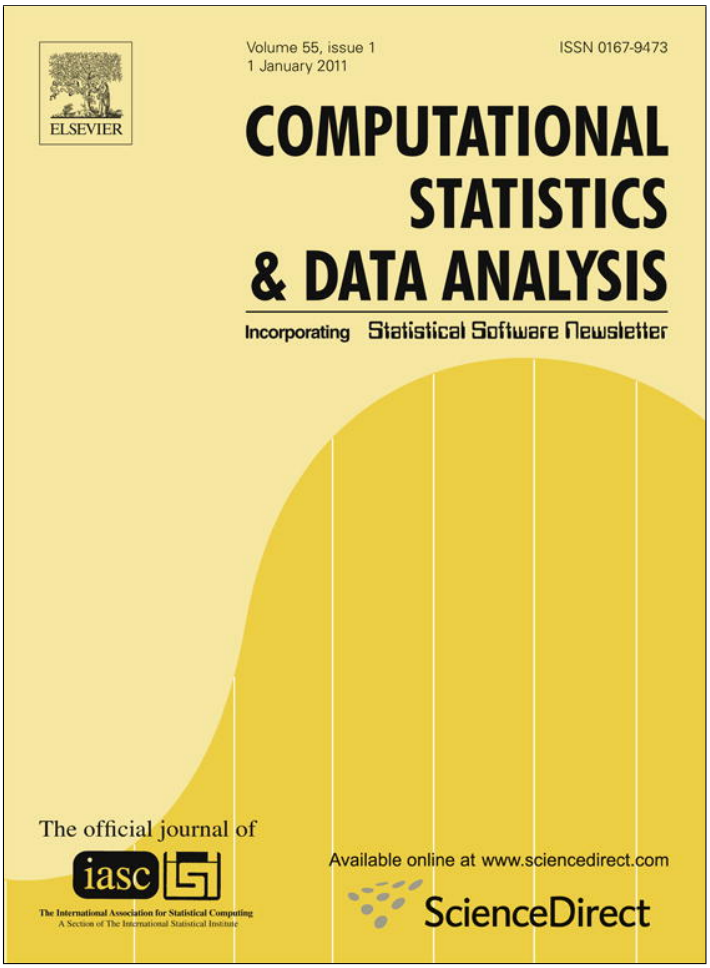

Computational Statistics and Data Analysis 55 (2011) 143–153

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

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

Acceleration of the alternating least squares algorithm for principal components analysis

Masahiro Kuroda [a,*], Yuichi Mori [a], Masaya Iizuka [b], Michio Sakakihara [c]

[a] Department of Socio-Information, Okayama University of Science, 1-1 Ridaicho, Kita-ku, Okayama 700-0005, Japan

[b] Graduate School of Environmental Science, Okayama University 1-1-1 Tsushima-naka, Kita-ku, Okayama 700-8530, Japan

[c] Department of Information Science, Okayama University of Science, 1-1 Ridaicho, Kita-ku, Okayama 700-0005, Japan

ARTICLE INFO

Article history:
Received 16 October 2009
Received in revised form 28 April 2010
Accepted 1 June 2010
Available online 10 June 2010

Keywords:
Alternating least squares algorithm
Vector ε algorithm
Acceleration of convergence
PRINCIPALS
PRINCALS

ABSTRACT

Principal components analysis (PCA) is a popular descriptive multivariate method for handling quantitative data and it can be extended to deal with qualitative data and mixed measurement level data. The existing algorithms for extended PCA are PRINCIPALS of Young et al. (1978) and PRINCALS of Gifi (1989) in which the alternating least squares algorithm is utilized. These algorithms based on the least squares estimation may require many iterations in their application to very large data sets and variable selection problems and may take a long time to converge. In this paper, we derive a new iterative algorithm for accelerating the convergence of PRINCIPALS and PRINCALS by using the vector ε algorithm of Wynn (1962). The proposed acceleration algorithm speeds up the convergence of the sequence of the parameter estimates obtained from PRINCIPALS or PRINCALS. Numerical experiments illustrate the potential of the proposed acceleration algorithm.

1. Introduction

Principal components analysis (PCA) is a popular descriptive multivariate method for handling quantitative data and is extended to deal with qualitative data and mixed measurement level data. The existing algorithms for extended PCA are PRINCIPALS of Young et al. (1978) and PRINCALS of Gifi (1989) in which the alternating least squares (ALS) algorithm is utilized. These algorithms alternate between optimal scaling for quantifying qualitative data and the analysis of the optimally scaled data using the ordinary PCA approach. We will refer to PRINCIPALS and PRINCALS as PCA.ALS when not distinguishing between them.

In the application of extended PCA for very large data sets and variable selection problems, many iterations and much computation time may be required for convergence of PCA.ALS. For example, for PCA based on a subset of variables for qualitative data, the PRINCIPALS approach taken by Mori et al. (1997) obtained estimates only after a large number of iterations. In this paper, we derive a new iterative algorithm for speeding up the convergence of PCA.ALS. The proposed algorithm uses the vector ε (vε) algorithm of Wynn (1962) for accelerating the convergence of PCA.ALS. We refer to vε accelerated PCA.ALS as vε-PCA.ALS. During iterations of vε-PCA.ALS, the vε algorithm generates an accelerated sequence of optimal scaling data estimated by PCA.ALS. Then the vε accelerated sequence converges faster than the original sequence of the estimated optimal scaled data.

The paper is organized as follows. We briefly describe PCA for a mixture of quantitative and qualitative data in Section 2, and describe PRINCIPALS and PRINCALS for finding least squares estimates of the model and optimal scaling parameters in

* Corresponding author.
E-mail address: kuroda@soci.ous.ac.jp (M. Kuroda).

0167-9473/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.csda.2010.06.001

Section 3. Section 4 presents the procedure of vε-PCA.ALS that adds the vε algorithm to PCA.ALS for speeding up convergence. Numerical experiments in Section 5 illustrate the performance and properties of the vε acceleration of PRINCIPALS. In Section 6, we present our concluding remarks.

2. Principal components analysis with variables measured with a variety of scale levels

PCA transforms linearly an original data set of variables into a substantially smaller set of uncorrelated variables that contains much of the information in the original data set. The original data matrix is then replaced by an estimate constructed by forming the product of matrices of component scores and eigenvectors.

Let $\mathbf{X} = (\mathbf{X}_1\ \mathbf{X}_2\ \ldots\ \mathbf{X}_p)$ be an $n \times p$ matrix of n observations on p variables and be columnwise standardized. In PCA, we postulate that $\mathbf{X}$ is approximated by the following bilinear form:

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

where $\mathbf{Z} = (\mathbf{Z}_1\ \mathbf{Z}_2\ \ldots\ \mathbf{Z}_r)$ is an $n \times r$ matrix of n component scores on r ($1 \le r \le p$) components, and $\mathbf{A} = (\mathbf{A}_1\ \mathbf{A}_2\ \ldots\ \mathbf{A}_r)$ 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.

