We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

7

Acceleration of Convergence of the Alternating Least Squares Algorithm for Nonlinear Principal Components Analysis

Masahiro Kuroda[1], Yuichi Mori[1], Masaya Iizuka[2] and Michio Sakakihara[1]
[1]*Okayama University of Science*
[2]*Okayama University*
Japan

1. Introduction

Principal components analysis (PCA) is a popular descriptive multivariate method for handling quantitative data. In PCA of a mixture of quantitative and qualitative data, it requires quantification of qualitative data to obtain optimal scaling data and use ordinary PCA. The extended PCA including such quantification is called *nonlinear PCA*, see Gifi [Gifi, 1990]. The existing algorithms for nonlinear PCA are PRINCIPALS of Young et al. [Young et al., 1978] and PRINCALS of Gifi [Gifi, 1990] in which the alternating least squares (ALS) algorithm is utilized. The algorithm alternates between quantification of qualitative data and computation of ordinary PCA of optimal scaling data.

In the application of nonlinear PCA for very large data sets and variable selection problems, many iterations and much computation time may be required for convergence of the ALS algorithm, because its speed of convergence is linear. Kuroda et al. [Kuroda et al., 2011] proposed an acceleration algorithm for speeding up the convergence of the ALS algorithm using the vector ε (vε) algorithm of Wynn [Wynn, 1962]. During iterations of the vε accelerated ALS algorithm, the vε algorithm generates an accelerated sequence of optimal scaling data estimated by the ALS algorithm. Then the vε accelerated sequence converges faster than the original sequence of the estimated optimal scaling data. In this paper, we use PRINCIPALS as the ALS algorithm for nonlinear PCA and provide the vε acceleration for PRINCIPALS (vε-PRINCIPALS). The computation steps of PRINCALS are given in Appendix A. As shown in Kuroda et al. [Kuroda et al., 2011], the vε acceleration is applicable to PRINCALS.

The paper is organized as follows. We briefly describe nonlinear PCA of a mixture of quantitative and qualitative data in Section 2, and describe PRINCIPALS for finding least squares estimates of the model and optimal scaling parameters in Section 3. Section 4 presents the procedure of vε-PRINCIPALS that adds the vε algorithm to PRINCIPALS for speeding up convergence and demonstrate the performance of the vε acceleration using numerical experiments. In Section 5, we apply vε-PRINCIPALS to variable selection in nonlinear PCA. Then we utilize modified PCA (M.PCA) approach of Tanaka and Mori [Tanaka and Mori, 1997] for variable selection problems and give the variable selection procedures in M.PCA of qualitative data. Numerical experiments examine the the performance and properties of vε-PRINCIPALS. In Section 6, we present our concluding remarks.

2. Nonlinear principal components analysis

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 $\mathbf{X} = (\mathbf{X}_1\ \mathbf{X}_2\ \cdots\ \mathbf{X}_p)$ 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 $\mathbf{Z} = (\mathbf{Z}_1\ \mathbf{Z}_2\ \cdots\ \mathbf{Z}_r)$ 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.

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 are a mixture of quantitative and qualitative data, 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 $\mathbf{X}^* = (\mathbf{X}_1^*\ \mathbf{X}_2^*\ \cdots\ \mathbf{X}_p^*)$ be an $n \times p$ matrix of optimally scaled observations to satisfy restrictions

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

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

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

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

3. Alternating least squares algorithm for nonlinear 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 [Krijnen, 2006] gave sufficient conditions for convergence of the ALS algorithm and discussed convergence properties in its application to several statistical models. Kiers [Kiers, 2002] described setting up the ALS and iterative majorization algorithms for solving various matrix optimization problems.

PRINCIPALS

PRINCIPALS proposed by Young et al. [Young et al., 1978] is a method for utilizing the ALS algorithm for nonlinear PCA of a mixture of quantitative and qualitative data. PRINCIPALS alternates between ordinary PCA and optimal scaling, and minimizes θ^* defined by Equation (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 Equation (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.

4. The vε acceleration of the ALS algorithm

We briefly introduce the vε algorithm of Wynn [Wynn, 1962] used in the acceleration of the ALS algorithm. 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 [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. [Wang et al., 2008] studied the convergence properties of the ε-accelerated EM algorithm.

