Variable selection in principal components analysis of qualitative data using the accelerated ALS algorithm

Masahiro Kuroda (Okayama University of Science)
Yuichi Mori (Okayama University of Science)
Masaya Iizuka (Okayama University)
Michio Sakakihara (Okayama University of Science)

* supported by the Japan Society for the Promotion of Science (JSPS), Grant-in-Aid for Scientific Research (C), No 20500263.

Motivation

Variable section in PCA of qualitative data

The ALS algorithm for quantifying qualitative data
PCA.ALS
- **PRINCIPALS**: Young, Takane & de Leeuw (1978) (SAS)
- **PRINCALS**: Gifi (1990) (SPSS)

Acceleration of PCA.ALS using the vector ε (vε) algorithm
- **vε-PCA.ALS**: Kuroda, Mori, Iizuka & Sakakihara (2011) in *CSDA*.

Modified PCA (M.PCA)
- Formulation of M.PCA: Tanaka & Mori (1997)
- Backward elimination & Forward selection procedures: Mori et al. (1998, 2006)

Application of vε-PCA.ALS to variable selection in M.PCA of qualitative data

We study how the ve acceleration reduces the numbers of iterations and CPU times of PCA.ALS for variable selection in M.PCA of qualitative data

Model parameters of PCA of quantitative data

$\mathbf{X}$: $n \times p$ matrix (n observations on p variables; columnwise standardized)

In PCA, $\mathbf{X}$ is postulated to be approximated by a bilinear structure of the form:

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

where

$\mathbf{Z}$ is an $n \times r$ matrix of n component scores on r components ($1 \leq r \leq p$),
$\mathbf{A}$ is a $p \times r$ matrix consisting of the eigenvectors of $\mathbf{X}^\top\mathbf{X}/n$ and $\mathbf{A}^\top\mathbf{A} = \mathbf{I}_r$.

We find $\mathbf{Z}$ and $\mathbf{A}$ such that

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

is minimized for the prescribed number of components r.

PCA with variables measured by mixed scaled levels

For only quantitative variables (interval and ratio scales)

We can find $\mathbf{Z}$ and $\mathbf{A}$ (or $\hat{\mathbf{X}} = \mathbf{Z}\mathbf{A}^\top$) minimizing

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

For mixed scaled variables (nominal, ordinal, interval and ratio scales)

Optimal scaling is necessary to quantify the observed qualitative data, i.e., we need to find an optimally scaled observation $\mathbf{X}^*$ minimizing

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

where

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

in addition to $\mathbf{Z}$ and $\mathbf{A}$, simultaneously.

Alternating least squares algorithm to find the optimal scaled observation $\mathbf{X}^*$

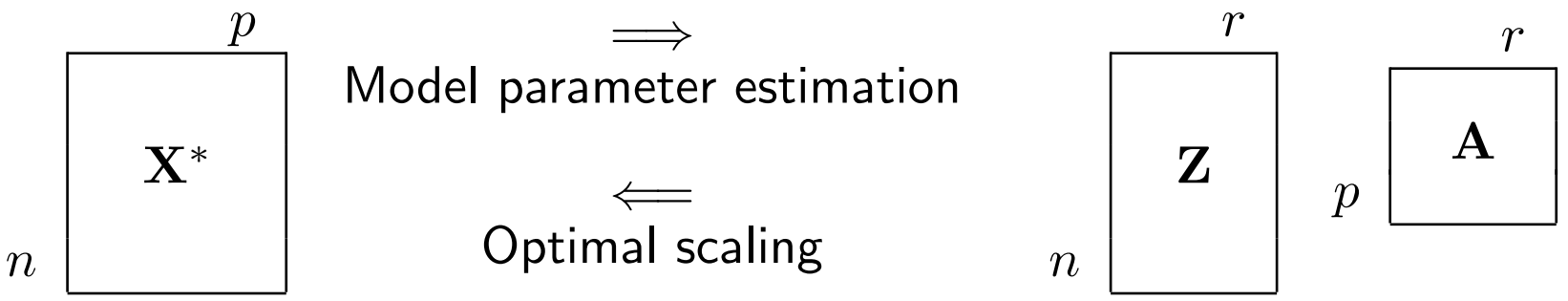
To find **model parameters** $(\mathbf{Z}, \mathbf{A})$ and **optimal scaling parameter** $\mathbf{X}^*$,

the Alternating Least Squares (ALS) algorithm

can be utilized: PCA.ALS (PRINCIPALS, PRINCALS)

Model parameter estimation step :
estimate $\mathbf{Z}$ and $\mathbf{A}$ conditionally on fixed $\mathbf{X}^*$.

