Two-stage acceleration for non-linear PCA

Masahiro Kuroda, *Okayama University of Science*, kuroda@soci.ous.ac.jp
Michio Sakakihara, *Okayama University of Science*, sakaki@mis.ous.ac.jp
Yuichi Mori, *Okayama University of Science*, mori@soci.ous.ac.jp
Masaya Iizuka, *Okayama University*, iizuka@ems.okayama-u.ac.jp

Abstract. Principal components analysis (PCA) is a descriptive multivariate method for analyzing quantitative data. For PCA of a mixture of quantitative and qualitative data, quantification of qualitative data requires obtaining optimal scaling data and using ordinary PCA. The extended PCA, including such quantification, is called non-linear PCA. Then, the alternating least squares (ALS) algorithm is used as the quantification method. However, the ALS algorithm for non-linear PCA of large data requires many iterations and much computation time due to its linear convergence. We provide a new acceleration method for the ALS algorithm using the vector ε (vε) and Graves-Morris (GM) algorithms. Both acceleration algorithms speed up the convergence of a linearly convergent sequence generated by the ALS algorithm. Acceleration of the ALS algorithm can be performed in two stages: 1) the vε algorithm generates an accelerated sequence of the ALS sequence and 2) the convergence of the vε accelerated sequence is accelerated using the GM algorithm. Thus, we expect that, by accelerating the convergence of the vε accelerated sequence, the GM algorithm improves the overall computational efficiency of the ALS algorithm. Numerical experiments examine the performance of the two-stage acceleration for non-linear PCA.

Keywords. non-linear PCA, alternating least squares algorithm, acceleration of convergence, vector ε algorithm, Graves-Morris algorithm

1 Introduction

Principal components analysis (PCA) is a descriptive multivariate method commonly used for analyzing quantitative data. For PCA of a mixture of quantitative and qualitative data, quantification of qualitative data requires obtaining optimal scaling data and using ordinary PCA. The extended PCA, including such quantification, is called *non-linear PCA*, see Gifi [4]. The existing algorithms for non-linear PCA are PRINCIPALS of Young et al. [9] and PRINCALS of Gifi [4], in which the alternating least squares (ALS) algorithm is utilized. This algorithm

alternates between quantification of qualitative data for optimal scaling and computation of ordinary PCA of optimal scaling data.

Application of non-linear PCA to very large data sets and variable selection problems requires numerous iterations and large computation time for convergence of the ALS algorithm, because the algorithm's convergence speed is linear. For example, in PCA based on a subset of variables for qualitative data of Mori et al. [7], the ALS algorithm requires a much larger number of iterations and longer computation time to search for a reasonable subset. For an iterative algorithm that generates a linearly convergent sequence such as the ALS algorithm, there exist several convergence acceleration algorithms, see Brezinski and Zaglia [3]. Kuroda et al. [6] proposed an acceleration algorithm for the convergence of the ALS sequence using the vector ε (vε) algorithm of Wynn [10]. During iterations of the vε accelerated ALS algorithm, the vε algorithm generates an accelerated sequence of optimal scaling data estimated by the ALS algorithm. Numerical experiments demonstrated that the vε acceleration greatly speeds up the convergence of the ALS sequence of the estimated optimal scaling data.

In this paper, we provide a new acceleration method for the ALS algorithm using the vε and Graves-Morris (GM) algorithms [5]. Both algorithms accelerate the convergence of a linearly convergent sequence. Acceleration of the ALS algorithm can be performed in two stages: the first stage using the vε algorithm generates an accelerated sequence of the ALS sequence and the second stage accelerates the convergence of the vε accelerated sequence using the GM algorithm.

The paper is organized as follows. We briefly describe the ALS algorithm for non-linear PCA and introduce the vε acceleration for the ALS algorithm proposed by Kuroda et al. [6] in Section 2. Section 3 gives the GM algorithm and its convergence properties. The two-stage acceleration for the ALS algorithm is described in Section 4. Numerical experiments in Section 5 examine the performance and properties of the two-stage acceleration for PRINCIPALS. In Section 6, we present our concluding remarks.

