takes 14.28% when $\gamma = 1$; the number of modified Newton's step is 57,005 and takes 83.96% when $\gamma = 0.25$. This implies that a majority of iterations are generated by Newton's step ($\gamma = 1$) or modified Newton's step ($\gamma = 0.25$). At the same time, our algorithms avoid the ill-conditioned cases. So our modified methods are successful.

6. Conclusions

In this paper, we discussed a modified Newton's method for best rank-one approximation to tensor. In order to speed up the convergence rate of the algorithm and overcome the ill-condition and singularity, we combine, respectively, the iterative matrix of JGN and ALS methods with the iterative matrix of GRQ-Newton method, and proposed an adaptive modified version of GRQ-Newton method with parameter γ. Preliminary numerical experiments show that our algorithms work efficiently and save the CPU time significantly.

Acknowledgement

This work was supported by National Natural Science Foundation of China (Grant No. 10871098) and the Natural Science Fund of Jiangsu Province of China at Grant No. BK2006214.

References

[1] J.T. Berge, J.D. Leeuw, P. Kroonenberg, Some additional results and principal components analysis of three-mode data by means of alternating least squares algorithms, Psychometrika 52 (1987) 183–191.
[2] J.F. Cardoso, A. Souloumiac, Jacobi angles for simultaneous diagonalization, SIAM J. Matrix Anal. Appl. 17 (1996) 161–164.
[3] J.D. Carroll, J. Chang, Analysis of individual differences in multidimensional scaling via an N-way genaralization of Eckart–Yong decomposition, Psychometrika 35 (1970) 283–319.
[4] H. Kiers, An alternating least squares algorithm for PARAFAC2 and three-way DEDICOM, Comput. Statist. Data Anal. 16 (1993) 103–118.
[5] P.M. Kroonenberg, Three-mode Principal Component Analysis, DSWO Press, Leiden, The Netherlands, 1983.
[6] P.M. Kroonenberg, Three-mode principal component analysis: illustrated with an example from attachment theory, in: H.G. Law, C.W. Synder, J.A. Hattie, R.P. McDonald (Eds.), Research Methods for Multimode Data Analysis, Praeger, New York, 1984, pp. 64–103.
[7] P.M. Kroonenberg, J.D. Leeuw, Principal component analysis of three-mode data by means of alternating least squares algorithms, Psychometrika 45 (1980) 69–97.
[8] L.D. Lathauwer, P. Comon, B.D. Moor, et al, High-order power method – application in independent component analysis, in: Proceedings of the International Symposium on Nonlinear Theory and its Applications (NOLTA'95), Las Vegas, NV, 1995, pp. 91–96.
[9] L.D. Lathauwer, B.D. Moor, J. Vandewalle, On the best rank-1 and rank-$(R_1, R_2, \ldots, R_N)$ approximation of higher-order tensors, SIAM J. Matrix Anal. Appl. 21 (2000) 1324–1342.
[10] L.D. Lathauwer, B.D. Moor, J. Vandewalle, A multilinear singular value decomposition, SIAM J. Matrix Anal. Appl. 21 (2000) 1253–1378.
[11] L.R. Tucker, Some mathematical notes on three mode factor analysis, Psychometrika 31 (1966) 279–311.
[12] L. Qi, W. Sun, Y. Wang, Numerical multilinear algebra and its applications, Front. Math. China 2 (2007) 501–526.
[13] L. Qi, Eigenvalues of a real supersymmetric tensor, J. Symbo. Comput. 40 (2005) 1302–1324.
[14] L. Qi, Rank and eigenvalues of a supersymmetric tensor, a multivariate homogeneous polynomial and an algebraic surface defined by them, J. Symb. Comput. 41 (2006) 1309–1327.
[15] L. Qi, Eigenvalues and invariants of tensors, J. Math. Anal. Appl. 325 (2007) 1363–1377.
[16] W. Sun, Y. Yuan, Optimization Theory and Methods: Nonlinear Programming, Springer, 2006.
[17] T. Zhang, G.H. Golub, Rank-one approximation to higher order tensors, SIAM J. Matrix Anal. Appl. 23 (2001) 534–550.