Optimal scaling step :
find $\mathbf{X}^*$ for minimizing θ^* conditionally on fixed $\mathbf{Z}$ and $\mathbf{A}$.



Alternating least squares algorithm to find the optimal scaled observation $\mathbf{X}^*$

[PCA.ALS algorithm] PRINCIPALS (Young et al, 1978)

Superscript (t) indicates the t-th iteration.

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

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

  where $\mathbf{A}^\top\mathbf{A} = \mathbf{I}_r$ and $\mathbf{D}_r$ is an $r \times r$ diagonal eigenvalue matrix.
  Compute $\mathbf{Z}^{(t)}$ from $\mathbf{Z}^{(t)} = \mathbf{X}^{*(t)}\mathbf{A}^{(t)}$.

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

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

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

Acceleration of PCA.ALS by the vector ε accelerator

To accelerate the computation, we can use

vector ε accelerator (vε accelerator)

by Wynn (1962), which

speeds up the convergence of a slowly convergent vector sequence,

is very effective for linearly converging sequences,

generates a sequence $\{\dot{\mathbf{Y}}^{(t)}\}_{t\geq 0}$ from the iterative sequence $\{\mathbf{Y}^{(t)}\}_{t\geq 0}$.

- **Convergence**: The accelerated sequence $\{\dot{\mathbf{Y}}^{(t)}\}_{t\geq 0}$ converges to the stationary point $\mathbf{Y}^{\infty}$ of $\{\mathbf{Y}^{(t)}\}_{t\geq 0}$ faster than $\{\mathbf{Y}^{(t)}\}_{t\geq 0}$.
- **Computational cost**: At each iteration, the vε algorithm requires only $O(d)$ arithmetic operations while the Newton-Raphson and quasi-Newton algorithms are achieved at $O(d^3)$ and $O(d^2)$ where d is the dimension of $\mathbf{Y}$.
- **Convergence speed**: The best speed of convergence is superlinear.

Acceleration of PCA.ALS by the vector ε accelerator

The vε accelerator is given by

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

where $[\mathbf{Y}]^{-1} = \mathbf{Y}/||\mathbf{Y}||^2$ and $||\mathbf{Y}||$ is the Euclidean norm of $\mathbf{Y}$.

- The accelerated vector $\dot{\mathbf{Y}}^{(t-1)}$ is obtained by the original sequence $(\mathbf{Y}^{(t-1)}, \mathbf{Y}^{(t)}, \mathbf{Y}^{(t+1)})$
- The vε accelerator does not depend on the statistical model $\{\mathbf{Y}^{(t)}\}_{t\geq 0}$. Therefore, when the v$\varepsilon$ algorithm is applied to ALS, it guarantees the convergence properties of the ALS.

Acceleration of PCA.ALS by the vector ε accelerator

To accelerate PCA.ALS, we introduce the vε algorithm into PCA.ALS, i.e.,

From a sequence $\{\mathbf{X}^{*(t)}\}_{t\geq 0} = \{\mathbf{X}^{*(0)}, \mathbf{X}^{*(1)}, \cdots, \mathbf{X}^{*(\infty)}\}$ in PCA.ALS, make an accelerated sequence $\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0} = \{\dot{\mathbf{X}}^{*(0)}, \dot{\mathbf{X}}^{*(1)}, \cdots, \mathbf{X}^{*(\infty)}\}$.

[General procedure of vε-PCA.ALS]