Ordinary PCA assumes that all variables are measured with interval and ratio scales and can be applied only to quantitative data. When the observed data contain several different types of variables with nominal, ordinal, interval and ratio scales, ordinary PCA cannot be directly applied to such data. In such situations, optimal scaling is used to quantify the observed qualitative data and then ordinary PCA can be applied.

To quantify $\mathbf{X}_j$ of qualitative variable j with K_j categories, the vector is coded by 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 $\mathbf{X}^* = (\mathbf{X}_1^*\ \mathbf{X}_2^*\ \ldots\ \mathbf{X}_p^*)$ be an $n \times p$ matrix of optimally scaled observations to satisfy restrictions

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

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

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

To apply PCA to data with mixed measurement levels, we determine the optimal scaling parameter $\mathbf{X}^*$, in addition to estimating $\mathbf{Z}$ and $\mathbf{A}$.

3. Alternating least squares algorithm for principal components analysis

A possible computational algorithm for estimating simultaneously $\mathbf{Z}$, $\mathbf{A}$ and $\mathbf{X}^*$ is the ALS algorithm. The algorithm involves dividing an entire set of parameters of a model into the model parameters and the optimal scaling parameters, and finds the least squares estimates for these parameters. The model parameters are used to compute the predictive values of the model. The optimal scaling parameters are obtained by solving the least squares regression problem for the predictive values. Krijnen (2006) gave sufficient conditions for convergence of the ALS algorithm and discussed convergence properties in its application to several statistical models. Kiers (2002) described setting up the ALS and iterative majorization algorithms for solving various matrix optimization problems.

3.1. PRINCIPALS

PRINCIPALS proposed by Young et al. (1978) is a method for utilizing the ALS algorithm for PCA of data with mixed measurement levels of single discrete and continuous, single nominal, ordinal and numerical variables. PRINCIPALS alternates between ordinary PCA and optimal scaling, and minimizes θ^* defined by Eq. (4) under the 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.

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

- *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 $\mathbf{Z}^{(t)} = \mathbf{X}^{*(t)}\mathbf{A}^{(t)}$.

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

$$\mathbf{X}^{*(t+1)} = \arg\min_{\mathbf{X}^*} \operatorname{tr}(\mathbf{X}^* - \hat{\mathbf{X}}^{(t+1)})^\top(\mathbf{X}^* - \hat{\mathbf{X}}^{(t+1)})$$

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.

3.2. PRINCALS

PRINCALS by Gifi (1989) can handle multiple nominal variables in addition to the single nominal, ordinal and numerical variables accepted in PRINCIPALS. We denote the set of multiple variables by $\mathcal{J}_M$ and the set of single variables with single nominal and ordinal scales and numerical measurements by $\mathcal{J}_S$. For $\mathbf{X}$ consisting of a mixture of multiple and single variables, the algorithm alternates between estimation of $\mathbf{Z}$, $\mathbf{A}$ and $\mathbf{X}^*$ subject to minimizing

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

under the restriction

$$\mathbf{Z}^\top\mathbf{1}_n = \mathbf{0}_r \quad \text{and} \quad \mathbf{Z}^\top\mathbf{Z} = n\mathbf{I}_p. \tag{6}$$

For the initialization of PRINCALS, we determine initial data $\mathbf{Z}^{(0)}$, $\mathbf{A}^{(0)}$ and $\mathbf{X}^{*(0)}$. The values of $\mathbf{Z}^{(0)}$ are initialized with random numbers under the restriction (6). For $j \in \mathcal{J}_M$, the initial value of $\mathbf{X}_j^*$ is obtained by $\mathbf{X}_j^{*(0)} = \mathbf{G}_j(\mathbf{G}_j^\top\mathbf{G}_j)^{-1}\mathbf{G}_j^\top\mathbf{Z}^{(0)}$. For $j \in \mathcal{J}_S$, $\mathbf{X}_j^{*(0)}$ is defined as the first K_j successive integers under the normalization restriction, and the initial value of $\mathbf{A}_j$ is calculated as the vector $\mathbf{A}_j^{(0)} = \mathbf{Z}^{(0)\top}\mathbf{X}_j^{*(0)}$. Given these initial values, PRINCALS as provided in Michailidis and de Leeuw (1998) iterates the following two steps:

- *Model parameter estimation step:* Calculate $\mathbf{Z}^{(t+1)}$ by

$$\mathbf{Z}^{(t+1)} = p^{-1}\left(\sum_{j\in\mathcal{J}_M}\mathbf{X}_j^{*(t)} + \sum_{j\in\mathcal{J}_S}\mathbf{X}_j^{*(t)}\mathbf{A}_j^{(t)}\right).$$

Columnwise center and orthonormalize $\mathbf{Z}^{(t+1)}$. Estimate $\mathbf{A}_j^{(t+1)}$ for the single variable j by $\mathbf{A}_j^{(t+1)} = \mathbf{Z}^{(t+1)\top}\mathbf{X}_j^{*(t)}/\mathbf{X}_j^{*(t)\top}\mathbf{X}_j^{*(t)}$.

- *Optimal scaling step:* Estimate the optimally scaled vector for $j \in \mathcal{J}_M$ by

$$\mathbf{X}_j^{*(t+1)} = \mathbf{G}_j(\mathbf{G}_j^\top\mathbf{G}_j)^{-1}\mathbf{G}_j^\top\mathbf{Z}^{(t+1)}$$

and for $j \in \mathcal{J}_S$ by

$$\mathbf{X}_j^{*(t+1)} = \mathbf{G}_j(\mathbf{G}_j^\top\mathbf{G}_j)^{-1}\mathbf{G}_j^\top\mathbf{Z}^{(t+1)}\mathbf{A}_j^{(t+1)}/\mathbf{A}_j^{(t+1)\top}\mathbf{A}_j^{(t+1)}$$

under measurement restrictions on each of the variables.

4. The vε acceleration of the ALS algorithm

We briefly introduce the vε algorithm of Wynn (1962) used in the acceleration of PCA.ALS. The vε algorithm is utilized to speed up the convergence of a slowly convergent vector sequence and is very effective for linearly converging sequences. Kuroda and Sakakihara (2006) proposed the ε-accelerated EM algorithm that speeds up the convergence of the EM sequence via the vε algorithm and demonstrated that its speed of convergence is significantly faster than that of the EM algorithm. Wang et al. (2008) studied the convergence properties of the ε-accelerated EM algorithm.

