# Improvement of acceleration of the ALS algorithm using the vector $\varepsilon$ 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.

## Contents

- Alternating least squares algorithm for PCA with variables measured by mixed scaled levels: PCA.ALS
  - **PRINCIPALS**: Young, Takane & de Leeuw (1978) in *Psychometrika* (SAS)
  - **PRINCALS**: Gifi (1990) in nonlinear multivariate analysis (SPSS)
- Acceleration of PCA.ALS by the vector $\varepsilon$ (v$\varepsilon$) algorithm: v$\varepsilon$-PCA.ALS
  $\Longrightarrow$ Kuroda, Mori, Iizuka & Sakakihara (2010) in *CSDA*.
- Improvement of the v$\varepsilon$ accelerated PCA.ALS: r-v$\varepsilon$-PCA.ALS $\Longleftarrow$ Main topic
  - Re-starting strategy for reducing both the number of iterations and the computational time
- Numerical experiments

# Contents

- Alternating least squares algorithm for PCA with variables measured by mixed scaled levels: PCA.ALS
  - **PRINCIPALS**: Young, Takane & de Leeuw (1978) in *Psychometrika* (SAS)
  - **PRINCALS**: Gifi (1990) in nonlinear multivariate analysis (SPSS)
- Acceleration of PCA.ALS by the vector $\varepsilon$ (v$\varepsilon$) algorithm: v$\varepsilon$-PCA.ALS
  $\Longrightarrow$ Kuroda, Mori, Iizuka & Sakakihara (2010) in *CSDA*.
- Improvement of the v$\varepsilon$ accelerated PCA.ALS: r-v$\varepsilon$-PCA.ALS $\Longleftarrow$ Main topic
  - Re-starting strategy for reducing both the number of iterations and the computational time
- Numerical experiments

Related works: Acceleration of the EM algorithm using the v$\varepsilon$ algorithm

- Kuroda & Sakakihara (2006) in *CSDA* propose the $\varepsilon$-accelerated EM algorithm
- Wang, Kuroda, Sakakihata & Geng (2008) in *Comput. Stat.* prove its convergence properties

# PCA with variables measured by mixed scaled levels

$\mathbf{X}$ : $n \times p$ 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 qualitative variables (interval and ratio scales)

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

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

## 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** $\mathbf{Z}$ and $\mathbf{A}$ and **optimal scaling parameter** $\mathbf{X}^*$,

Alternative Least Squares (ALS) algorithms

can be utilized: PCA.ALS

PCA.ALS algorithm is to determine $\theta^*$ by

– updating each of the parameters in turn,

– keeping the others fixed.

i.e., to alternate the following two steps until the algorithm is converged:

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

*Optimal scaling step* :
finding $\mathbf{X}^*$ for minimizing $\theta^*$ 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

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

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

# Acceleration of PCA.ALS by the vector $\varepsilon$ accelerator

To accelerate the computation, we can use

**vector $\varepsilon$ accelerator** (v$\varepsilon$ 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 v$\varepsilon$ 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 $\varepsilon$ accelerator