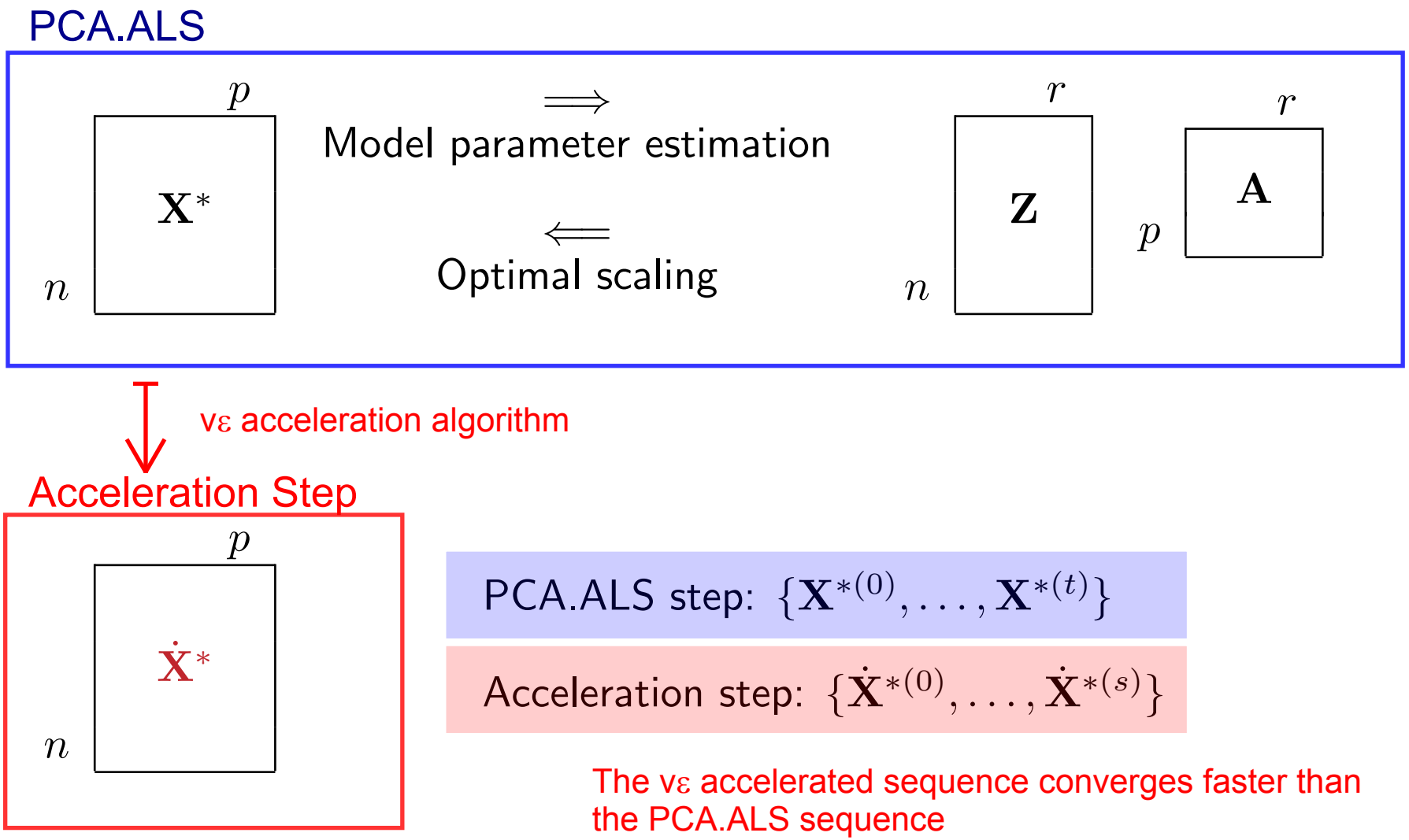
Alternate the following two steps until the algorithm is converged:

- *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:

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

where vec$\mathbf{X}^*$ stands for the vectors of columns of $\mathbf{X}^*$, and check the convergence by $\|\mathrm{vec}(\dot{\mathbf{X}}^{*(t-1)} - \dot{\mathbf{X}}^{*(t-2)})\|^2 < \delta$, where δ is a desired accuracy.

Acceleration of PCA.ALS by the vector ε accelerator



PCA.ALS step: $\{\mathbf{X}^{*(0)}, \ldots, \mathbf{X}^{*(t)}\}$

Acceleration step: $\{\dot{\mathbf{X}}^{*(0)}, \ldots, \dot{\mathbf{X}}^{*(s)}\}$

The vε accelerated sequence converges faster than the PCA.ALS sequence

Acceleration of PCA.ALS by the vector ε accelerator

Since vε-PCA.ALS is designed to generate $\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0}$ converging to $\mathbf{X}^{*(\infty)}$,

- 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.