Let $\{\mathbf{Y}^{(t)}\}_{t\geq 0} = \{\mathbf{Y}^{(0)}, \mathbf{Y}^{(1)}, \mathbf{Y}^{(2)}, \ldots\}$ be a linear convergent sequence generated by an iterative computational procedure and let $\{\dot{\mathbf{Y}}^{(t)}\}_{t\geq 0} = \{\dot{\mathbf{Y}}^{(0)}, \dot{\mathbf{Y}}^{(1)}, \dot{\mathbf{Y}}^{(2)}, \ldots\}$ be the accelerated sequence of $\{\mathbf{Y}^{(t)}\}_{t\geq 0}$. Then the vε algorithm generates $\{\dot{\mathbf{Y}}^{(t)}\}_{t\geq 0}$ by using

$$\dot{\mathbf{Y}}^{(t-1)} = \mathbf{Y}^{(t)} + \left[\left[(\mathbf{Y}^{(t-1)} - \mathbf{Y}^{(t)})\right]^{-1} + \left[(\mathbf{Y}^{(t+1)} - \mathbf{Y}^{(t)})\right]^{-1}\right]^{-1}, \tag{7}$$

where $[\mathbf{Y}]^{-1} = \mathbf{Y}/\|\mathbf{Y}\|^2$ and $\|\mathbf{Y}\|$ is the Euclidean norm of $\mathbf{Y}$. For the detailed derivation of Eq. (7), see the Appendix. 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 $\{\mathbf{Y}^{(t)}\}_{t\geq 0}$.

Table 1
The numbers of iterations and CPU times of PRINCIPALS and vε-PRINCIPALS for $r = 1, \ldots, 15$: the mean, minimum and maximum numbers of iterations from 50 simulated data.

r	The number of iterations				CPU time			
	PRINCIPALS		vε-PRINCIPALS		PRINCIPALS		vε-PRINCIPALS	
	Mean	(Min., Max.)	Mean	(Min., Max.)	Mean	(Min., Max.)	Mean	(Min., Max.)
1	17.60	(12, 28)	7.28	(6, 10)	1.92	(1.31, 3.07)	1.03	(0.89, 1.36)
2	168.80	(64, 1158)	53.42	(21, 149)	21.59	(7.18, 189.63)	7.14	(2.66, 25.66)
3	189.16	(74, 1392)	62.00	(45, 245)	24.94	(8.60, 242.08)	8.47	(5.63, 45.19)
4	244.04	(66, 1996)	81.90	(24, 435)	34.82	(7.53, 397.92)	12.42	(3.03, 93.89)
5	260.98	(83, 1141)	85.38	(41, 295)	34.92	(9.75, 187.49)	12.01	(5.21, 52.19)
6	218.26	(88, 481)	75.40	(32, 128)	27.68	(10.36, 65.58)	10.23	(4.95, 18.51)
7	229.98	(96, 739)	79.74	(27, 189)	29.49	(10.61, 109.23)	10.98	(3.28, 29.61)
8	218.80	(79, 596)	77.10	(33, 130)	26.63	(8.58, 80.99)	9.96	(3.89, 18.73)
9	205.22	(87, 1138)	65.30	(33, 301)	25.68	(9.59, 180.51)	8.66	(3.96, 51.19)
10	228.94	(53, 809)	75.20	(29, 375)	29.08	(5.84, 120.67)	10.20	(3.48, 61.00)
11	164.74	(78, 368)	54.04	(32, 126)	20.07	(8.94, 48.23)	7.04	(3.98, 17.48)
12	239.30	(85, 982)	72.34	(32, 182)	31.17	(9.78, 152.03)	9.93	(3.97, 29.75)
13	185.18	(77, 618)	55.96	(23, 155)	23.06	(8.67, 87.61)	7.43	(2.92, 23.27)
14	298.48	(78, 1518)	97.18	(26, 365)	41.32	(8.81, 265.95)	14.32	(3.24, 68.77)
15	235.50	(84, 2732)	77.32	(32, 1153)	35.46	(9.51, 627.89)	13.55	(3.96, 311.59)

We assume that $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$ generated by PCA.ALS converges to a limit point $\mathbf{X}^{*(\infty)}$. Then vε-PCA.ALS produces a faster convergent sequence $\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$ of $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$ by using the vε algorithm and enables the acceleration of convergence of PCA.ALS. The general procedure of vε-PCA.ALS iterates the following two steps:

- *PCA.ALS step:* Compute model parameters $\mathbf{A}^{(t)}$ and $\mathbf{Z}^{(t)}$ and determine optimal scaling parameter $\mathbf{X}^{*(t+1)}$.
- *Acceleration step:* Calculate $\dot{\mathbf{X}}^{*(t-1)}$ using $\{\mathbf{X}^{*(t-1)}, \mathbf{X}^{*(t)}, \mathbf{X}^{*(t+1)}\}$ from the vε algorithm:

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

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

$$\left\|\operatorname{vec}(\dot{\mathbf{X}}^{*(t-1)} - \dot{\mathbf{X}}^{*(t-2)})\right\|^2 < \delta,$$

where δ is a desired accuracy.

Before starting the iteration, we determine initial data $\mathbf{X}^{*(0)}$ satisfying the restriction (3) and execute the *PCA.ALS step* twice to generate $\{\mathbf{X}^{*(0)}, \mathbf{X}^{*(1)}, \mathbf{X}^{*(2)}\}$.

vε-PCA.ALS is designed to generate $\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$ converging to $\mathbf{X}^{*(\infty)}$. Thus the estimate of $\mathbf{X}^*$ can be obtained from the final value of $\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$ when vε-PCA.ALS terminates. 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 PCA.ALS.

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

5. Numerical experiments

In this section, we examine the performance of vε-PCA.ALS by employing simulated and real data, and demonstrate the advantage of vε-PCA.ALS over PCA.ALS in terms of the number of iterations and CPU time (in seconds) required for convergence. All computations are performed with the statistical package R (R Development Core Team, 2008) executing on a Celeron 2.8 GHz computer with 1 GB of memory. CPU times taken are measured by the function `proc.time`.[1] Here we use PRINCIPALS as PCA.ALS. For all experiments, δ for convergence of vε-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 Eq. (4).

5.1. Simulated data: comparison of the number of iterations and CPU time

In this experiments, we study how much faster vε-PRINCIPALS converges than PRINCIPALS. We apply these algorithms to a random data matrix of 60 observations on 40 variables with 10 levels and measure the number of iterations and CPU time taken for $r = 1, \ldots, 15$. The procedure is replicated 50 times.

[1] Times are typically available to 10 ms.

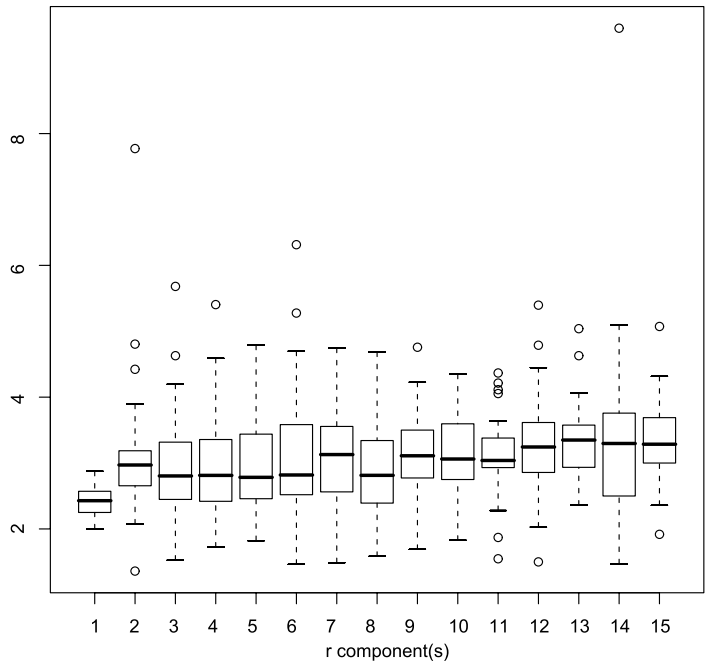

Fig. 1. Boxplots of iteration speed-up from 50 simulated data for $r = 1, \ldots, 15$.