Let $\{\mathbf{Y}^{(t)}\}_{t\geq 0} = \{\mathbf{Y}^{(0)}, \mathbf{Y}^{(1)}, \mathbf{Y}^{(2)}, \dots\}$ be a linear convergent sequence generated by an iterative computational procedure and let $\{\dot{\mathbf{Y}}^{(t)}\}_{t\geq 0} = \{\dot{\mathbf{Y}}^{(0)}, \dot{\mathbf{Y}}^{(1)}, \dot{\mathbf{Y}}^{(2)}, \dots\}$ be the accelerated

sequence of $\{\mathbf{Y}^{(t)}\}_{t\geq 0}$. Then the v$\varepsilon$ algorithm generates $\{\dot{\mathbf{Y}}^{(t)}\}_{t\geq 0}$ by using

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

where $[\mathbf{Y}]^{-1} = \mathbf{Y}\,/||\mathbf{Y}||^2$ and $||\mathbf{Y}||$ is the Euclidean norm of $\mathbf{Y}$. For the detailed derivation of Equation (6), see Appendix B. 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 $\{\mathbf{Y}^{(t)}\}_{t\geq 0}$.

We assume that $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$ generated by PRINCIPALS converges to a limit point $\mathbf{X}^{*(\infty)}$. Then vε-PRINCIPALS 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 PRINCIPALS. The general procedure of vε-PRINCIPALS iterates the following two steps:

- *PRINCIPALS 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:

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

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

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

  where δ is a desired accuracy.

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

vε-PRINCIPALS 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-PRINCIPALS 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 PRINCIPALS.

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 *PRINCIPALS step*. Thus vε-PRINCIPALS speeds up the convergence of $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$ without affecting the convergence properties of ordinary PRINCIPALS.

Numerical experiments 1: Comparison of the number of iterations and CPU time

We study how much faster vε-PRINCIPALS converges than ordinary PRINCIPALS. All computations are performed with the statistical package `R` [R Development Core Team, 2008] executing on Intel Core i5 3.3 GHz with 4 GB of memory. CPU times (in seconds) taken are measured by the function `proc.time`[1]. For all experiments, δ for convergence

[1] Times are typically available to 10 msec.

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 Equation (4). The maximum number of iterations is also set to 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 taken for $r = 3$. The procedure is replicated 50 times.

Table 1 is summary statistics of the numbers of iterations and CPU times of PRINCIPALS and vε-PRINCIPALS from 50 simulated data. Figure 1 shows the scatter plots of the number of iterations and CPU time. The values of the second to fifth columns of the table and the figure show that PRINCIPALS requires more iterations and takes a longer computation time than vε-PRINCIPALS. The values of the sixth and seventh columns in the table are summary statistics of 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 can see from the values of the iteration and CPU time speed-ups that vε-PRINCIPALS converges 3.23 times in terms of the mean number of iterations and 2.92 times in terms of the mean CPU time faster than PRINCIPALS. Figure 2 shows the boxplots of the iteration and CPU time speed-ups. Table 1 and Figure 2 show that vε-PRINCIPALS well accelerates the convergence of $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$.

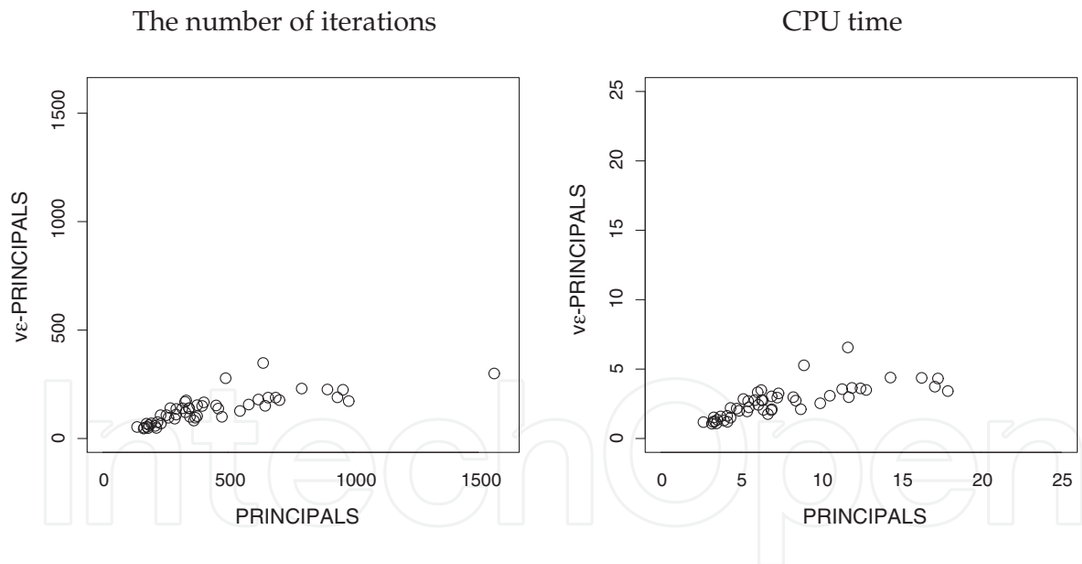

Fig. 1. Scatter plots of the number iterations and CPU time from 50 simulated data.