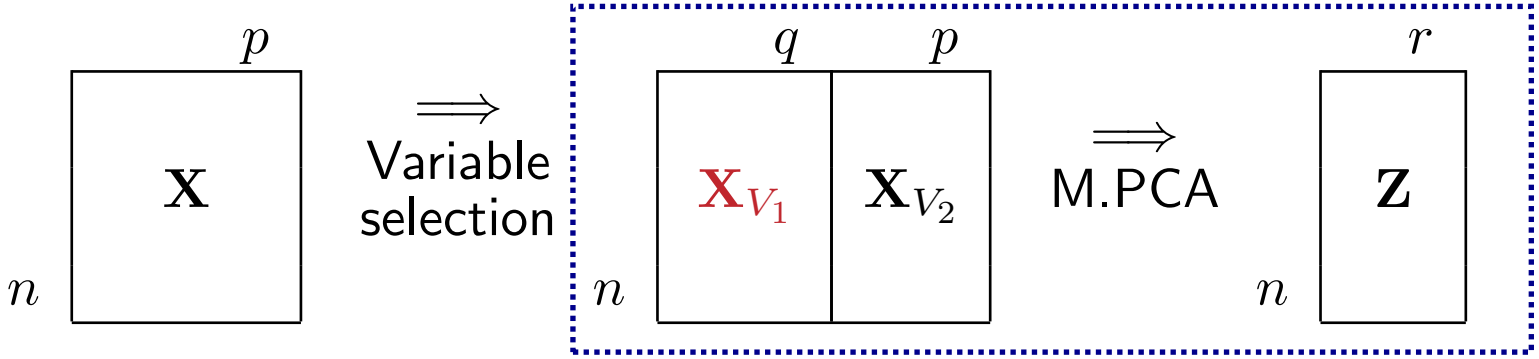
Note that

- $\dot{\mathbf{X}}^{*(t-1)}$ 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 PCA.ALS procedure.

Modified PCA (M.PCA) of Tanaka & Mori (1997)

Modified PCA (M.PCA) derives principal components that
- are computed as linear combinations of a subset of variables
- but 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 $\mathbf{Z} = \mathbf{X}_{V_1}\mathbf{A}$.



Formulation of Modified PCA (M.PCA)

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:

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

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 $P = \sum_{j=1}^{r} \lambda_j / \mathrm{tr}(\mathbf{S})$ $\Longleftarrow$ we use P as variable selection criteria.
- the RV-coefficient $RV = \left\{ \sum_{j=1}^{r} \lambda_j^2 / \mathrm{tr}(\mathbf{S}^2) \right\}^{1/2}$.

Variable selection procedures in M.PCA

In order to find a subset of q variables,
we employ two variable selection procedures of Mori et al. (1998, 2006)

- Backward elimination procedure and
- Forward selection procedure.

These procedures

- are cost-saving stepwise selection procedures and
- are remove or add only one variable sequentially.

Backward elimination

[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.
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

[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 $q := q+1$.

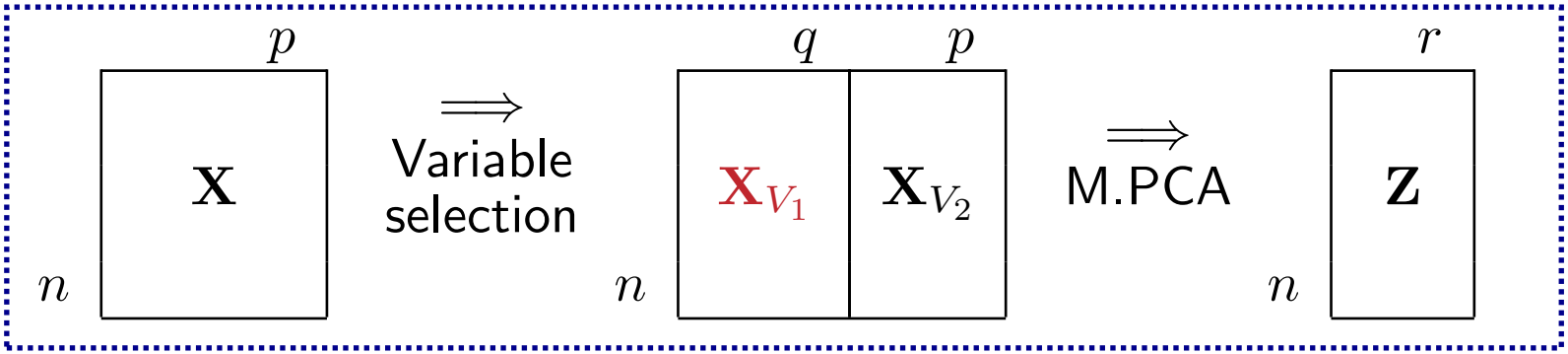
B-2 If the P or q are smaller (or larger) than preassigned values, go back to **B-1**. Otherwise stop.

Variable selection in M.PCA

Variable selection is to
find a subset of q variables that best approximates all the variables.