2 ALS algorithm for non-linear PCA and vε acceleration

Computation of non-linear PCA

Let $\mathbf{X} = (\mathbf{X}_1\ \mathbf{X}_2\ \cdots\ \mathbf{X}_p)$ be an $n \times p$ matrix of n observations on p quantitative variables and be columnwise standardized. In PCA, $\mathbf{X}$ is linearly transformed into a substantially smaller set of uncorrelated variables and are approximated by the following bilinear form:

$$\hat{\mathbf{X}} = \mathbf{Z}\mathbf{A}^\top, \tag{1}$$

where $\mathbf{Z} = (\mathbf{Z}_1\ \mathbf{Z}_2\ \cdots\ \mathbf{Z}_r)$ is an $n \times r$ matrix of n component scores on r ($1 \leq r \leq p$) components, and $\mathbf{A} = (\mathbf{A}_1\ \mathbf{A}_2\ \cdots\ \mathbf{A}_r)$ is a $p \times r$ matrix consisting of the eigenvectors of $\mathbf{X}^\top\mathbf{X}/n$ and $\mathbf{A}^\top\mathbf{A} = \mathbf{I}_r$. Then, we determine model parameters $\mathbf{Z}$ and $\mathbf{A}$ such that

$$\theta = \mathrm{tr}(\mathbf{X} - \hat{\mathbf{X}})^\top(\mathbf{X} - \hat{\mathbf{X}}) = \mathrm{tr}(\mathbf{X} - \mathbf{Z}\mathbf{A}^\top)^\top(\mathbf{X} - \mathbf{Z}\mathbf{A}^\top) \tag{2}$$

is minimized for the prescribed r components.

For no-linear PCA with optimal scaling, quantification of qualitative data requires obtaining optimal scaling data and using ordinary PCA. To quantify $\mathbf{X}_j$ of qualitative variable j with K_j categories, the vector is coded using an $n \times K_j$ indicator matrix $\mathbf{G}_j$ with entries $g_{(j)ik} = 1$ if object i belongs to category k, and $g_{(j)ik'} = 0$ if object i belongs to some other category

$k'(\neq k)$, $i = 1, \ldots, n$ and $k = 1, \ldots, K_j$. Then, the optimally scaled vector $\mathbf{X}_j^*$ of $\mathbf{X}_j$ is given by $\mathbf{X}_j^* = \mathbf{G}_j \alpha_j$, where α_j is a $K_j \times 1$ score vector for categories of $\mathbf{X}_j$. Let $\mathbf{X}^* = (\mathbf{X}_1^* \ \mathbf{X}_2^* \ \cdots \ \mathbf{X}_p^*)$ be an $n \times p$ matrix of optimally scaled observations to satisfy restrictions

$$\mathbf{X}^{*\top} \mathbf{1}_n = \mathbf{0}_p \qquad \text{and} \qquad \operatorname{diag}\left[\frac{\mathbf{X}^{*\top}\mathbf{X}^*}{n}\right] = \mathbf{I}_p, \tag{3}$$

where $\mathbf{1}_n$ and $\mathbf{0}_p$ are vectors of ones and zeros of length n and p, respectively. In the presence of qualitative variables, the optimization criterion (2) is replaced by

$$\theta^* = \operatorname{tr}(\mathbf{X}^* - \hat{\mathbf{X}})^\top (\mathbf{X}^* - \hat{\mathbf{X}}) = \operatorname{tr}(\mathbf{X}^* - \mathbf{Z}\mathbf{A}^\top)^\top (\mathbf{X}^* - \mathbf{Z}\mathbf{A}^\top). \tag{4}$$

In non-linear PCA, we determine the optimal scaling parameter $\mathbf{X}^*$, in addition to estimating $\mathbf{Z}$ and $\mathbf{A}$.

ALS algorithm for non-linear PCA: PRINCIPALS of Young et al.

The ALS algorithm for non-linear PCA alternates between ordinary PCA and optimal scaling, and minimizes θ^* of Equation (4) under restriction (3). Then, θ^* is to be determined by model parameters $\mathbf{Z}$ and $\mathbf{A}$ and optimal scaling parameter $\mathbf{X}^*$, by updating each of the parameters in turn, keeping the others fixed. We use PRINCIPALS of Young et al. [9] as the ALS algorithm for non-linear PCA.

For the initialization, the observed data $\mathbf{X}$ may be used as initial data $\mathbf{X}^{*(0)}$ after it is standardized to satisfy restriction (3). For given initial data $\mathbf{X}^{*(0)}$, PRINCIPALS iterates the following steps:

Algorithm 2.1 (PRINCIPALS)**.**

Step1 Model parameter estimation step: *Obtain* $\mathbf{A}^{(t)}$ *by solving*

$$\left[\frac{\mathbf{X}^{*(t)\top}\mathbf{X}^{*(t)}}{n}\right] \mathbf{A} = \mathbf{A}\mathbf{D}_r, \tag{5}$$

where $\mathbf{A}^\top\mathbf{A} = \mathbf{I}_r$ *and* $\mathbf{D}_r$ *is an* $r \times r$ *diagonal matrix of eigenvalues, and the superscript* (t) *indicates the* t*-th iteration. Compute* $\mathbf{Z}^{(t)}$ *from* $\mathbf{Z}^{(t)} = \mathbf{X}^{*(t)}\mathbf{A}^{(t)}$*.*