Figure 3 is the scatter plots of iteration and CPU time speed-ups for the number of iterations of PRINCIPALS. The figure demonstrates that the vε acceleration speeds up greatly the convergence of $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$ and its speed of convergence is faster for the larger number of iterations of PRINCIPALS. For more than 400 iterations of PRINCIPALS, the speed of the vε acceleration is faster 3 times more than that of PRINCIPALS and the maximum values of both speed-ups are for around 1,000 iterations of PRINCIPALS. The advantage of the vε acceleration is very obvious.

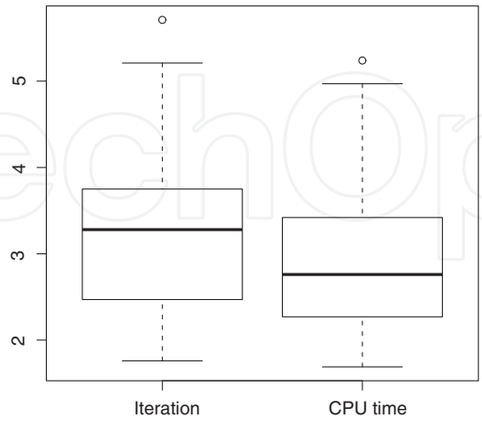

Fig. 2. Boxplots of iteration and CPU time speed-ups from 50 simulated data.

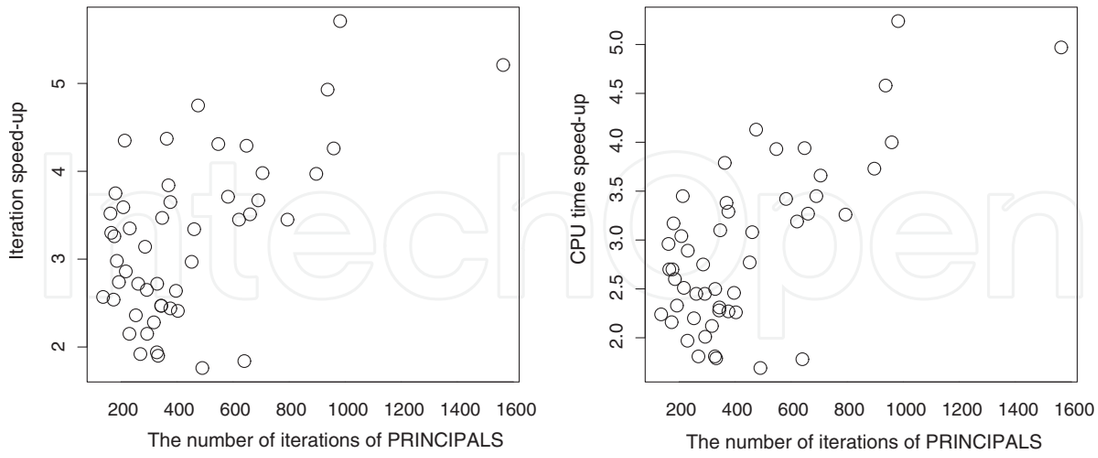

Fig. 3. Scatter plots of iteration and CPU time speed-ups for the number of iterations of PRINCIPALS from 50 simulated data.

	PRINCIPALS		vε-PRINCIPALS		Speed-up	
	Iteration	CPU time	Iteration	CPU time	Iteration	CPU time
Minimum	136.0	2.64	46.0	1.07	1.76	1.69
1st Quartile	236.5	4.44	85.0	1.81	2.49	2.27
Median	345.5	6.37	137.0	2.72	3.28	2.76
Mean	437.0	8.02	135.0	2.70	3.23	2.92
3rd Quartile	573.2	10.39	171.2	3.40	3.74	3.41
Maximum	1564.0	28.05	348.0	6.56	5.71	5.24

Table 1. Summary statistics of the numbers of iterations and CPU times of PRINCIPALS and vε-PRINCIPALS and iteration and CPU time speed-ups from 50 simulated data.

