Principal components based on a subset of qualitative variables and its accelerated computational algorithm

Kuroda Masahiro
Department of Socio-Information, Okayama University of Science
1-1 Ridaicho, Kita-ku
Okayama 700-0005, Japan
E-mail: kuroda@soci.ous.ac.jp

Iizuka Masaya
Graduate School of Environmental Science, Okayama University
1-3-1 Tsushima-naka, Kita-ku
Okayama 700-8530, Japan
E-mail: iizuka@ems.okayama-u.ac.jp

Mori Yuichi
Department of Socio-Information, Okayama University of Science
1-1 Ridaicho, Kita-ku
Okayama 700-0005, Japan
E-mail: mori@soci.ous.ac.jp

Sakakihara Michio
Department of Information Science, Okayama University of Science
1-1 Ridaicho, Kita-ku
Okayama 700-0005, Japan
E-mail: sakaki@mis.ous.ac.jp

Abstract

Principal components analysis (PCA) is a popular dimension-reducing tool that replaces the variables in a data set by a smaller number of derived variables. In the context of PCA, we sometimes meet variable selection problems. For example, in order to give a simple interpretation of principal components, we select a subset of variables that best approximates all the variables. Modified PCA (M.PCA) proposed by Tanaka and Mori (1997) derives principal components which are computed as a linear combination of a subset of variables but can reproduce all the variables very well. When applying M.PCA to qualitative data, the alternating least squares (ALS) algorithm can be used as a quantification method. In this situation, the computation time has been a big issue so far, because the total number of iterations of the algorithm is much larger and it takes a long computational time until its convergence even though a cost-saving selection procedure such as Backward elimination or Forward selection is employed. In order to accelerate the convergence of the ALS algorithm in PCA of qualitative data, Kuroda et al. (2011) derived a new iterative algorithm using using the vector ε (vε) algorithm by Wynn (1962). In this paper, we investigate how much the proposed vε acceleration algorithm improves the computational efficiency when applying the vε accelerated ALS algorithm to the variable selection problem in M.PCA of qualitative data.

1 Introduction

In the analysis of data with large numbers of variables, a common objective is to reduce the dimensionality of the data set. Principal components analysis (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. For other several situations, we meet variable selection problems in the context of PCA. Various variable selection criteria in PCA has been proposed by Jolliffe (1972, 1973), McCabe (1984), Robert and Escoufier (1976) and Krzanowski (1987). Al-Kandari et al. (2001, 2005) gave guidelines as to the types of data for which each variable selection criteria is useful. 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 (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. (1997) extended M.PCA to qualitative data and provided variable selection procedures, in which the alternating least squares (ASL) algorithm of Young et al. (1978) is utilized. Then the ALS algorithm iterates between optimal scaling for quantifying qualitative data and estimating parameters in M.PCA. In this situation, the computation time of the ALS algorithm has been a big issue so far, because the total number of iterations of the algorithm is much larger for quantification using the ALS algorithm, and it takes a long computational time until its convergence for searching a reasonable subset even though a cost-saving selection procedure such as Backward elimination or Forward selection is employed. For such situations, to accelerate the convergence of the ALS algorithm in PCA of qualitative data, Kuroda et al. (2011) derived a new iterative algorithm using the vector ε (vε) algorithm by Wynn (1962).

In this paper, we investigate how much the proposed vε acceleration algorithm improves the computational efficiency for the variable selection problem in M.PCA of qualitative data. Section 2 introduces the formulation of M.PCA, and Backward elimination and Forward selection procedures used in variable selection. In Section 3, we describe the ALS algorithm for quantifying qualitative data in PCA and, in Section 4, present the vε accelerated ALS algorithm for speeding up the convergence. Numerical experiments in Section 5 illustrate the performance of the vε accelerated ALS algorithm. In Section 6, we present our concluding remarks.

2 Modified PCA for variable selection

2.1 Formulation of modified 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 variables and be columnwise standardized. In PCA, we postulate that $\mathbf{X}$ is approximated by the following bilinear form:

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

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 $\mathbf{A}^\top\mathbf{A} = \mathbf{I}_r$.

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:

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

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.

2.2 Variable selection procedures

In order to find a subset of q variables, we employ Backward elimination and Forward selection of Mori et al. (1998, 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 (2).

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)*
Basically the opposites of Stage B in Backward elimination.

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 ${}_pC_r + (p-r) + (p-(r+1)) + \cdots + 2 = {}_pC_r + (p-r-1)(p-r+2)/2$.

3 Alternating least squares algorithm for M.PCA of qualitative data

When observed data are qualitative, ordinary PCA can not 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

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

where $\mathbf{1}_n$ and $\mathbf{0}_p$ are vectors of ones and zeros of length n and p respectively.