Step2 Optimal scaling step: *Calculate* $\hat{\mathbf{X}}^{(t+1)} = \mathbf{Z}^{(t)}\mathbf{A}^{(t)\top}$ *from Equation (1). Find* $\mathbf{X}^{*(t+1)}$ *such that*

$$\mathbf{X}^{*(t+1)} = \underset{\mathbf{X}^*}{arg\min}\, tr(\mathbf{X}^* - \hat{\mathbf{X}}^{(t+1)})^\top (\mathbf{X}^* - \hat{\mathbf{X}}^{(t+1)})$$

for fixed $\hat{\mathbf{X}}^{(t+1)}$ *under measurement restrictions on each of the variables. Scale* $\mathbf{X}^{*(t+1)}$ *by columnwise centering and normalizing.*

vε acceleration for PRINCIPALS

The vε algorithm of Wynn [10] is a method for accelerating the convergence of a slowly convergent vector sequence and works effective for linearly convergent sequences. It is known that the maximum speed of convergence of the vε algorithm is superlinear.

Let $\{\mathbf{Y}^{(t)}\}_{t\geq 0} = \{\mathbf{Y}^{(0)}, \mathbf{Y}^{(1)}, \mathbf{Y}^{(2)}, \dots\}$ be a linearly convergent sequence generated by an iterative computational procedure and let $\{\dot{\mathbf{Y}}^{(t)}\}_{t\geq 0} = \{\dot{\mathbf{Y}}^{(0)}, \dot{\mathbf{Y}}^{(1)}, \dot{\mathbf{Y}}^{(2)}, \dots\}$ be the accelerated sequence of $\{\mathbf{Y}^{(t)}\}_{t\geq 0}$. We denote $\triangle\mathbf{Y}^{(t)} = \mathbf{Y}^{(t+1)} - \mathbf{Y}^{(t)}$ and define the inverse of vector $\mathbf{Y}$ by $[\mathbf{Y}]^{-1} = \mathbf{Y}/\langle \mathbf{Y}, \mathbf{Y}\rangle$, where $\langle\cdot,\cdot\rangle$ is the inner product of vectors. Then, the vε algorithm generates $\{\dot{\mathbf{Y}}^{(t)}\}_{t\geq 0}$ by using

$$\dot{\mathbf{Y}}^{(t-1)} = \mathbf{Y}^{(t)} + \left[\left[\triangle\mathbf{Y}^{(t)}\right]^{-1} - \left[\triangle\mathbf{Y}^{(t-1)}\right]^{-1}\right]^{-1}. \tag{6}$$

When $\{\mathbf{Y}^{(t)}\}_{t\geq 0}$ converges to a limit point $\mathbf{Y}^{(\infty)}$ of $\{\mathbf{Y}^{(t)}\}_{t\geq 0}$, it is known that, in many cases, $\{\dot{\mathbf{Y}}^{(t)}\}_{t\geq 0}$ generated by the vε algorithm converges to $\mathbf{Y}^{(\infty)}$ faster than $\{\mathbf{Y}^{(t)}\}_{t\geq 0}$.

Kuroda et al. [6] proposed the vε acceleration for the ALS algorithm for non-linear PCA. The vε acceleration algorithm speeds up the convergence of the ALS sequence. Numerical experiments demonstrated that its speed of convergence is significantly faster than that of the ALS algorithm. We assume that $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$ generated by PRINCIPALS converges to a limit point $\mathbf{X}^{*(\infty)}$. Then, vε accelerated PRINCIPALS (vε-PRINCIPALS) produces a fast convergent sequence $\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$ of $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$. The procedure of v$\varepsilon$-PRINCIPALS of Kuroda et al. [6] iterates the following two steps:

Algorithm 2.2 (vε-PRINCIPALS)**.**

Step1 PRINCIPALS step: *Compute model parameters* $\mathbf{A}^{(t)}$ *and* $\mathbf{Z}^{(t)}$ *and determine optimal scaling parameter* $\mathbf{X}^{*(t+1)}$*.*

Step2 vε acceleration step: *Calculate* $\dot{\mathbf{X}}^{*(t-1)}$ *using* $\{\mathbf{X}^{*(t-1)}, \mathbf{X}^{*(t)}, \mathbf{X}^{*(t+1)}\}$ *from the vε algorithm:*

$$vec\dot{\mathbf{X}}^{*(t-1)} = vec\mathbf{X}^{*(t)} + \left[\left[\triangle vec\mathbf{X}^{*(t)}\right]^{-1} - \left[\triangle vec\mathbf{X}^{*(t-1)}\right]^{-1}\right]^{-1},$$

where $vec\mathbf{X}^* = (\mathbf{X}_1^{*\top}\ \mathbf{X}_2^{*\top}\cdots\ \mathbf{X}_p^{*\top})^\top$ *and check the convergence by*

$$\left\|\triangle vec\dot{\mathbf{X}}^{*(t-2)}\right\|^2 < \delta,$$

where δ *is a desired accuracy.*

Before starting of the iteration of vε-PRINCIPALS, we perform the PRINCIPALS step twice to obtain $\mathbf{X}^{*(0)}$ and $\mathbf{X}^{*(1)}$. When $\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$ generated by vε-PRINCIPALS converges to $\mathbf{X}^{*(\infty)}$, the estimate of $\mathbf{X}^*$ can be obtained from the final value of $\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$. The estimates of $\mathbf{Z}$ and $\mathbf{A}$ can then be calculated immediately from the estimate of $\mathbf{X}^*$ in the Model parameter estimation step of PRINCIPALS.