Numerical experiments 2: Studies of convergence

We introduce the result of studies of convergence of vε-PRINCIPALS from Kuroda et al. [Kuroda et al., 2011]. The data set used in the experiments is obtained in teacher evaluation by students and consists of 56 observations on 13 variables with 5 levels each; the lowest evaluation level is 1 and the highest 5.

The rates of convergence of these algorithms are assessed as

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

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

If the inequality $0 < \dot{\tau} < \tau < 1$ holds, we say that $\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$ converges faster than $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$. Table 2 provides the rates of convergence τ and $\dot{\tau}$ for each r. We see from the table 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 speed of convergence of vε-PRINCIPALS is investigate by

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

Table 2. Rates of convergence τ and $\dot{\tau}$ of PRINCIPALS to vε-PRINCIPALS.

$$\dot{\rho} = \lim_{t\to\infty} \dot{\rho}^{(t)} = \lim_{t\to\infty} \frac{\|\dot{\mathbf{X}}^{*(t)} - \mathbf{X}^{*(\infty)}\|}{\|\mathbf{X}^{*(t+2)} - \mathbf{X}^{*(\infty)}\|} = 0. \tag{7}$$

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 Equation (7) holds, we say that $\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$ accelerates the convergence of $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$. See Brezinski and Zaglia [Brezinski and Zaglia, 1991]. In the experiments, $\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$ converges to the final value of $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$ and $\dot{\rho}$ is reduced to zero for all r. We see from the results that vε-PRINCIPALS accelerates the convergence of $\{\mathbf{X}^{*(t)}\}$.

5. Variable selection in nonlinear PCA: Modified PCA approach

In the analysis of data with large numbers of variables, a common objective is to reduce the dimensionality of the data set. PCA is a popular dimension-reducing tool that replaces the variables in the data set by a smaller number of derived variables. However, for example, in PCA of a data set with a large number of variables, the result may not be easy to interpret. One way to give a simple interpretation of principal components is to select a subset of variables that best approximates all the variables. Various variable selection criteria in PCA has been proposed by Jolliffe [Jolliffe, 1972], McCabe [McCabe, 1984], Robert and Escoufier [Robert and Escoufier, 1976], Krzanowski [Krzanowski, 1987]. Al-Kandari et al. [Al-Kandari et al., 2001; Al-Kandari et al., 2005] gave guidelines as to the types of data for which each variable selection criteria is useful. Cadima et al. [Cadima et al., 2004] reported computational experiments carried out with several heuristic algorithms for the optimization problems resulting from the variable selection criteria in PCA found in the above literature.

Tanaka and Mori [Tanaka and Mori, 1997] proposed modified PCA (M.PCA) for deriving principal components which are computed by using only a selected subset of variables but which represent all the variables including those not selected. Since M.PCA includes variable selection procedures in the analysis, its criteria can be used directly to find a reasonable subset of variables. Mori et al. [Mori et al., 1997] extended M.PCA to qualitative data and provided variable selection procedures, in which the ASL algorithm is utilized.

5.1 Formulation of modified PCA

M.PCA derives principal components which are computed as linear combinations of a subset of variables but which can reproduce all the variables very well. Let $\mathbf{X}$ be decomposed into an $n \times q$ submatrix $\mathbf{X}_{V_1}$ and an $n \times (p-q)$ remaining submatrix $\mathbf{X}_{V_2}$. Then M.PCA finds r linear combinations $\mathbf{Z} = \mathbf{X}_{V_1}\mathbf{A}$. The matrix $\mathbf{A}$ consists of the eigenvectors associated with the largest r eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r$ and is obtained by solving the eigenvalue problem:

$$[(\mathbf{S}_{11}^2 + \mathbf{S}_{12}\mathbf{S}_{21}) - \mathbf{D}\mathbf{S}_{11}]\mathbf{A} = 0, \tag{8}$$

where $\mathbf{S} = \begin{pmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} \\ \mathbf{S}_{21} & \mathbf{S}_{22} \end{pmatrix}$ is the covariance matrix of $\mathbf{X} = (\mathbf{X}_{V_1}, \mathbf{X}_{V_2})$ and $\mathbf{D}$ is a $q \times q$ diagonal matrix of eigenvalues. A best subset of q variables has the largest value of the proportion $P = \sum_{j=1}^{r} \lambda_j / \mathrm{tr}(\mathbf{S})$ or the RV-coefficient $RV = \left\{\sum_{j=1}^{r} \lambda_j^2 / \mathrm{tr}(\mathbf{S}^2)\right\}^{1/2}$. Here we use P as variable selection criteria.