Table 1 shows the summaries of the number of iterations and CPU time taken by PRINCIPALS and vε-PRINCIPALS. We see from the table that PRINCIPALS requires more iterations and takes a longer time than vε-PRINCIPALS. Table 2 reports the iteration and CPU time speed-ups for comparing the speed of convergence of PRINCIPALS with that of vε-PRINCIPALS. The iteration speed-up is defined as the number of iterations required for PRINCIPALS divided by the number of iterations required for vε-PRINCIPALS. The CPU time speed-up is calculated similarly to the iteration speed-up. We see from the second to fifth columns in Table 2 that vε-PRINCIPALS converges about 3 times faster than PRINCIPALS in terms of the mean number of iterations and the mean CPU time. The boxplots of the iteration and CPU time speed-ups in Figs. 1 and 2 also indicate that vε-PRINCIPALS converges 2 to 4 times faster than PRINCIPALS for both. The advantage of the vε acceleration is very obvious.

The computation time per iteration of vε-PRINCIPALS is greater than that of PRINCIPALS due to computation of the *Acceleration step*. We calculate the CPU time ratio of PRINCIPALS to vε-PRINCIPALS. The ratio is defined by

$$\text{the CPU time ratio} = \frac{\text{CPU time of PRINCIPALS}}{\text{the number of iterations of PRINCIPALS}} \Big/ \frac{\text{CPU time of v}\varepsilon\text{-PRINCIPALS}}{\text{the number of iterations of v}\varepsilon\text{-PRINCIPALS}}.$$

The values of the mean CPU time ratio reported in the sixth and seventh columns of Table 2 demonstrate that the computation time of the *PRINCIPALS step* accounts for 90–95% of that of vε-PRINCIPALS except when $r = 1$ and thus considerable time is required for computation of the *PRINCIPALS step*. This means that the computational cost of the vε acceleration in the *Acceleration step* is less expensive than that of PRINCIPALS. Therefore the vε acceleration enables a significant reduction in the numbers of iterations and computation times with less computational effort.

5.2. Real data

5.2.1. Studies of convergence

We use data obtained in teacher evaluation by students. The data are obtained from 56 students and consisted of 13 categorical variables with 5 levels each; the lowest evaluation level is 1 and the highest 5.

The second and third columns in Table 3 and Fig. 3 show the numbers of iterations of PRINCIPALS and vε-PRINCIPALS. The fourth and fifth columns in Table 3 give CPU times taken by PRINCIPALS and vε-PRINCIPALS. The values of the iteration speed-up in the sixth column of Table 3 also indicate that vε-PRINCIPALS converges 3–4 times faster than PRINCIPALS. The CPU time speed-up in the seventh column of the table demonstrates that the computation times of vε-PRINCIPALS are 2.5–3.5 times shorter than those of PRINCIPALS.

To check the convergence of both algorithms, we calculate the maximum absolute errors defined by

$$\mathrm{Err}(\mathbf{X}^{*(t)}) = \max|\mathrm{vec}(\mathbf{X}^{*(t)} - \mathbf{X}^{*(\infty)})|,$$
$$\mathrm{Err}(\dot{\mathbf{X}}^{*(t)}) = \max|\mathrm{vec}(\dot{\mathbf{X}}^{*(t)} - \mathbf{X}^{*(\infty)})|,$$

where $\mathbf{X}^{*(\infty)}$ is the final value of $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$. Fig. 4 illustrates the traces of $\{\mathrm{Err}(\mathbf{X}^{*(t)})\}_{t\geq 0}$ and $\{\mathrm{Err}(\dot{\mathbf{X}}^{*(t)})\}_{t\geq 0}$ for $r = 2, 4, 6$ and 8. The figures demonstrate that $\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$ converges to $\mathbf{X}^{*(\infty)}$ significantly faster than $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$. For each r, $\mathrm{Err}(\dot{\mathbf{X}}^{*(t)})$

Table 2
Iteration and CPU time speed-ups, and CPU time ratios for $r = 1, \ldots, 15$: the mean, minimum and maximum numbers of iterations from 50 simulated data.

r	Speed-up				CPU time ratio	
	Iteration		CPU time			
	Mean	(Min., Max.)	Mean	(Min., Max.)	Mean	(Min., Max.)
1	2.41	(2.00, 2.88)	1.86	(1.47, 2.30)	0.77	(0.73, 0.81)
2	3.06	(1.36, 7.77)	2.84	(1.25, 7.39)	0.93	(0.89, 0.95)
3	2.93	(1.53, 5.68)	2.73	(1.43, 5.36)	0.93	(0.91, 0.95)
4	2.96	(1.72, 5.40)	2.76	(1.52, 5.12)	0.93	(0.87, 0.95)
5	2.97	(1.82, 4.79)	2.78	(1.71, 4.56)	0.94	(0.92, 0.95)
6	3.07	(1.46, 6.32)	2.87	(1.33, 5.95)	0.93	(0.91, 0.95)
7	3.11	(1.49, 4.74)	2.91	(1.33, 4.45)	0.93	(0.89, 0.95)
8	2.92	(1.58, 4.69)	2.74	(1.47, 4.44)	0.94	(0.92, 0.95)
9	3.13	(1.69, 4.76)	2.92	(1.60, 4.47)	0.93	(0.91, 0.95)
10	3.09	(1.83, 4.35)	2.89	(1.68, 4.12)	0.93	(0.90, 0.95)
11	3.11	(1.55, 4.37)	2.90	(1.44, 4.12)	0.93	(0.90, 0.95)
12	3.25	(1.50, 5.40)	3.04	(1.38, 5.11)	0.93	(0.91, 0.95)
13	3.31	(2.36, 5.04)	3.07	(2.22, 4.71)	0.93	(0.89, 0.95)
14	3.30	(1.47, 9.60)	3.08	(1.37, 9.18)	0.93	(0.89, 0.96)
15	3.32	(1.92, 5.07)	3.09	(1.81, 4.79)	0.93	(0.85, 0.96)

Table 3
The numbers of iterations and CPU times of PRINCIPALS and vε-PRINCIPALS, and the iteration and CPU time speed-ups.