Note that $\dot{\mathbf{X}}^{*(t-1)}$ obtained in the vε acceleration step is not used as the estimate $\mathbf{X}^{*(t+1)}$ at the $(t+1)$-th iteration of the PRINCIPALS step. Thus, vε-PRINCIPALS speeds up the convergence of $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$ without affecting the convergence properties of ordinary PRINCIPALS.

3 Graves-Morris acceleration algorithm

Graves-Morris [5] studied generalization of Aitken's δ^2 of Aitken [1] for vector cases. The development of the GM algorithm is motivated by its numerical performance.

For a linearly convergent sequence $\{\mathbf{Y}^{(t)}\}_{t\geq 0}$, the GM algorithm generates an accelerated sequence $\{\ddot{\mathbf{Y}}^{(t)}\}_{t\geq 0}$ of $\{\mathbf{Y}^{(t)}\}_{t\geq 0}$ by using the following equation:

$$\ddot{\mathbf{Y}}^{(t-1)} = \mathbf{Y}^{(t+1)} - \frac{\langle \triangle \mathbf{Y}^{(t)}, \triangle \mathbf{Y}^{(t)} \rangle}{\langle \triangle \mathbf{Y}^{(t)}, \triangle^2 \mathbf{Y}^{(t)} \rangle} \triangle \mathbf{Y}^{(t+1)}, \tag{7}$$

where $\triangle^2 \mathbf{Y}^{(t)} = \triangle \mathbf{Y}^{(t+1)} - \triangle \mathbf{Y}^{(t)}$. Then, Equation (7) takes a hybrid form of the vector-valued Padé approximants. The Padé approximants are a particular type of rational approximation of functions and are a very important tool for deriving a fast convergent sequence. Baker and Graves-Morris [2] provided the detailed description of Padé approximants and the derivation of the GM algorithm.

We study the convergence of the GM algorithm. We consider the transformation such as

$$\ddot{\mathbf{Y}}^{(t-1)} = \mathbf{Y}^{(t+1)} + \frac{\triangle \mathbf{Y}^{(t+1)}}{1 - \dfrac{\langle \triangle \mathbf{Y}^{(t+1)}, \triangle \mathbf{Y}^{(t)} \rangle}{\langle \triangle \mathbf{Y}^{(t)}, \triangle \mathbf{Y}^{(t)} \rangle}}. \tag{8}$$

We suppose the following:

ASSUMPTION A: $\mathbf{Y}^{(t)} \to \mathbf{Y}^{(\infty)}$ as $t \to \infty$ in the sense of the norm.

The assumption means that there exists a constant $0 < K < 1$ such that

$$\|\triangle \mathbf{Y}^{(t+1)}\| \leq K \|\triangle \mathbf{Y}^{(t)}\|.$$

Lemma 3.1. *We suppose ASSUMPTION A. Then, we have* $\ddot{\mathbf{Y}}^{(t)} \to \mathbf{Y}^{(\infty)}$ *as* $k \to \infty$.

Proof. From Equation (8), we have

$$\|\ddot{\mathbf{Y}}^{(t-1)} - \mathbf{Y}^{(t+1)}\| = \left\| \frac{\triangle \mathbf{Y}^{(t+1)}}{1 - \dfrac{\langle \triangle \mathbf{Y}^{(t+1)}, \triangle \mathbf{Y}^{(t)} \rangle}{\langle \triangle \mathbf{Y}^{(t)}, \triangle \mathbf{Y}^{(t)} \rangle}} \right\| \leq \frac{\|\triangle \mathbf{Y}^{(t+1)}\|}{1 - \left| \dfrac{\langle \triangle \mathbf{Y}^{(t+1)}, \triangle \mathbf{Y}^{(t)} \rangle}{\langle \triangle \mathbf{Y}^{(t)}, \triangle \mathbf{Y}^{(t)} \rangle} \right|}.$$