5.2 Variable selection procedures

In order to find a subset of q variables, we employ Backward elimination and Forward selection of Mori et al. [Mori et al., 1998; Mori et al., 2006] as cost-saving stepwise selection procedures in which only one variable is removed or added sequentially.

[Backward elimination]

Stage A: *Initial fixed-variables stage*

A-1 Assign q variables to subset $\mathbf{X}_{V_1}$, usually $q := p$.

A-2 Solve the eigenvalue problem (8).

A-3 Look carefully at the eigenvalues, determine the number r of principal components to be used.

A-4 Specify kernel variables which should be involved in $\mathbf{X}_{V_1}$, if necessary. The number of kernel variables is less than q.

Stage B: *Variable selection stage (Backward)*

B-1 Remove one variable from among q variables in $\mathbf{X}_{V_1}$, make a temporary subset of size $q-1$, and compute P based on the subset. Repeat this for each variable in $\mathbf{X}_{V_1}$, then obtain q values on P. Find the best subset of size $q-1$ which provides the largest P among these q values and remove the corresponding variable from the present $\mathbf{X}_{V_1}$. Put $q := q-1$.

B-2 If P or q is larger than preassigned values, go to **B-1**. Otherwise stop.

[Forward selection]

Stage A: *Initial fixed-variables stage*

A-1 $\sim$ **A-3** Same as A-1 to A-3 in Backward elimination.

A-4 Redefine q as the number of kernel variables (here, $q \geq r$). If you have kernel variables, assign them to $\mathbf{X}_{V_1}$. If not, put $q := r$, find the best subset of q variables which provides the largest P among all possible subsets of size q and assign it to $\mathbf{X}_{V_1}$.

Stage B: *Variable selection stage (Forward)*

B-1 Adding one of the $p-q$ variables in $\mathbf{X}_{V_2}$ to $\mathbf{X}_{V_1}$, make a temporary subset of size $q+1$ and obtain P. Repeat this for each variable in $\mathbf{X}_{V_2}$, then obtain $p-q$ Ps. Find the best subset of size $q+1$ which provides the largest (or smallest) P among the $p-q$ Ps and add the corresponding variable to the present subset of $\mathbf{X}_{V_1}$. Put $q := q+1$.

B-2 If the P or q are smaller (or larger) than preassigned values, go back to **B-1**. Otherwise stop.

In Backward elimination, to find the best subset of $q-1$ variables, we perform 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 for M.PCA from $q = p-1$ to $q = r$ is therefore large, i.e., $p+(p-1)+\cdots+(r+1) = (p-r)(p+r+1)/2$. In Forward selection, the total number of estimations for M.PCA from $q = r$ to $q = p-1$ is ${}_pC_r + (p-r) + (p-(r+1)) + \cdots + 2 = {}_pC_r + (p-r-1)(p-r+2)/2$.

Numerical experiments 3: Variable selection in M.PCA for simulated data

We apply PRINCIPALS and vε-PRINCIPALS to variable selection in M.PCA of qualitative data using simulated data consisting of 100 observations on 10 variables with 3 levels.

Table 3 shows the number of iterations and CPU time taken by two algorithms for finding a subset of q variables based on 3 ($= r$) principal components. The values of the second to fifth columns in the table indicate that the number of iterations of PRINCIPALS is very large and a long computation time is taken for convergence, while vε-PRINCIPALS converges considerably faster than PRINCIPALS. We can see from the sixth and seventh columns in the

table that vε-PRINCIPALS requires the number of iterations 3 - 5 times smaller and CPU time 2 - 5 times shorter than vε-PRINCIPALS. In particular, the vε acceleration effectively works to accelerate the convergence of $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$ for the larger number of iterations of PRINCIPALS.