r	The number of iterations		CPU time		Speed-up	
	PRINCIPALS	vε-PRINCIPALS	PRINCIPALS	vε-PRINCIPALS	Iteration	CPU time
1	9	4	0.35	0.23	2.25	1.50
2	92	23	3.87	1.10	4.00	3.51
3	28	9	1.14	0.46	3.11	2.49
4	25	8	1.00	0.36	3.57	2.78
5	28	10	1.13	0.49	2.80	2.29
6	29	9	1.16	0.46	3.22	2.52
7	28	9	1.11	0.44	3.11	2.55
8	47	14	1.97	0.68	3.36	2.89
9	45	13	1.82	0.63	3.46	2.90
10	45	14	1.84	0.67	3.21	2.73
11	33	10	1.35	0.49	3.30	2.75
12	40	10	1.61	0.48	4.00	3.33

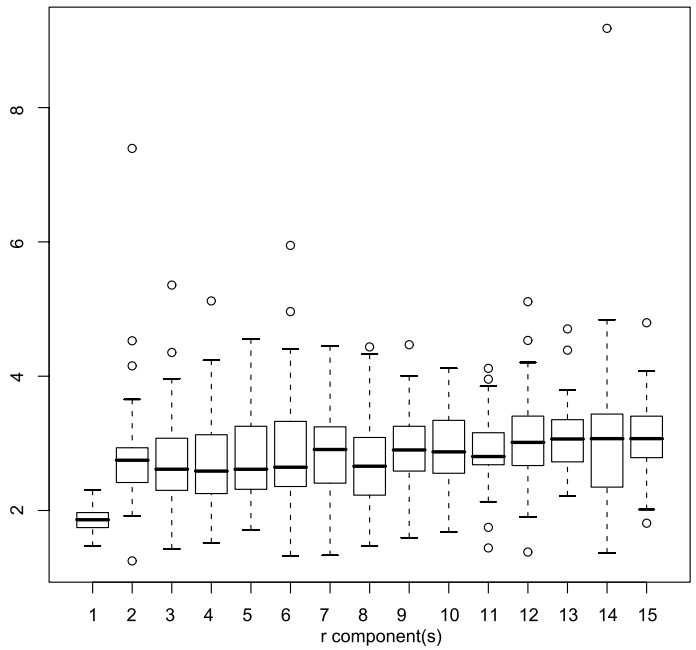

Fig. 2. Boxplots of CPU time speed-up from 50 simulated data for $r = 1, \ldots, 15$.

after the convergence of vε-PRINCIPALS is smaller than 10^{-4}; this therefore means that $\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$ converges to the same final value of $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$.

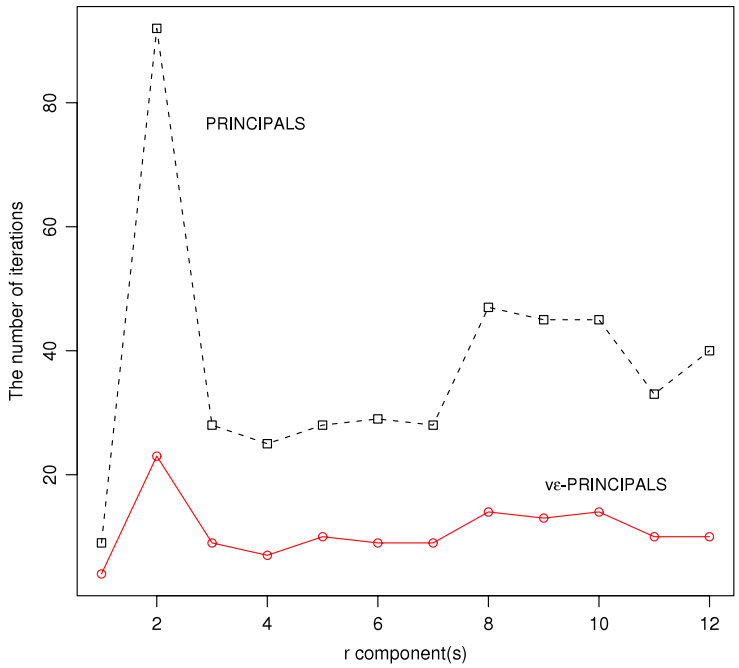

Fig. 3. Plots of the number of iterations for PRINCIPALS and vε-PRINCIPALS for each r.

Table 4
Rates of convergence τ and $\dot{\tau}$ of PRINCIPALS to vε-PRINCIPALS.

r	τ	$\dot{\tau}$
1	0.060	0.001
2	0.812	0.667
3	0.489	0.323
4	0.466	0.257
5	0.493	0.388
6	0.576	0.332
7	0.473	0.372
8	0.659	0.553
9	0.645	0.494
10	0.678	0.537
11	0.592	0.473
12	0.648	0.465

Next we measure the rates of convergence of PRINCIPALS and vε-PRINCIPALS. The rates of convergence of these algorithms are assessed as

$$\tau = \lim_{t\to\infty} \tau^{(t)} = \lim_{t\to\infty} \frac{\|\mathbf{X}^{*(t)} - \mathbf{X}^{*(t-1)}\|}{\|\mathbf{X}^{*(t-1)} - \mathbf{X}^{*(t-2)}\|} \quad \text{for PRINCIPALS,}$$

$$\dot{\tau} = \lim_{t\to\infty} \dot{\tau}^{(t)} = \lim_{t\to\infty} \frac{\|\dot{\mathbf{X}}^{*(t)} - \dot{\mathbf{X}}^{*(t-1)}\|}{\|\dot{\mathbf{X}}^{*(t-1)} - \dot{\mathbf{X}}^{*(t-2)}\|} \quad \text{for v}\varepsilon\text{-PRINCIPALS.}$$

If the inequality $0 < \dot{\tau} < \tau < 1$ holds, we say that $\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$ converges faster than $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$. We also investigate the speed of convergence of vε-PRINCIPALS by

$$\dot{\rho} = \lim_{t\to\infty} \dot{\rho}^{(t)} = \lim_{t\to\infty} \frac{\|\dot{\mathbf{X}}^{*(t)} - \mathbf{X}^{*(\infty)}\|}{\|\mathbf{X}^{*(t+2)} - \mathbf{X}^{*(\infty)}\|} = 0. \tag{8}$$

From Brezinski and Zaglia (1991) if $\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$ converges to th same limit point $\mathbf{X}^{*(\infty)}$ as $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$ and Eq. (8) holds, we say that $\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$ accelerates the convergence of $\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$. The second and third columns of Table 4 provide the rates of convergence τ and $\dot{\tau}$. Fig. 5 shows the traces of $\{\tau^{(t)}\}$, $\{\dot{\tau}^{(t)}\}$ and $\{\dot{\rho}^{(t)}\}$ for $r = 2, 4, 6$ and 8. From these columns in Table 4 and Fig. 5, we see that $\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$ converges faster than $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$ in comparison between τ and $\dot{\tau}$ for each r and thus conclude that vε-PRINCIPALS significantly improves the rate of convergence of PRINCIPALS. The behavior of $\{\dot{\rho}^{(t)}\}$ for each r in Fig. 5 also indicates that vε-PRINCIPALS accelerates the convergence of $\{\mathbf{X}^{*(t)}\}$. For other values of r not drawn in the figure, $\dot{\rho}$ is reduced to zero.