From ASSUMPTION A, we have $\lim_{t\to\infty} \|\triangle \mathbf{Y}^{(t+1)}\| = 0$ and $|\langle \triangle \mathbf{Y}^{(t+1)}, \triangle \mathbf{Y}^{(t)} \rangle / \langle \triangle \mathbf{Y}^{(t)}, \triangle \mathbf{Y}^{(t)} \rangle| \neq 1$, since

$$|\langle \triangle \mathbf{Y}^{(t+1)}, \triangle \mathbf{Y}^{(t)} \rangle| \leq \|\triangle \mathbf{Y}^{(t+1)}\| \|\triangle \mathbf{Y}^{(t)}\| \leq K \|\triangle \mathbf{Y}^{(t)}\|^2 = K \langle \triangle \mathbf{Y}^{(t)}, \triangle \mathbf{Y}^{(t)} \rangle.$$

Then, we have the lemma. □

Below, we show the convergence of the GM algorithm.

Theorem 3.2. *Suppose that a vector sequence* $\{\mathbf{Y}^{(t)}\}_{t\geq 0}$ *fulfills ASSUMPTION A. Then, the vector sequence* $\{\ddot{\mathbf{Y}}^{(t)}\}_{t\geq 0}$ *generated by Equation (7) has the same accumulate point as* $\{\mathbf{Y}^{(t)}\}_{t\geq 0}$.

Proof. From

$$\begin{aligned}\langle \triangle \mathbf{Y}^{(t)}, \triangle^2 \mathbf{Y}^{(t)} \rangle &= \langle \triangle \mathbf{Y}^{(t)}, \triangle \mathbf{Y}^{(t+1)} - \triangle \mathbf{Y}^{(t)} \rangle \\ &= \langle \triangle \mathbf{Y}^{(t)}, \triangle \mathbf{Y}^{(t+1)} \rangle - \langle \triangle \mathbf{Y}^{(t)}, \triangle \mathbf{Y}^{(t)} \rangle,\end{aligned}$$

we have

$$\frac{\left\langle \triangle \mathbf{Y}^{(t)}, \triangle \mathbf{Y}^{(t)} \right\rangle}{\left\langle \triangle \mathbf{Y}^{(t)}, \triangle^2 \mathbf{Y}^{(t)} \right\rangle} = \frac{1}{\frac{\langle \triangle \mathbf{Y}^{(t)}, \triangle \mathbf{Y}^{(t+1)} \rangle}{\langle \triangle \mathbf{Y}^{(t)}, \triangle \mathbf{Y}^{(t)} \rangle} - 1}.$$

Equation (7) can be written by Equation (8). From Lemma 3.1, we complete the proof. □

4 Two-stage acceleration for PRINCIPALS

We show the two-stage acceleration for PRINCIPALS. The acceleration can be performed to obtain a fast convergent sequence of $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$: the first stage generates an accelerated sequence $\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$ of $\{\mathbf{X}^{(t)}\}_{t\geq 0}$ by the vε algorithm, and the GM algorithm in the second stage finds an accelerated sequence of $\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$. We denote the accelerated sequence obtained from the GM algorithm by $\{\ddot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$. Note that the first stage acceleration corresponds to vε-PRINCIPALS. In our experiments, the vε algorithm generates a fast linearly convergent sequence by improving the rate of convergence but rarely converges superlinearly. When $\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$ is a linearly convergent sequence, we expect that the GM algorithm generates the accelerated sequence $\{\ddot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$ of $\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$ such that it converges faster than $\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$.

We refer to the procedure of the two-stage acceleration as vεGM-PRINCIPALS. By alternating among the PRINCIPAL and two acceleration steps, vεGM-PRINCIPALS generates $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$, $\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$, and $\{\ddot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$ independently and alternatively.

Algorithm 4.1 (vεGM-PRINCIPALS)**.**

Step1 PRINCIPALS step: *Compute model parameters* $\mathbf{A}^{(t)}$ *and* $\mathbf{Z}^{(t)}$ *and determine optimal scaling parameter* $\mathbf{X}^{*(t+1)}$*.*

Step2 vε acceleration step: *Calculate* $\dot{\mathbf{X}}^{*(t-1)}$ *using* $\{\mathbf{X}^{*(t-1)}, \mathbf{X}^{*(t)}, \mathbf{X}^{*(t+1)}\}$ *from the vε algorithm:*

$$vec\dot{\mathbf{X}}^{*(t-1)} = vec\mathbf{X}^{*(t)} + \left[\left[\triangle vec\mathbf{X}^{*(t)}\right]^{-1} - \left[\triangle vec\mathbf{X}^{*(t-1)}\right]^{-1}\right]^{-1}.$$