(a) Backward elimination

	PRINCIPALS		vε-PRINCIPALS		Speed-up	
q	Iteration	CPU time	Iteration	CPU time	Iteration	CPU time
10	141	1.70	48	0.68	2.94	2.49
9	1,363	17.40	438	6.64	3.11	2.62
8	1,620	20.19	400	5.98	4.05	3.37
7	1,348	16.81	309	4.80	4.36	3.50
6	4,542	53.72	869	11.26	5.23	4.77
5	13,735	159.72	2,949	35.70	4.66	4.47
4	41,759	482.59	12,521	148.13	3.34	3.26
3	124	1.98	44	1.06	2.82	1.86
Total	64,491	752.40	17,530	213.57	3.68	3.52

(b) Forward selection

	PRINCIPALS		vε-PRINCIPALS		Speed-up	
q	Iteration	CPU time	Iteration	CPU time	Iteration	CPU time
3	4,382	67.11	1442	33.54	3.04	2.00
4	154,743	1,786.70	26,091	308.33	5.93	5.79
5	13,123	152.72	3,198	38.61	4.10	3.96
6	3,989	47.02	1,143	14.24	3.49	3.30
7	1,264	15.27	300	4.14	4.21	3.69
8	340	4.38	108	1.70	3.15	2.58
9	267	3.42	75	1.17	3.56	2.93
10	141	1.73	48	0.68	2.94	2.54
Total	178,249	2,078.33	32,405	402.40	5.50	5.16

Table 3. The numbers of iterations and CPU times of PRINCIPALS and vε-PRINCIPALS and their speed-ups in application to variable selection for finding a subset of q variables using simulated data.

The last row in Table 3 shows the total number of iterations and total CPU time for selecting 8 subsets for $q = 3, \ldots, 10$. When searching the best subset for each q, PRINCIPALS requires 64,491 iterations in Backward elimination and 178,249 iterations in Forward selection, while vε-PRINCIPALS finds the subsets after 17,530 and 32,405 iterations, respectively. These values show that the computation times by vε-PRINCIPALS are reduced to only $28\%(= 1/3.52)$ and $19\% = (1/5.16)$ of those of ordinary PRINCIPALS. The iteration and CPU time speed-ups given in the sixth and seventh columns of the table demonstrate that the vε acceleration works well to speed up the convergence of $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$ and consequently results in greatly reduced computation times in variable selection problems.

Numerical experiments 4: Variable selection in M.PCA for real data

We consider the variable selection problems in M.PCA of qualitative data to mild distribution of consciousness (MDOC) data from Sano et al. [Sano et al. 1977]. MDOC is the data matrix of 87 individuals on 23 variables with 4 levels. In the variable selection problem, we select a suitable subset based on 2 ($= r$) principal components.

Table 4 summarizes the results of variable selection using Backward elimination and Forward selection procedures for finding a subset of q variables. We see from the last row of the table

(a) Backward elimination

	PRINCIPALS		vε-PRINCIPALS		Speed-up		
q	Iteration	CPU time	Iteration	CPU time	Iteration	CPU time	P
23	36	1.39	10	0.65	3.60	2.13	0.694
22	819	32.42	231	15.40	3.55	2.11	0.694
21	779	30.79	221	14.70	3.52	2.10	0.693
20	744	29.37	212	14.05	3.51	2.09	0.693
19	725	28.43	203	13.41	3.57	2.12	0.692
18	705	27.45	195	12.77	3.62	2.15	0.692
17	690	26.67	189	12.25	3.65	2.18	0.691
16	671	25.73	180	11.61	3.73	2.22	0.690
15	633	24.26	169	10.85	3.75	2.24	0.689
14	565	21.79	153	10.02	3.69	2.17	0.688
13	540	20.69	147	9.48	3.67	2.18	0.687
12	498	19.09	132	8.64	3.77	2.21	0.686
11	451	17.34	121	7.95	3.73	2.18	0.684
10	427	16.29	117	7.46	3.65	2.18	0.682
9	459	16.99	115	7.05	3.99	2.41	0.679
8	419	15.43	106	6.42	3.95	2.40	0.676
7	382	14.02	100	5.89	3.82	2.38	0.673
6	375	13.51	96	5.41	3.91	2.50	0.669
5	355	12.58	95	5.05	3.74	2.49	0.661
4	480	16.11	117	5.33	4.10	3.02	0.648
3	2,793	86.55	1,354	43.48	2.06	1.99	0.620
2	35	1.92	10	1.34	3.50	1.43	0.581
Total	13,581	498.82	4,273	229.20	3.18	2.18	