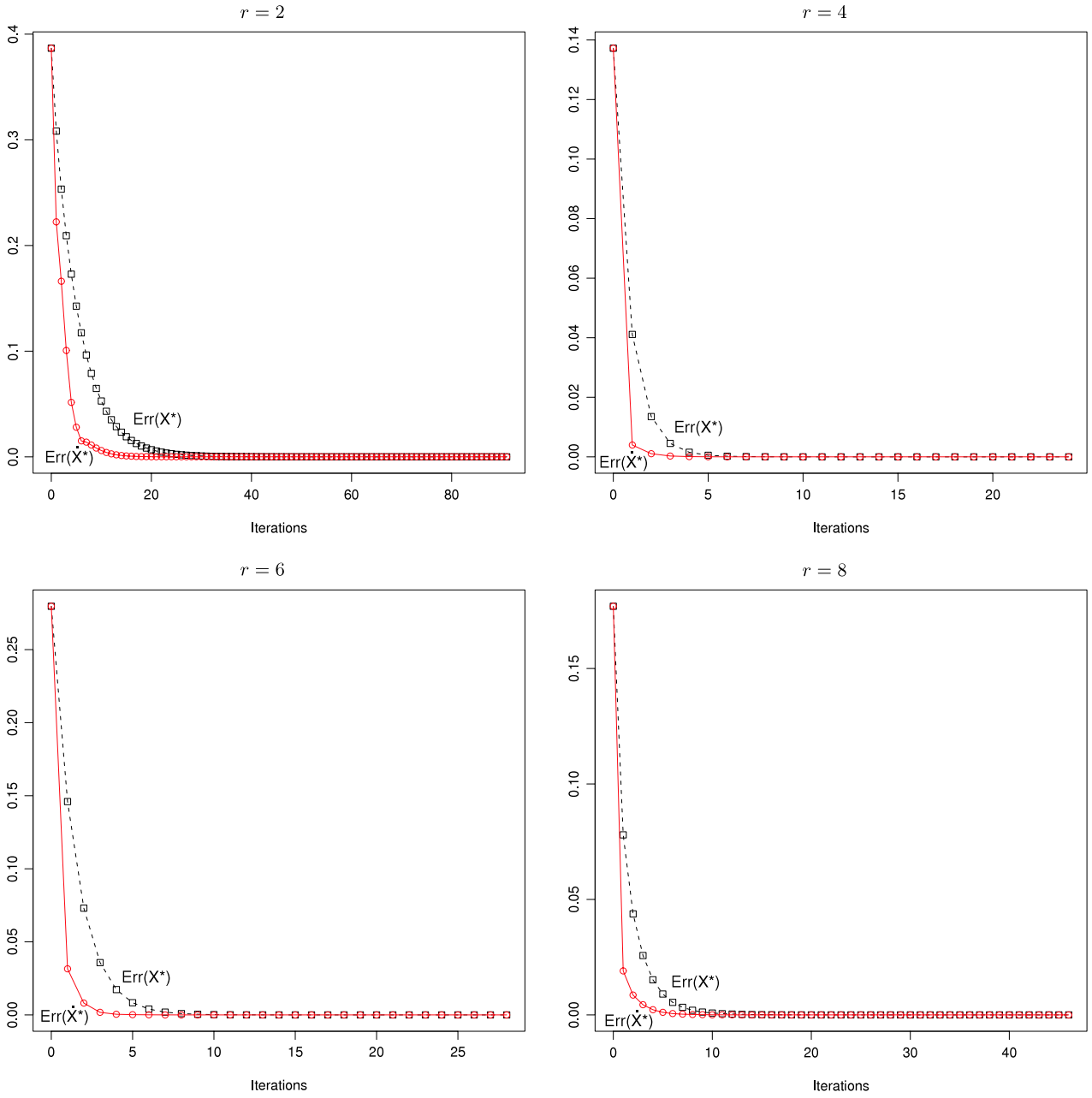

Fig. 4. Trace of $\{\mathrm{Err}(\mathbf{X}^{*(t)})\}$ and $\{\mathrm{Err}(\dot{\mathbf{X}}^{*(t)})\}$.

5.2.2. Application to variable selection problems

We consider a situation where more computation is required and investigate how much the vε acceleration algorithm improves the computational efficiency. In order to do this, we apply the algorithm to the variable selection problem in Modified PCA (M.PCA) for qualitative data (Mori et al., 2007).

Suppose that we wish to obtain the best subset of variables consisting of q variables among p original variables ($1 \leq q < p$). Let $\mathbf{X}$ be an $n \times p$ original data set including categorical variables, $\mathbf{X}_{V_1}$ an $n \times q$ submatrix of $\mathbf{X}$ and $\mathbf{X}_{V_2}$ the $n \times (p-q)$ remaining submatrix of $\mathbf{X}$. Variable selection using the criterion in M.PCA is to select the best subset of q variables which has the largest value of the proportion $P = \sum_{j=1}^{r} \lambda_j / \mathrm{tr}(\mathbf{S})$ or the *RV*-coefficient $RV = \left\{\sum_{j=1}^{r} \lambda_j^2 / \mathrm{tr}(\mathbf{S}^2)\right\}^{1/2}$, where λ_j is the j-th diagonal element of $\mathbf{D}_r$ (the j-th eigenvalue) in the generalized eigenvalue problem

$$[(\mathbf{S}_{11}^2 + \mathbf{S}_{12}\mathbf{S}_{21}) - \mathbf{D}_r\mathbf{S}_{11}]\mathbf{A} = 0, \tag{9}$$

and $\mathbf{S}$, $\mathbf{S}_{11}$, and $\mathbf{S}_{12}(= \mathbf{S}_{21}^{\top})$ are covariance matrices of $\mathbf{X}$, of $\mathbf{X}_{V_1}$, and of $\mathbf{X}_{V_1}$ and $\mathbf{X}_{V_2}$, respectively (Tanaka and Mori, 1997).

Based on the above, variable selection in M.PCA for qualitative data (Mori et al., 2007) uses the PRINCIPAL technique, in which Eq. (5) is replaced by Eq. (9), to obtain λ_j in the *Model parameter estimation step* and $\mathbf{S} = \mathbf{X}^{*\top}\mathbf{X}^{*}/n$ in the *Optimal scaling step* for each q.