Step3 GM acceleration step: *Calculate* $\ddot{\mathbf{X}}^{*(t-1)}$ *using* $\{\dot{\mathbf{X}}^{*(t-1)}, \dot{\mathbf{X}}^{*(t)}, \dot{\mathbf{X}}^{*(t+1)}, \dot{\mathbf{X}}^{*(t+2)}\}$ *from the GM algorithm:*

$$vec\ddot{\mathbf{X}}^{*(t-1)} = vec\dot{\mathbf{X}}^{*(t+1)} - \frac{\left\langle \triangle vec\dot{\mathbf{X}}^{(t)}, \triangle vec\dot{\mathbf{X}}^{(t)} \right\rangle}{\left\langle \triangle vec\dot{\mathbf{X}}^{(t)}, \triangle^2 vec\dot{\mathbf{X}}^{(t)} \right\rangle} \triangle vec\dot{\mathbf{X}}^{*(t+1)},$$

and check the convergence by

$$\left\| \triangle vec\ddot{\mathbf{X}}^{*(t-2)} \right\|^2 < \delta,$$

where δ *is a desired accuracy.*

We show the sequences generated by three steps and the numbers of iterations when the vε and GM acceleration steps start:

PRINCIPALS step	$\{\mathbf{X}^{*(t)}\}_{t\geq 0}$:	$\mathbf{X}^{*(0)}$	$\mathbf{X}^{*(1)}$	$\mathbf{X}^{*(2)}$	$\mathbf{X}^{*(3)}$	$\mathbf{X}^{*(4)}$	$\mathbf{X}^{*(5)}$	$\mathbf{X}^{*(6)}$	...
vε acceleration step	$\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$:			$\dot{\mathbf{X}}^{*(0)}$	$\dot{\mathbf{X}}^{*(1)}$	$\dot{\mathbf{X}}^{*(2)}$	$\dot{\mathbf{X}}^{*(3)}$	$\dot{\mathbf{X}}^{*(4)}$	...
GM acceleration step	$\{\ddot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$:						$\ddot{\mathbf{X}}^{*(0)}$	$\ddot{\mathbf{X}}^{*(1)}$	...

The GM acceleration step starts after executing the vε acceleration step thrice. Thus, to obtain $\ddot{\mathbf{X}}^{*(0)}$, the GM acceleration step requires the subsequence $\{\dot{\mathbf{X}}^{*(t)}\}_{0\leq t\leq 3}$ in the vε acceleration step and $\{\mathbf{X}^{*(t)}\}_{0\leq t\leq 5}$ in the PRINCIPALS step.

During iterations of vεGM-PRINCIPALS, the value of $\ddot{\mathbf{X}}$ obtained in the GM acceleration step is not used as the estimate at the next PRINCIPALS step similar to vε-PRINCIPALS. Therefore, the GM acceleration step does not affect the convergence of PRINCIPALS. We obtain the estimates of $\mathbf{X}^*$, $\mathbf{Z}$, and $\mathbf{A}$ in the same manner as vε-PRINCIPALS when vεGM-PRINCIPALS terminates.

5 Numerical experiment

We study how much faster vεGM-PRINCIPALS converges than PRINCIPALS and vε-PRINCIPALS. All computations are performed with the statistical package R [8] executing on Intel Core i5 3.3 GHz with 4 GB RAM. CPU times (in seconds) taken are measured by the function `proc.time`[1]. For all experiments, δ for convergence of vε-PRINCIPALS and vεGM-PRINCIPALS is set to 10^{-8}, and PRINCIPALS terminates when $|\theta^{(t+1)} - \theta^{(t)}| < 10^{-8}$, where $\theta^{(t)}$ is the t-th update of θ calculated from Equation (4). Each algorithm also stops when the number of iterations exceeds 100,000. We apply these algorithms to a random data matrix of 100 observations on 20 variables with 10 levels and measure the number of iterations and CPU time for $r = 2$. The procedure is replicated 100 times.

Table 1 shows the summary of statistics of the numbers of iterations and CPU times of these algorithms from 100 simulated data. The table shows that vεGM-PRINCIPALS considerably reduces the number of iterations and the CPU time. Table 2 shows the summary of statistics of the iteration and CPU time speed-ups for comparing the speed of convergence of PRINCIPALS with those of two acceleration algorithms. The iteration speed-up is defined as the number of iterations required for PRINCIPALS divided by the number of iterations required for the acceleration algorithm. The CPU time speed-up is calculated similarly to the iteration speed-up. The values of the iteration and CPU time speed-ups indicate that PRINCIPALS requires 3.2 to 4.4 times greater number of iterations and 2.7 to 3.9 times longer CPU time than those of vεGM-PRINCIPALS.