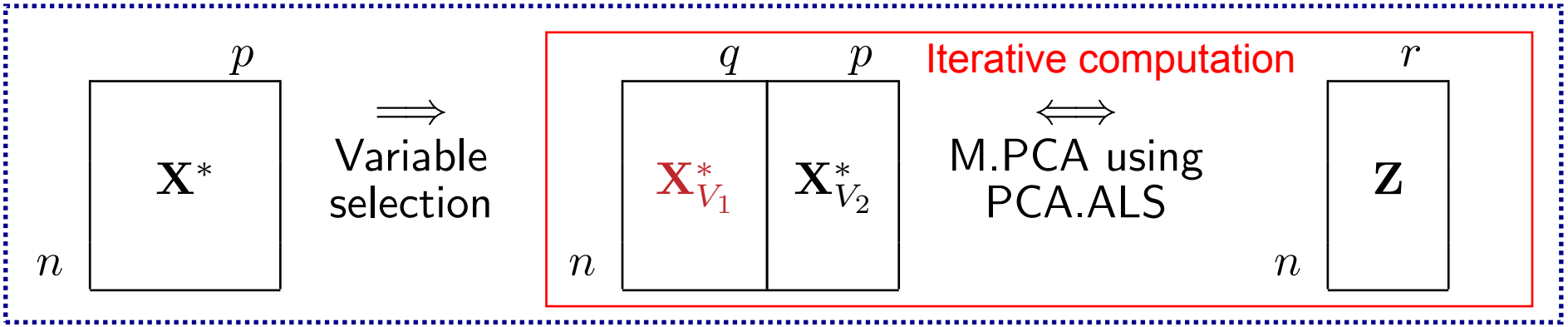
$\Downarrow$

Find $\mathbf{X}_{V_1}$ using Backward elimination or Forward selection procedures.



Variable selection in M.PCA of qualitative data

M.PCA of qualitative data: Iteration between Variable selection and PCA.ALS

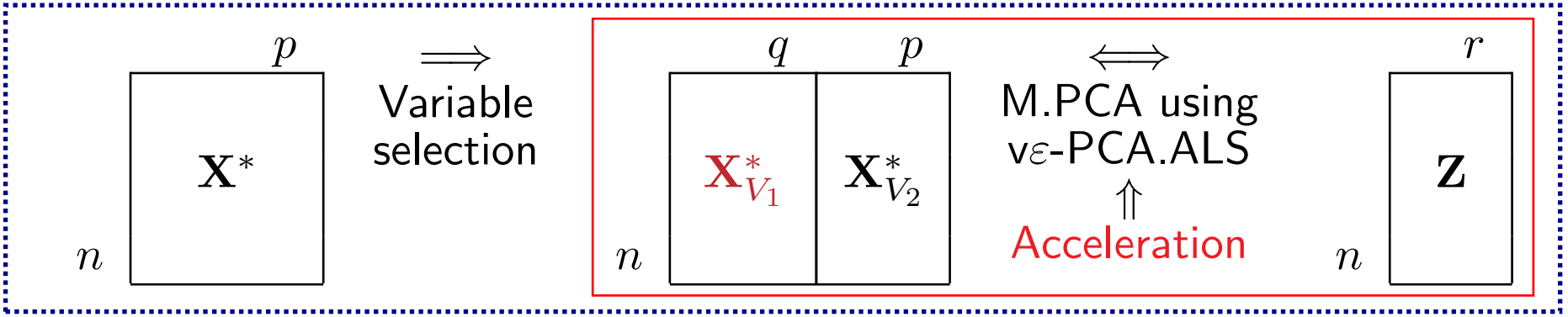


Steps of variable selection and PCA.ALS

- Variable selection:
  Select $\mathbf{X}_{V_1}$ using Backward elimination or Forward selection.

- PCA.ALS:
  Quantify $\mathbf{X}^*_{V_1}$ and compute $\mathbf{A}$ and $\mathbf{Z}$ using the ALS algorithm.

vε-PCA.ALS for variable selection in M.PCA of qualitative data



Steps of variable selection and PCA.ALS

- Variable selection:
  Select $\mathbf{X}_{V_1}$ using Backward elimination or Forward selection.

- vε-PCA.ALS:
  - PCA.ALS:
    Quantify $\mathbf{X}^*_{V_1}$ and compute $\mathbf{A}$ and $\mathbf{Z}$ using the ALS algorithm
  - vε acceleration:
    Generate an accelerated sequence of $\mathbf{X}^*_{V_1}$ using the vε algorithm

Numerical experiments: Variable selection in M.PCA of qualitative data

Computation

- Algorithm: PRINCIPALS and vε-PRINCIPALS
- Convergence criteria: $\delta = 10^{-8}$
- The number of principal components (PCs): $r = 3$

Compare

- The number of iterations and CPU time
- Iteration and CPU time speed-ups

$$\text{Iteration (CPU time) speed-ups} = \frac{\text{The number of iterations (CPU time) of PRINCIPALS}}{\text{The number of iterations (CPU time) of vε-PRINCIPALS}}$$