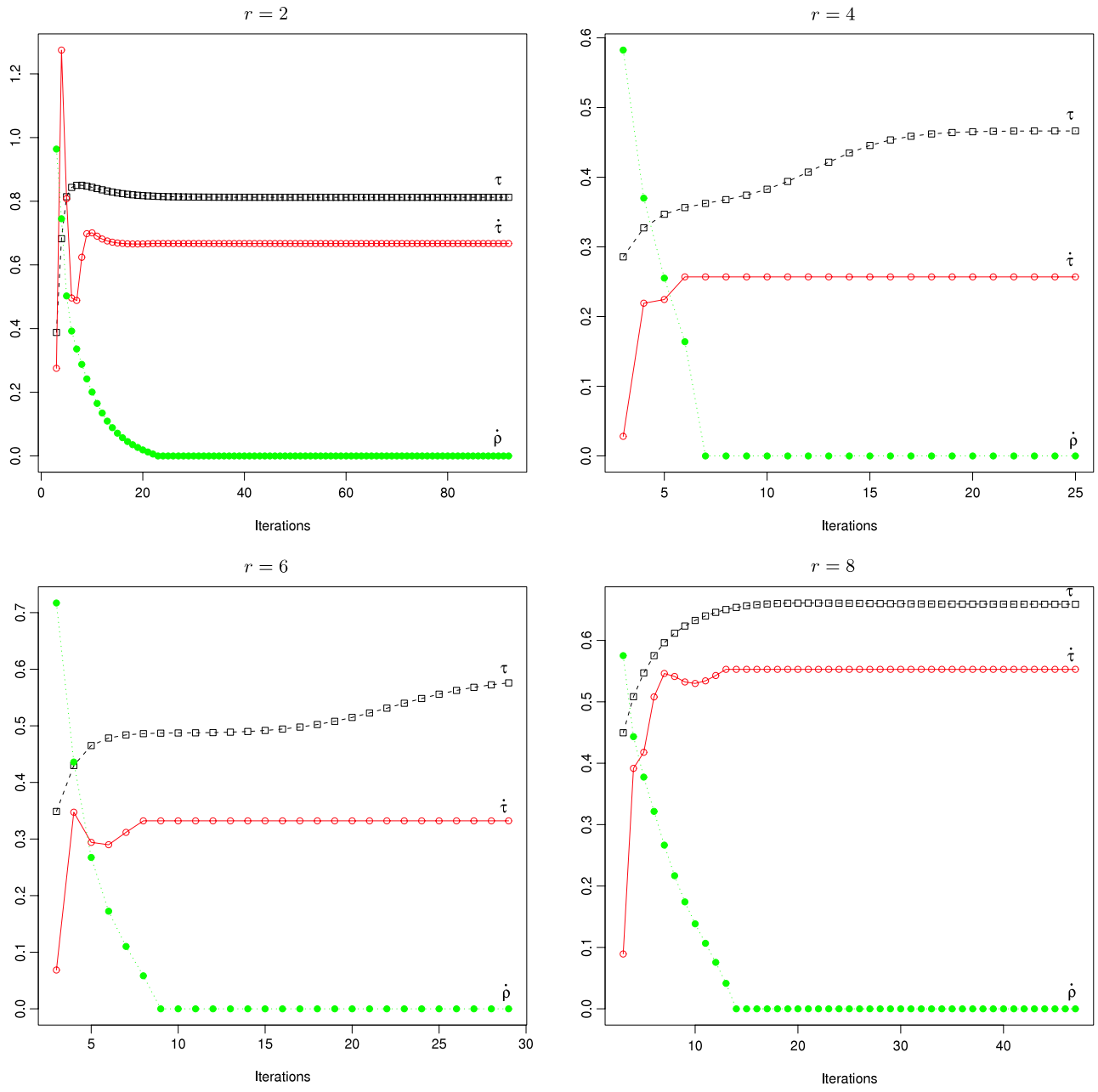

Fig. 5. Traces of $\{\tau^{(t)}\}$, $\{\dot{\tau}^{(t)}\}$ and $\{\dot{\rho}^{(t)}\}$.

Here we employ Backward elimination and Forward selection as cost-saving stepwise selection procedures in which only one variable is removed or added sequentially (see http://mo161.soci.ous.ac.jp/vaspca/theory/procedureE.html for the details of procedures). In Backward elimination, to find the best subset of $q-1$ variables, we perform M.PCA with the ALS algorithm (PRINCIPALS for M.PCA) for each of q possible subsets of the $q-1$ variables among q variables selected in the previous selection step. The total number of estimations by PRINCIPALS for M.PCA from $q=p-1$ to $q=r$ is, therefore, $p+(p-1)+\cdots+(r+1)=(p-r)(p+r+1)/2$. Similarly, in Forward selection, the total number of estimations by PRINCIPALS for M.PCA from $q=r$ to $q=p-1$ is ${}_pC_r+(p-r)+(p-(r+1))+\cdots+2={}_pC_r+(p-r-1)(p-r+2)/2$.

In the experiments, we use the same data as in Section 5.2.1 and apply both variable selection procedures using PRINCIPALS and vε-PRINCIPALS to the data. The selection criterion is the proportion P. For each r ($r=1,\ldots,12$), we measure the total number of iterations and total CPU time when all the best subsets of q variables ($q=r,\ldots,12$) are selected.

The values in the second to fifth columns of Table 5 for Backward elimination and those of Table 6 for Forward selection indicate that the number of iterations of PRINCIPALS is very large and a long time is taken for convergence, while vε-PRINCIPALS converges considerably faster than PRINCIPALS. We can see from the sixth and seventh columns in these tables that PRINCIPALS requires the number of iterations 3–4 times greater and CPU time 2.5–3.4 times longer than those of

Table 5
The numbers of iterations and CPU times of PRINCIPALS and vε-PRINCIPALS, and their speed-ups in applying the variable selection problems using Backward elimination procedure.

r	The number of iteration		CPU time		Speed-up	
	PRINCIPALS	vε-PRINCIPALS	PRINCIPALS	vε-PRINCIPALS	Iteration	CPU time
1	1203	441	51.95	27.06	2.73	1.92
2	7013	2114	325.07	108.87	3.32	2.99
3	5331	1437	267.89	77.85	3.71	3.44
4	2922	982	127.11	50.87	2.98	2.50
5	2730	858	118.80	44.87	3.18	2.65
6	2645	842	115.06	43.70	3.14	2.63
7	3121	964	140.45	50.57	3.24	2.78
8	3320	904	153.15	47.53	3.67	3.22
9	2523	721	114.42	37.61	3.50	3.04
10	1707	507	75.43	26.32	3.37	2.87
11	1183	318	52.20	16.63	3.72	3.14
12	506	129	22.17	6.97	3.92	3.18

Table 6
The numbers of iterations and CPU times of PRINCIPALS and vε-PRINCIPALS, and their speed-ups in applying the variable selection problems using Forward selection procedure.

r	The number of iteration		CPU time		Speed-up	
	PRINCIPALS	vε-PRINCIPALS	PRINCIPALS	vε-PRINCIPALS	Iteration	CPU time
1	1 255	423	53.60	24.96	2.97	2.15
2	5 718	1 675	260.86	89.47	3.41	2.92
3	16 646	5 085	742.15	261.64	3.27	2.84
4	39 891	11 946	1780.50	612.57	3.34	2.91
5	81 640	23 711	3747.87	1229.88	3.44	3.05
6	123 543	34 670	5886.94	1834.72	3.56	3.21
7	126 576	35 396	6072.00	1888.88	3.58	3.21
8	91 280	25 619	4352.79	1365.93	3.56	3.19
9	47 913	13 538	2345.94	738.53	3.54	3.18
10	15 691	4 624	714.00	243.09	3.39	2.94
11	3 737	1 069	167.91	56.46	3.50	2.97
12	506	129	22.38	7.07	3.92	3.17

vε-PRINCIPALS. The values of the iteration and CPU time speed-ups demonstrate that the vε acceleration algorithm works well to accelerate the convergence of $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$ and consequently results in greatly reduced computation times.