We compare the performance of vεGM-PRINCIPALS with that of vε-PRINCIPALS. Figure 1 shows the boxplots of the iteration and CPU time speed-ups of two acceleration algorithms. Table 2 and Figure 1 indicate that the GM acceleration step finds an accelerated sequence of $\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$ in the vε acceleration step, and thus the two-stage acceleration improves the overall computational efficiency of PRINCIPALS. Figure 2 presents the scatter plots of the iteration and CPU time speed-ups of vεGM-PRINCIPALS for those of vε-PRINCIPALS. The figure shows that vεGM-PRINCIPALS always converges faster than vε-PRINCIPALS and accelerates the convergence of $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$ more than vε-PRINCIPALS whenever vε-PRINCIPALS converges

[1] Times are typically available to 10 msec.

faster. For the case that the values of the iteration and CPU time speed-ups of vε-PRINCIPALS are more than 4, vεGM-PRINCIPALS gives higher values. Figure 3 shows the scatter plots of the iteration and CPU time speed-ups of vεGM-PRINCIPALS for the number of iterations of PRINCIPALS. The figure indicates that vεGM-PRINCIPALS works well to accelerate the convergence more than vε-PRINCIPALS for the larger number of iterations of PRINCIPALS. Thus, it is clear that the two-stage acceleration is very advantageous.

6 Concluding remarks

In this paper, we proposed two-stage acceleration for the ALS algorithm for non-linear PCA. The first stage generates the accelerated sequence $\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$ of $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$ using the vε algorithm and the next stage accelerates the convergence of $\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$ using the GM algorithm. Then, both acceleration algorithms find fast convergent sequences without modification of the estimation equations of the ALS algorithm. Therefore, the two-stage acceleration speeds up the convergence of $\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$ while still preserving the stable convergence property of the ALS algorithm. The vε and GM algorithms are fairly simple computational procedures, and their computational costs are less expensive than those for matrix inversion and for solving the eigenvalue problem in the ALS algorithm for non-linear PCA.

The numerical experiments employing simulated data demonstrated that the two-stage acceleration improves the computational efficiency of the vε acceleration of Kuroda et al. [6], and then significantly speeds up the convergence of $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$ in terms of the number of iterations and computation time. In this paper, we described the two-stage acceleration only for PRINCIPALS, but it is applicable to PRINCALS as well.

Acknowledgement

This research is supported by the Japan Society for the Promotion of Science (JSPS), Grant-in-Aid for Scientific Research (C), No 22500265 and No 24500353.

Bibliography

[1] Aitken, A. (1926). *On Bernoulli's numerical solution of algebraic equations.* Proceedings of the Royal Society of Edinburgh, **46**, 289-305.

[2] Baker, G. A., Jr. and Graves-Morris, P.R. (1996). *Padé Approximants.* Cambridge University Press, Cambridge.

[3] Brezinski, C. and Zaglia, M. (1991). *Extrapolation methods: theory and practice.* Elsevier Science Ltd. North-Holland, Amsterdam.

[4] Gifi, A. (1990). *Nonlinear multivariate analysis.* John Wiley & Sons, Ltd., Chichester.tem

[5] Graves-Morris, P.R. (1992). *Extrapolation methods for vector sequences.* Numerische Mathematik, **61**, 475-487.

[6] Kuroda, M., Mori, Y., Iizuka, M. and Sakakihara, M. (2011). *Acceleration of the alternating least squares algorithm for principal components analysis.* Computational Statistics and Data Analysis, **55**, 143-153.

[7] Mori, Y., Tanaka, Y. and Tarumi, T. (1997). *Principal component analysis based on a subset of variables for qualitative data.* Data Science, Classification, and Related Methods (Proceedings of IFCS-96), 547-554, Springer-Verlag.

[8] R Development Core Team (2011). *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL `http://www.R-project.org/`.

[9] Young, F.W., Takane, Y., and de Leeuw, J. (1978). *Principal components of mixed measurement level multivariate data: An alternating least squares method with optimal scaling features.* Psychometrika, **43**, 279-281.

[10] Wynn, P. (1962). *Acceleration techniques for iterated vector and matrix problems.* Mathematics of Computation, **16**, 301-322.

	The number of iterations			CPU time		
	PRINCIPALS	vε	vεGM	PRINCIPALS	vε	vεGM
Min	181.0	65.0	57.00	3.510	1.410	1.310
1st Qu.	332.5	103.8	91.25	6.258	2.145	1.992
Median	475.5	154.5	134.00	8.780	3.035	2.730
Mean	605.9	193.8	166.37	11.130	3.789	3.361
3rd Qu.	740.0	241.0	207.75	13.340	4.620	4.062
Max.	3595.0	820.0	615.00	67.250	15.690	12.170

Table 1. Summary of statistics of the numbers of iterations and CPU times of PRINCIPALS, vε-PRINCIPLAS (vε) and vεGM-PRINCIPALS (vεGM) from 100 simulated data ($r = 2$).