[Simulation 1]: Computation of PCs

- The size of sample (n) : 100

** The number of items (p) : 40 items with 20 levels (Data 1)

** The number of items (p) : 20 items with 10 levels (Data 2)

++ Replication: 50 times

[Simulation 2]: Variable selection in M.PCA

- The size of sample (n): 100
- The number of items (p): 10 items with 5 levels (Data 3)

Numerical experiments: Data 1 & 2

Table 1: Summary of statistics and speed-ups

(a) Data 1 (100×40 with 20 levels)

	PRINCIPALS		vε-PRINCIPALS		Speed-ups	
	Iteration	CPU time	Iteration	CPU time	Iteration	CPU time
Min.	208.0	10.38	75.0	4.140	1.340	1.310
1st Qu.	435.8	21.27	160.0	8.268	2.200	2.138
Median	592.5	28.72	215.0	10.930	2.675	2.530
Mean	716.2	34.64	277.8	13.991	2.762	2.623
3rd Qu.	788.2	38.20	318.2	15.900	3.180	2.960
Max.	2682.0	128.54	1107.0	54.260	5.430	5.170

Up!!

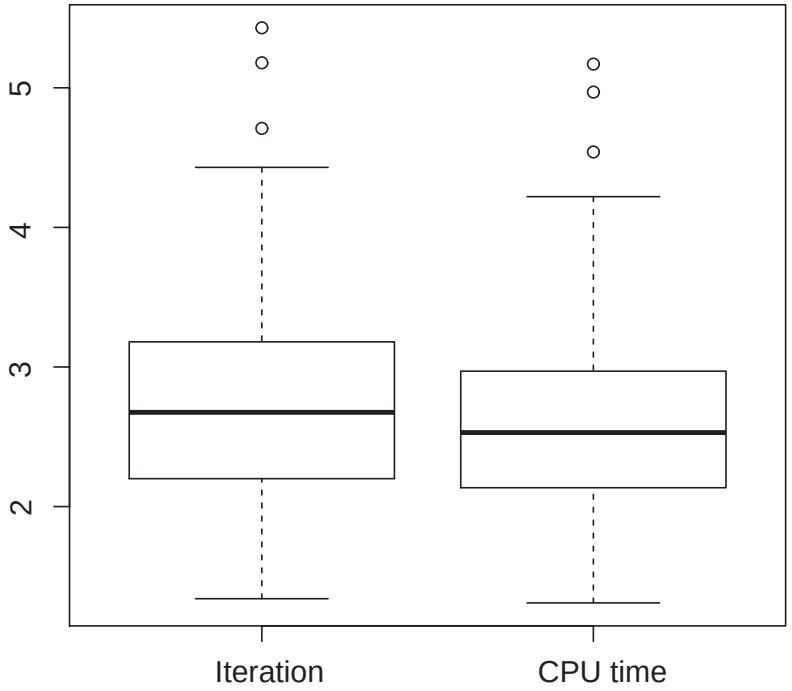
(b) Data 2 (100×20 with 10 levels)

	PRINCIPALS		vε-PRINCIPALS		Speed-ups	
	Iteration	CPU time	Iteration	CPU time	Iteration	CPU time
Min	136.0	2.640	46.0	1.070	1.760	1.690
1st Qu.	236.5	4.435	85.0	1.808	2.487	2.272
Median	345.5	6.370	137.0	2.715	3.280	2.760
Mean	437.0	8.021	135.0	2.702	3.232	2.917
3rd Qu.	573.2	10.390	171.2	3.397	3.740	3.410
Max.	1564.0	28.050	348.0	6.560	5.710	5.240