(b) Forward selection

	PRINCIPALS		vε-PRINCIPALS		Speed-up		
q	Iteration	CPU time	Iteration	CPU time	Iteration	CPU time	P
2	3,442	176.76	1,026	119.07	3.35	1.48	0.597
3	5,389	170.82	1,189	44.28	4.53	3.86	0.633
4	1,804	60.96	429	20.27	4.21	3.01	0.650
5	1,406	48.53	349	17.41	4.03	2.79	0.662
6	1,243	43.25	305	15.75	4.08	2.75	0.668
7	1,114	39.03	278	14.61	4.01	2.67	0.674
8	871	31.35	221	12.39	3.94	2.53	0.677
9	789	28.57	202	11.52	3.91	2.48	0.680
10	724	26.32	187	10.74	3.87	2.45	0.683
11	647	23.69	156	9.39	4.15	2.52	0.685
12	578	21.30	142	8.60	4.07	2.48	0.687
13	492	18.39	125	7.76	3.94	2.37	0.688
14	432	16.23	110	6.94	3.93	2.34	0.689
15	365	13.91	95	6.13	3.84	2.27	0.690
16	306	11.80	80	5.30	3.83	2.22	0.691
17	267	10.32	71	4.66	3.76	2.21	0.691
18	226	8.77	60	3.96	3.77	2.21	0.692
19	193	7.48	51	3.39	3.78	2.21	0.692
20	152	5.91	40	2.65	3.80	2.23	0.693
21	108	4.26	30	2.00	3.60	2.13	0.693
22	72	2.85	20	1.33	3.60	2.14	0.694
23	36	1.39	10	0.66	3.60	2.11	0.694
Total	20,656	771.88	5,176	328.81	3.99	2.35	

Table 4. The numbers of iterations and CPU times of PRINCIPALS and vε-PRINCIPALS, their speed-ups and P in application to variable selection for finding a subset of q variables using MDOC.

that the iteration speed-ups are 3.18 in Backward elimination and 3.99 in Forward selection and thus vε-PRINCIPALS well accelerates the convergence of $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$. The CPU time speed-ups are 2.18 in Backward elimination and 2.35 in Forward selection, and are not as large as the iteration speed-ups. The computation time per iteration of vε-PRINCIPALS is greater than that of PRINCIPALS due to computation of the *Acceleration step*. Therefore, for the smaller number of iterations, the CPU time of vε-PRINCIPALS is almost same as or may be longer than that of PRINCIPALS. For example, in Forward selection for $q = 2$, PRINCIPALS converges in almost cases after less than 15 iterations and then the CPU time speed-up is 1.48.

The proportion P in the eighth column of the table indicates the variation explained by the first 2 principal components for the selected q variables. Iizuka et al. [Iizuka et al., 2003] selected the subset of 6 variables found by either procedures as a best subset, since P slightly changes until $q = 6$ in Backward elimination and after $q = 6$ in Forward selection.

6. Concluding remarks

In this paper, we presented vε-PRINCIPALS that accelerates the convergence of PRINCIPALS 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 in PRINCIPALS. Therefore the algorithm enables an acceleration of the convergence of PRINCIPALS, while still preserving the stable convergence property of PRINCIPALS. 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 PRINCIPALS.

The most appealing points of the vε algorithm are that, if an original sequence converges to a limit point then the accelerated sequence converges to the same limit point as the original sequence and its speed of convergence is faster than the original sequence. In all the numerical experiments, the vε accelerated sequence $\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$ converges to the final value of $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$ after the significantly fewer number of iterations than that of PRINCIPALS.

The numerical experiments employing simulated data in Section 4 demonstrated that vε acceleration for PRINCIPALS significantly speeds up the convergence of $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$ in terms of the number of iterations and the computation time. In particular, the vε acceleration effectively works to speed up the convergence for the larger number of iterations of PRINCIPALS. Furthermore, we evaluate the performance of the vε acceleration for PRINCIPALS by applying to variable selection in M.PCA of qualitative data. Numerical experiments using simulated and real data showed that vε-PRINCIPALS improves the speed of convergence of ordinary PRINCIPALS and enables greatly the reduction of computation times in the variable selection for finding a suitable variable set using Backward elimination and Forward selection procedures. The results indicate that the vε acceleration well works in saving the computational time in variable selection problems.

The computations of variable selection in M.PCA of qualitative data are partially performed by the statistical package VASpca(VAriable Selection in principal component analysis) that was developed by Mori, Iizuka, Tarumi and Tanaka in 1999 and can be obtained from Mori's website in Appendix C. We will provide VASpca using vε-PRINCIPALS as the iterative algorithm for PCA and M.PCA of qualitative data.

7. Acknowledgment

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.