	Iteration speed-up		CPU time speed-up	
	vε	vεGM	vε	vεGM
Min	1.383	1.414	1.356	1.356
1st Qu.	2.729	3.175	2.577	2.748
Median	3.175	3.677	2.895	3.209
Mean	3.201	3.758	2.953	3.332
3rd Qu.	3.773	4.377	3.422	3.895
Max.	6.076	9.038	5.548	8.451

Table 2. Summary of statistics of the number of iterations and CPU time speed-ups of vε-PRINCIPLAS (vε) and vεGM-PRINCIPALS (vεGM) from 100 simulated data ($r = 2$).

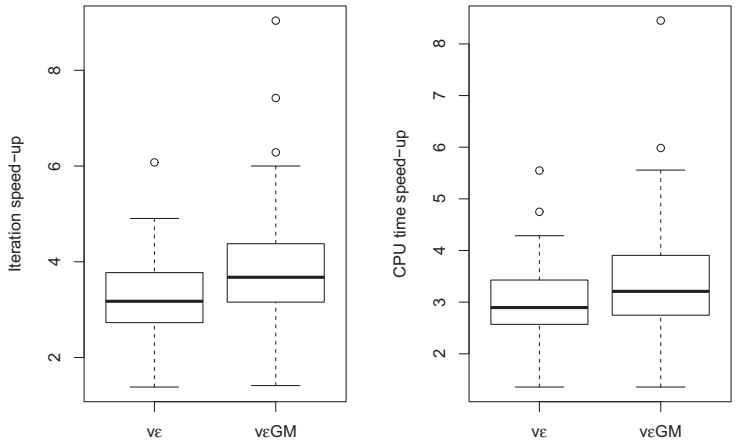

Figure 1. Boxplots of iteration and CPU time speed-ups of vε-PRINCIPALS (vε) and vεGM-PRINCIPALS (vεGM) from 100 simulated data ($r = 2$).

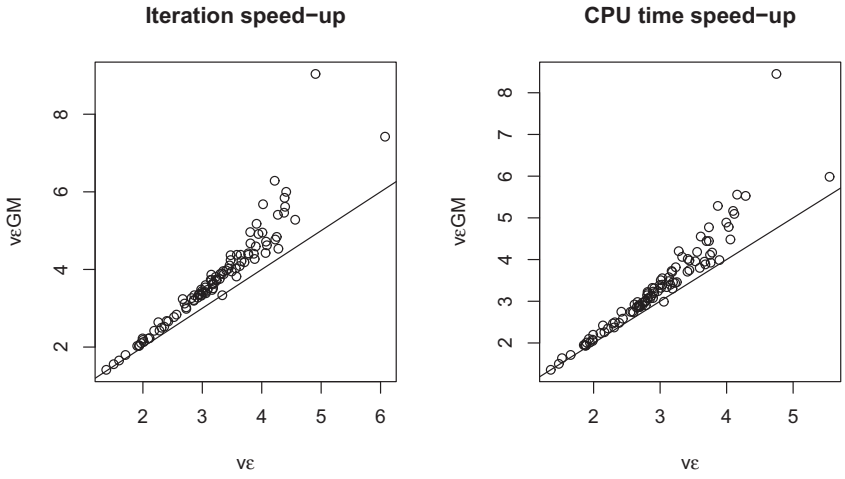

Figure 2. Scatter plots of iteration and CPU time speed-ups of vεGM-PRINCIPALS (vεGM) for vε-PRINCIPALS(vε) from 100 simulated data ($r = 2$).

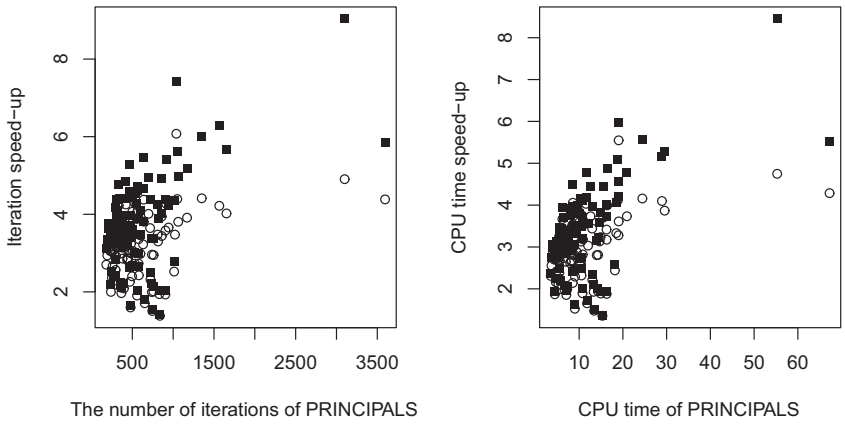

Figure 3. Scatter plots of iteration and CPU time speed-ups of vε-PRINCIPALS (◯) and vεGM-PRINCIPALS (■) for the number of iterations of PRINCIPALS from 100 simulated data ($r = 2$).