To apply PCA to qualitative data, we determine the optimal scaling parameter $\mathbf{X}^*$, in addition to estimating $\mathbf{Z}$ and $\mathbf{A}$. A possible computational algorithm for estimating simultaneously $\mathbf{Z}$, $\mathbf{A}$ and $\mathbf{X}^*$ is the ALS algorithm. The existing ALS algorithms for PCA of qualitative data are PRINCIPALS of Young et al. (1978) and PRINCALS of Giff (1989). Mori et al. (1996) used PRINCIPALS in applying M.PCA to qualitative data. PRINCIPALS alternates between ordinary PCA and optimal scaling, and minimizes

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

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 vector ε acceleration of the ALS algorithm

We briefly introduce the vε algorithm of Wynn (1962). 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 et al. (2011) provided the vε accelerated ALS algorithm for PCA of mixed measurement level data. The acceleration algorithm speeds up the convergence of the ALS sequence via

the vε algorithm and demonstrated that its speed of convergence is significantly faster than that of the ALS 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 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 Kuroda et al. (2011). 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 most 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 the vε accelerated PRINCIPALS (vε-PRINCIPALS) produces a faster convergent sequence $\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$ of $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$ 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.

5 Numerical experiments

In this section, we study how much faster vε-PRINCIPALS converges than ordinary PRINCIPALS when applying to variable selection in M.PCA of qualitative data. Backward elimination and Forward selection procedures are used in the problem. All computations are performed with the statistical package `R`. CPU times taken are measured by the function `proc.time`[1]. 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 Equation (4).

[1]Times are typically available to 10 msec.

5.1 Simulated data

We apply PRINCIPALS and vε-PRINCIPALS to variable selection in M.PCA of qualitative data using simulated data that consist of 100 observations on 10 variables with 5 levels.

Table 1 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. We compute 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. 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 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 to 5 times smaller and CPU time 2 to 5 times shorter than vε-PRINCIPALS. In particular, vε-PRINCIPALS speeds up well the convergence for the larger numbers of iterations.

The last row in the table 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 64491 iterations in Backward elimination and 178249 iterations in Forward selection, while vε-PRINCIPALS finds the subsets after 17530 and 32405 iterations, respectively. These values show that the computational 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 algorithm works well to accelerate the convergence of $\{\mathbf{X}^{*(t)}\}_{t\geq 0}$ and consequently results in greatly reduced computation times in variable selection problems.

5.2 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. (1977). The data consist of 87 individuals and 23 categorical variables with 4 levels. In the variable selection problem, we select a suitable subset based on 2 $(= r)$ principal components.

Table 2 summarizes the results of variable selection using Backward elimination and Forward selection procedures for finding a subset of q variables. 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. (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. The values of the second to fifth columns in the table show that vε-PRINCIPALS finds a subset of q variables by taking the smaller numbers of iterations and shorter CPU times than PRINCIPALS. We see from the last row in the table that the computational time of vε-PRINCIPALS is less than one-half of that of PRINCIPALS when finding 22 subsets using two selection procedures.

6 Concluding remarks

In this paper, we examine the performance of the vε accelerated ALS algorithm by applying to variable selection problems in M.PCA of qualitative data. The vε algorithm in itself is a fairly simple computational procedure and speeds up the convergence of the ALS algorithm without losing its stable convergence property.

Numerical experiments employing simulated and real data demonstrated that the vε accelerated ALS algorithm improves the speed of convergence of the ordinary ALS algorithm and enables greatly

the reduction of computation times in the variable selection problems for finding a suitable variable set using Backward elimination and Forward selection. The results indicate that the vε acceleration works effectively 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. We will provide VASpca using vε-PRINCIPALS as the iterative algorithm for PCA and M.PCA of qualitative data.

Acknowledgment

This research is supported by the Japan Society for the Promotion of Science (JSPS), Grant-in-Aid for Scientific Research (C), No 20500263 and No 22500265.

Appendix

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

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.

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. (1989). Algorithm descriptions for ANACOR, HOMALS, PRINCALS, and OVERALS. Report RR 89-01. Leiden: Department of Data Theory, University of Leiden.

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.

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

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.

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.

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

Table 1: 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.

(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	1363	17.40	438	6.64	3.11	2.62
8	1620	20.19	400	5.98	4.05	3.37
7	1348	16.81	309	4.80	4.36	3.50
6	4542	53.72	869	11.26	5.23	4.77
5	13735	159.72	2949	35.70	4.66	4.47
4	41759	482.59	12521	148.13	3.34	3.26
3	124	1.98	44	1.06	2.82	1.86
Total	64491	752.40	17530	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	4382	67.11	1442	33.54	3.04	2.00
4	154743	1786.70	26091	308.33	5.93	5.79
5	13123	152.72	3198	38.61	4.10	3.96
6	3989	47.02	1143	14.24	3.49	3.30
7	1264	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	178249	2078.33	32405	402.40	5.50	5.16

Table 2: 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.

(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	2793	86.55	1354	43.48	2.06	1.99	0.620
2	35	1.92	10	1.34	3.50	1.43	0.581
Total	13581	498.82	4273	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	3442	176.76	1026	119.07	3.35	1.48	0.597
3	5389	170.82	1189	44.28	4.53	3.86	0.633
4	1804	60.96	429	20.27	4.21	3.01	0.650
5	1406	48.53	349	17.41	4.03	2.79	0.662
6	1243	43.25	305	15.75	4.08	2.75	0.668
7	1114	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	20656	771.88	5176	328.81	3.99	2.35	