8. Appendix A: PRINCALS

PRINCALS by Gifi [Gifi, 1990] 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

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

under the restriction

$$\mathbf{Z}^\top \mathbf{1}_n = \mathbf{0}_r \qquad \text{and} \qquad \mathbf{Z}^\top\mathbf{Z} = n\mathbf{I}_p. \tag{9}$$

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 (9). 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 [Michailidis and 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.

9. Appendix B: 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

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

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

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

10. Appendix C: VASpca

URL of VASpca
`http://mo161.soci.ous.ac.jp/vaspca/indexE.html`

11. References

Al-Kandari, N.M. and Jolliffe, I.T. (2001). Variable selection and interpretation of covariance principal components. *Communications in Statistics. Simulation and Computation*, 30, 339-354.

Al-Kandari, N.M. and Jolliffe, I.T. (2005). Variable selection and interpretation in correlation principal components. *Environmetrics*, 16, 659-672.

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

Cadima, J., Cerdeira, J.O. and Manuel, M. (2004). Computational aspects of algorithms for variable selection in the context of principal components. *Computational Statistics and Data Analysis*, 47, 225-236.

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

Iizuka, M., Mori, Y., Tarumi, T. and Tanaka, Y. (2003). Computer intensive trials to determine the number of variables in PCA. *Journal of the Japanese Society of Computational Statistics*, 15, 337-345.

Jolliffe, I.T. (1972). Discarding variables in a principal component analysis. I. Artificial data. *Applied Statistics*, 21, 160-173.

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.

Krzanowski, W.J. (1987). Selection of variables to preserve multivariate data structure using principal components. *Applied Statistics*, 36, 22-33.

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

Kuroda, M., Mori, Y., Iizuka, M. and Sakakihara, M. (2011). Accelerating the convergence of the EM algorithm using the vector epsilon algorithm. *Computational Statistics and Data Analysis*, 55, 143-153.

McCabe, G.P. (1984). Principal variables. *Technometrics*, 26, 137-144.

Michailidis, G. and de Leeuw, J. (1998). The Gifi system of descriptive multivariate analysis. *Statistical Science*, 13, 307-336.

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.

Mori, Y., Tarumi, T. and Tanaka, Y. (1998). Principal component analysis based on a subset of variables - Numerical investigation on variable selection procedures -. *Bulletin of the Computational Statistics Society of Japan*, 11, 1-12 (in Japanese).

Mori, Y., Iizuka, M., Tanaka, Y. and Tarumi, T. (2006). Variable Selection in Principal Component Analysis. Härdle, W., Mori, Y. and Vieu, P. (eds), *Statistical Methods for Biostatistics and Related Fields*, 265-283, Springer.

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.

Robert, P. and Escoufier, Y. (1976). A unifying tool for linear multivariate statistical methods: the RV-coefficient. *Applied Statistics*, 25, 257-265.

Sano, K., Manaka, S., Kitamura, K., Kagawa M., Takeuchi, K., Ogashiwa, M., Kameyama, M., Tohgi, H. and Yamada, H. (1977). Statistical studies on evaluation of mild disturbance of consciousness - Abstraction of characteristic clinical pictures by cross-sectional investigation. *Sinkei Kenkyu no Shinpo* 21, 1052-1065 (in Japanese).

Tanaka, Y. and 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., 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.

Wang, M., Kuroda, M., Sakakihara, M. and 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.

IntechOpen

IntechOpen

Principal Component Analysis

Edited by Dr. Parinya Sanguansat

ISBN 978-953-51-0195-6
Hard cover, 300 pages
Publisher InTech
Published online 02, March, 2012
Published in print edition March, 2012

This book is aimed at raising awareness of researchers, scientists and engineers on the benefits of Principal Component Analysis (PCA) in data analysis. In this book, the reader will find the applications of PCA in fields such as image processing, biometric, face recognition and speech processing. It also includes the core concepts and the state-of-the-art methods in data analysis and feature extraction.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Masahiro Kuroda, Yuichi Mori, Masaya Iizuka and Michio Sakakihara (2012). Acceleration of Convergence of the Alternating Least Squares Algorithm for Nonlinear Principal Components Analysis, Principal Component Analysis, Dr. Parinya Sanguansat (Ed.), ISBN: 978-953-51-0195-6, InTech, Available from: http://www.intechopen.com/books/principal-component-analysis/acceleration-of-convergence-of-the-alternating-least-squares-algorithm-for-nonlinear-principal-compo

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2012 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the Creative Commons Attribution 3.0 License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IntechOpen

IntechOpen