6. Conclusion

In this paper, we presented vε-PCA.ALS that accelerates the convergence of PCA.ALS by using the vε algorithm. The algorithm generates the vε accelerated sequence $\{\dot{\mathbf{X}}^{*(t)}\}$ using $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$ but it does not modify the estimation equations of PCA.ALS. Therefore the algorithm enables an acceleration of the convergence of PCA.ALS while still preserving the stable convergence property of PCA.ALS. The vε algorithm in itself is a fairly simple computational procedure and, at each iteration, it requires only $O(np)$ arithmetic operations. For each iteration, the computational complexity of the vε algorithm may be less expensive than that for computing a matrix inversion and for solving the eigenvalue problem in PCA.ALS.

The most appealing points of vε-PCA.ALS are that, if $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$ convergences to a limit point $\mathbf{X}^{*(\infty)}$, then $\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$ converges to $\mathbf{X}^{*(\infty)}$ of $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$ and its speed of convergence is faster than that of PCA.ALS. In the application of vε-PCA.ALS, we do not have to take into account the structure of optimally scaled data sets. The numerical experiments employing simulated and real data demonstrated that vε accelerated PRINCIPALS improves the speed of convergence of ordinary PRINCIPALS and significantly speeds up the convergence of $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$ in terms of the number of iterations and the computation time. Moreover, in the variable selection problems for finding a suitable variable set using Backward elimination and Forward selection procedures, the vε acceleration enables greatly reducing computation times.

We intend to evaluate theoretically the convergence properties of vε-PCA.ALS. As listed in Young (1981) and Krijnen (2006), there exist many other ALS types of algorithms. We attempt to speed up the convergence of their ALS algorithms by incorporating the vε acceleration.

Acknowledgements

The authors would like to thank the editor and two referees whose valuable comments and kind suggestions that led to an improvement of this paper. This research is supported by the Japan Society for the Promotion of Science (JSPS), Grant-in-Aid for Scientific Research (C), No 20500263.

Appendix. The vε algorithm

Let $\mathbf{Y}^{(t)}$ denote a vector of dimensionality d that converges to a vector $\mathbf{Y}^{(\infty)}$ as $t \to \infty$. Let the inverse $[\mathbf{Y}]^{-1}$ of a vector $\mathbf{Y}$ be defined by

$$[\mathbf{Y}]^{-1} = \frac{\mathbf{Y}}{\|\mathbf{Y}\|^2},$$

where $\|\mathbf{Y}\|$ is the Euclidean norm of $\mathbf{Y}$.

In general, the vε algorithm for a sequence $\{\mathbf{Y}^{(t)}\}_{t\geq 0}$ starts with

$$\varepsilon^{(t,-1)} = 0, \qquad \varepsilon^{(t,0)} = \mathbf{Y}^{(t)},$$

and then generates a vector $\varepsilon^{(t,k+1)}$ by

$$\varepsilon^{(t,k+1)} = \varepsilon^{(t+1,k-1)} + \left[\varepsilon^{(t+1,k)} - \varepsilon^{(t,k)}\right]^{-1}, \quad k = 0, 1, 2, \ldots. \tag{10}$$

For practical implementation, we apply the vε algorithm for $k = 1$ to accelerate the convergence of $\{\mathbf{Y}^{(t)}\}_{t\geq 0}$. From Eq. (10), we have

$$\begin{aligned}
\varepsilon^{(t,2)} &= \varepsilon^{(t+1,0)} + \left[\varepsilon^{(t+1,1)} - \varepsilon^{(t,1)}\right]^{-1} \quad \text{for } k = 1,\\
\varepsilon^{(t,1)} &= \varepsilon^{(t+1,-1)} + \left[\varepsilon^{(t+1,0)} - \varepsilon^{(t,0)}\right]^{-1} = \left[\varepsilon^{(t+1,0)} - \varepsilon^{(t,0)}\right]^{-1} \quad \text{for } k = 0.
\end{aligned}$$

Then the vector $\varepsilon^{(t,2)}$ becomes as follows:

$$\begin{aligned}
\varepsilon^{(t,2)} &= \varepsilon^{(t+1,0)} + \left[\left[\varepsilon^{(t,0)} - \varepsilon^{(t+1,0)}\right]^{-1} + \left[\varepsilon^{(t+2,0)} - \varepsilon^{(t+1,0)}\right]^{-1}\right]^{-1}\\
&= \mathbf{Y}^{(t+1)} + \left[\left[\mathbf{Y}^{(t)} - \mathbf{Y}^{(t+1)}\right]^{-1} + \left[\mathbf{Y}^{(t+2)} - \mathbf{Y}^{(t+1)}\right]^{-1}\right]^{-1}.
\end{aligned}$$

References

Brezinski, C., Zaglia, M., 1991. Extrapolation Methods: Theory and Practice. Elsevier Science Ltd., North-Holland, Amsterdam.

Gifi, A., 1989. Algorithm descriptions for ANACOR, HOMALS, PRINCIPALS, and OVERALS. Report RR 89-01. Department of Data Theory, University of Leiden, Leiden.

Kiers, H.A.L., 2002. Setting up alternating least squares and iterative majorization algorithm for solving various matrix optimization problems. Computational Statistics and Data Analysis 41, 157–170.

Krijnen, W.P., 2006. Convergence of the sequence of parameters generated by alternating least squares algorithms. Computational Statistics and Data Analysis 51, 481–489.

Kuroda, M., Sakakihara, M., 2006. Accelerating the convergence of the EM algorithm using the vector epsilon algorithm. Computational Statistics and Data Analysis 51, 1549–1561.

Michailidis, G., de Leeuw, J., 1998. The Gifi system of descriptive multivariate analysis. Statistical Science 13, 307–336.

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

Mori, Y., Matsumoto, Y., Iizuka, M., Tanaka, Y., 2007. A variable selection in modified principal component analysis for qualitative data. In: The 56th Session of the International Statistical Institute. Abstract Book, 337, Proceedings CD-ROM.

R Development Core Team, 2008. 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.

Tanaka, Y., Mori, Y., 1997. Principal component analysis based on a subset of variables: variable selection and sensitivity analysis. The American Journal of Mathematical and Management Sciences 17, 61–89.

Young, F.W., 1981. Quantitative analysis of qualitative data. Psychometrika 46, 357–388.

Young, F.W., Takane, Y., 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.

Wang, M., Kuroda, M., Sakakihara, M., Geng, Z., 2008. Acceleration of the EM algorithm using the vector epsilon algorithm. Computational Statistics 23, 469–486.

Wynn, P., 1962. Acceleration techniques for iterated vector and matrix problems. Mathematics of Computation 16, 301–322.