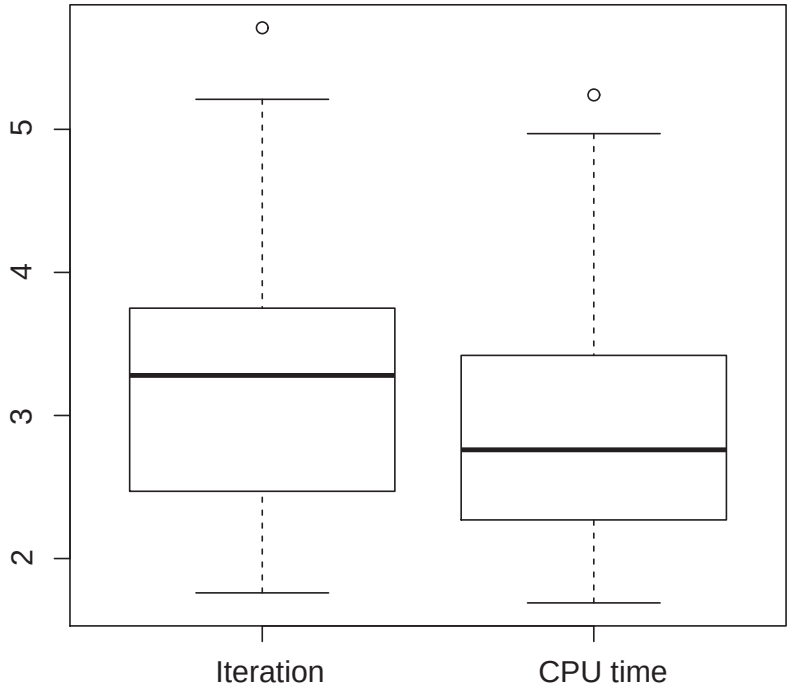
Up!!

Numerical experiments: Boxplots of iteration and CPU time speed-ups

Data 1 (100×40 with 20 levels)



Data 2 (100×20 with 10 levels)



Numerical experiments: Data 3

Table 2(a): 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	Comb.	Iteration	CPU time	Iteration	CPU time	Iteration	CPU time
10	1	141	1.70	48	0.68	2.94	2.49
9	10	1363	17.40	438	6.64	3.11	2.62
8	9	1620	20.19	400	5.98	4.05	3.37
7	8	1348	16.81	309	4.80	4.36	3.50
6	7	4542	53.72	869	11.26	5.23	4.77
5	6	13735	159.72	2949	35.70	4.66	4.47
4	5	41759	482.59	12521	148.13	3.34	3.26
3	4	124	1.98	44	1.06	2.82	1.86
Total	50	64491	752.40	17530	213.57	3.68	3.52

Numerical experiments: Data 3

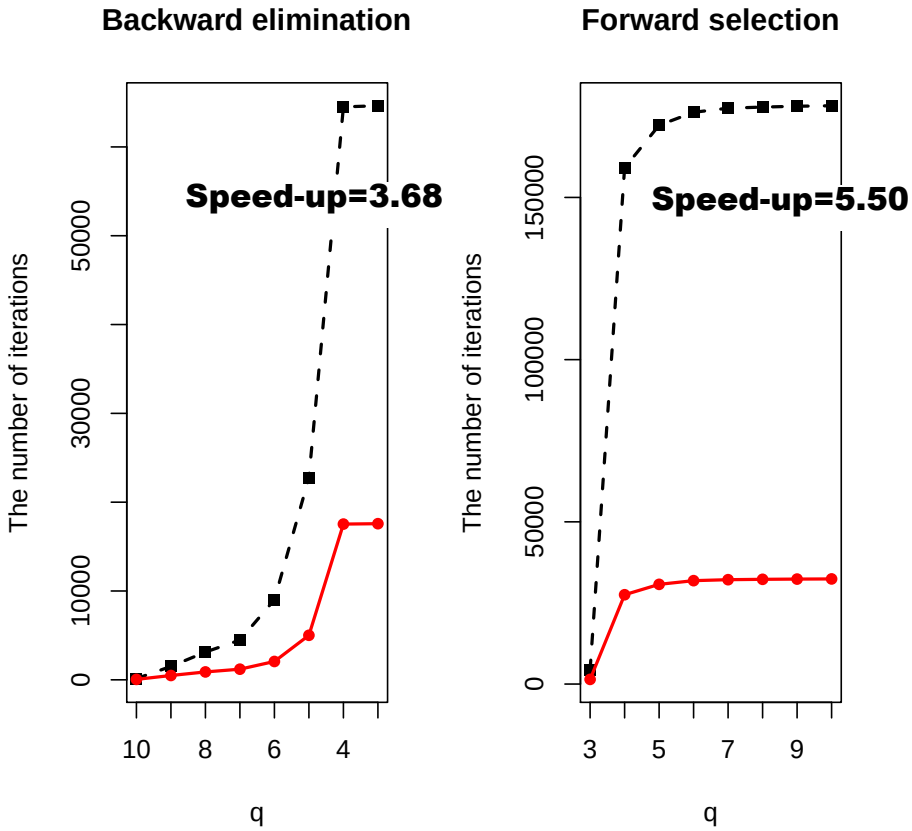
Table 2(b): The numbers of iterations and CPU times of PRINCIPALS and vε-PRINCIPALS and their speed-ups in application to variable selection for finding a subset of q variables using simulated data.

(b) Forward selection

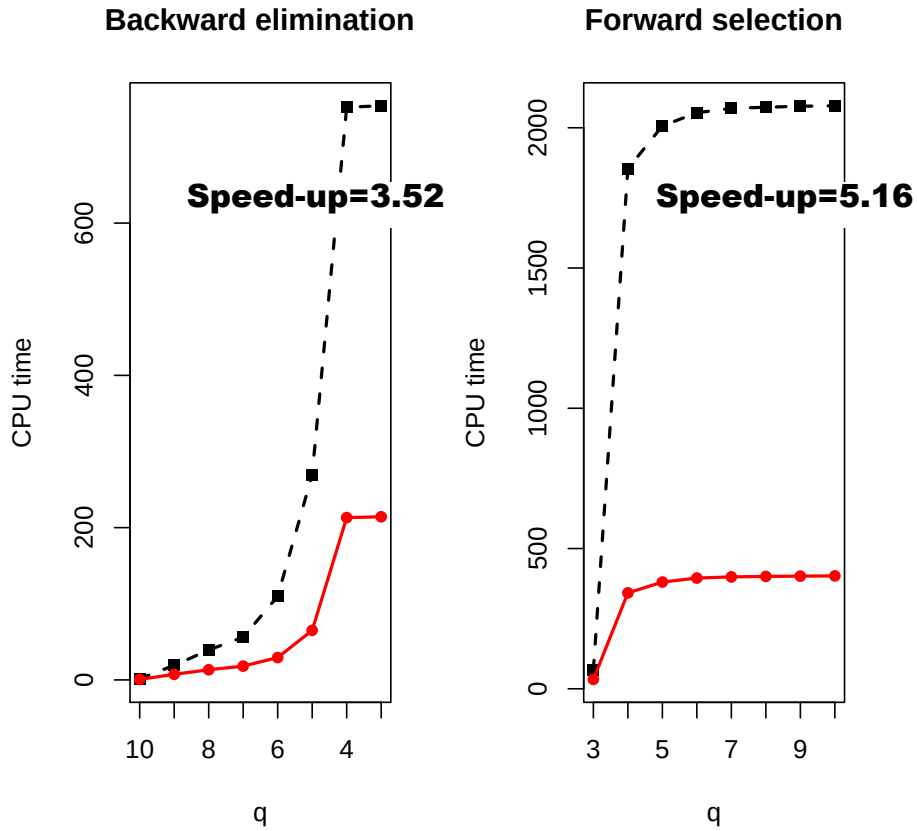
		PRINCIPALS		vε-PRINCIPALS		Speed-up	
q	Comb.	Iteration	CPU time	Iteration	CPU time	Iteration	CPU time
3	120	4382	67.11	1442	33.54	3.04	2.00
4	7	154743	1786.70	26091	308.33	5.93	5.79
5	6	13123	152.72	3198	38.61	4.10	3.96
6	5	3989	47.02	1143	14.24	3.49	3.30
7	4	1264	15.27	300	4.14	4.21	3.69
8	3	340	4.38	108	1.70	3.15	2.58
9	2	267	3.42	75	1.17	3.56	2.93
10	1	141	1.73	48	0.68	2.94	2.54
Total	148	178249	2078.33	32405	402.40	5.50	5.16

Numerical experiments: Data 3

The number of iterations



CPU time



Black line=PRINCIPALS **Red line=vε-PRINCIPALS**

Conclusion

[Numerical experiments]

The **vε acceleration works well** to reduce the computational time of PRINCIPALS.

Means of iteration & CPU time speed-ups: Computation of PCs

	Iteration	CPU time
Data 1 (100×40 with 20 levels)	2.762 (36%)	2.623 (38%)
Data 2 (100×20 with 10 levels)	3.232 (31%)	2.917 (34%)

Iteration & CPU time speed-ups: Variable selection

Data 3	Iteration	CPU time
Backward elimination	3.68 (27%)	3.52 (28%)
Forward selection	5.50 (18%)	5.16 (19%)

[Future works]

We apply vε-PRINCIPALS with a re-start procedure to variable selection in PCA of qualitative data.

References

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

KURODA, M. and SAKAKIHARA, M. (2006): Accelerating the convergence of the EM algorithm using the vector ε algorithm. *Computational Statistics and Data Analysis 51, 1549-1561*.

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

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. In: C. Hayashi, K. Yajima, H. Bock, N. Ohsumi, Y. Tanaka, Y. Baba (Eds.): *Data Science, Classification, and Related Methods (Proceedings of IFCS-96)*. Springer-Verlag, 547-554.

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*.