The v$\varepsilon$ accelerator is given by

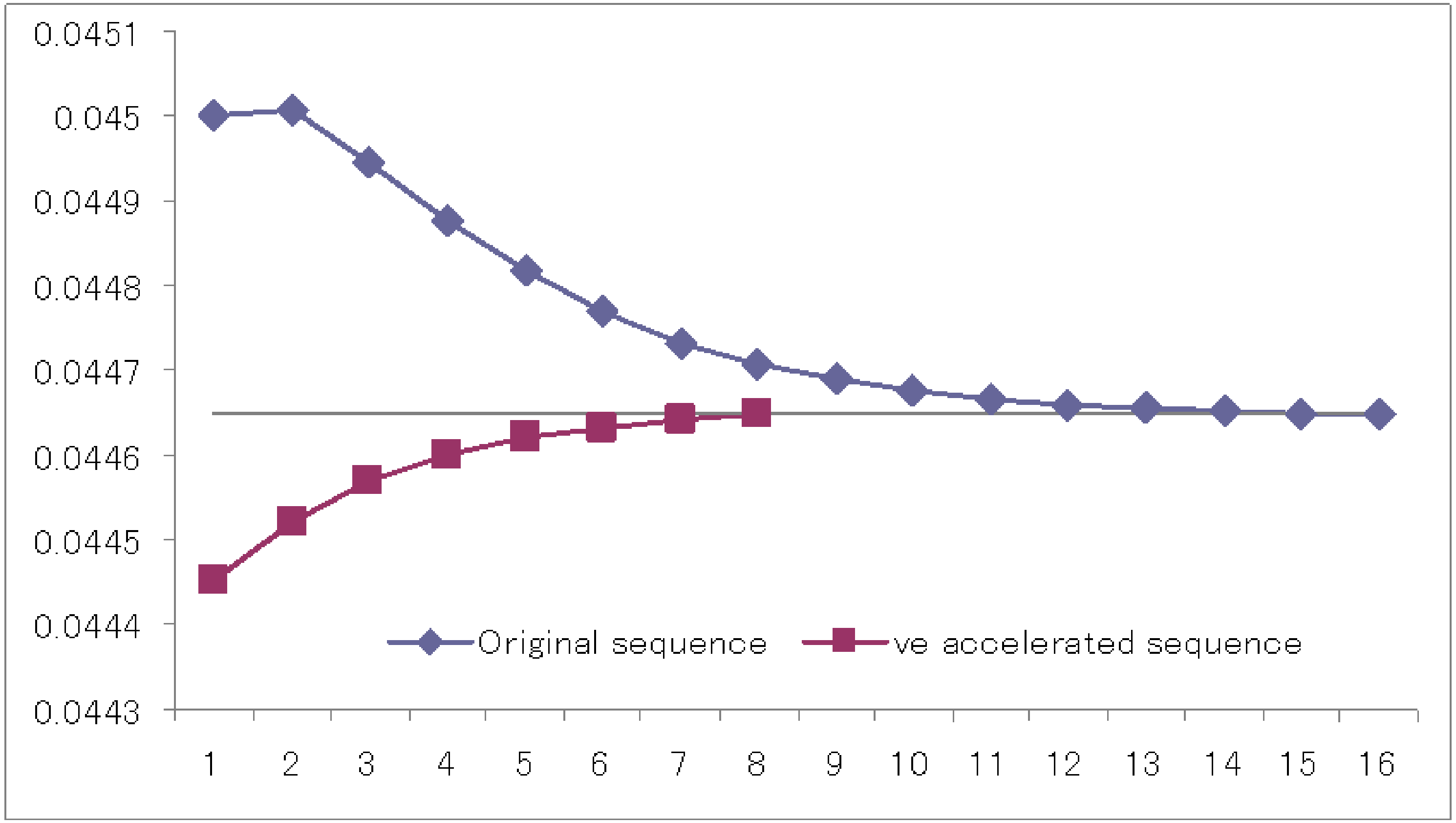
$$\dot{\mathbf{Y}}^{(t-1)} = \theta^{(t)} + \left[\left[\mathbf{Y}^{(t-1)} - \mathbf{Y}^{(t)}\right]^{-1} + \left[\mathbf{Y}^{(t+1)} - \mathbf{Y}^{(t)}\right]^{-1}\right]^{-1},$$

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

$$\begin{array}{ll} \{\mathbf{Y}^{(t)}\} : \mathbf{Y}^{(0)} \rightarrow \mathbf{Y}^{(1)} \rightarrow \mathbf{Y}^{(2)} \rightarrow \mathbf{Y}^{(3)} \rightarrow \cdots \rightarrow \mathbf{Y}^{(S)} \rightarrow \cdots \rightarrow \mathbf{Y}^{(T)} & = \mathbf{Y}^{\infty} \\ \{\dot{\mathbf{Y}}^{(t)}\} : \dot{\mathbf{Y}}^{(0)} \rightarrow \dot{\mathbf{Y}}^{(1)} \rightarrow \dot{\mathbf{Y}}^{(2)} \rightarrow \dot{\mathbf{Y}}^{(3)} \rightarrow \cdots \rightarrow \dot{\mathbf{Y}}^{(S)} & = \mathbf{Y}^{\infty} \end{array}$$

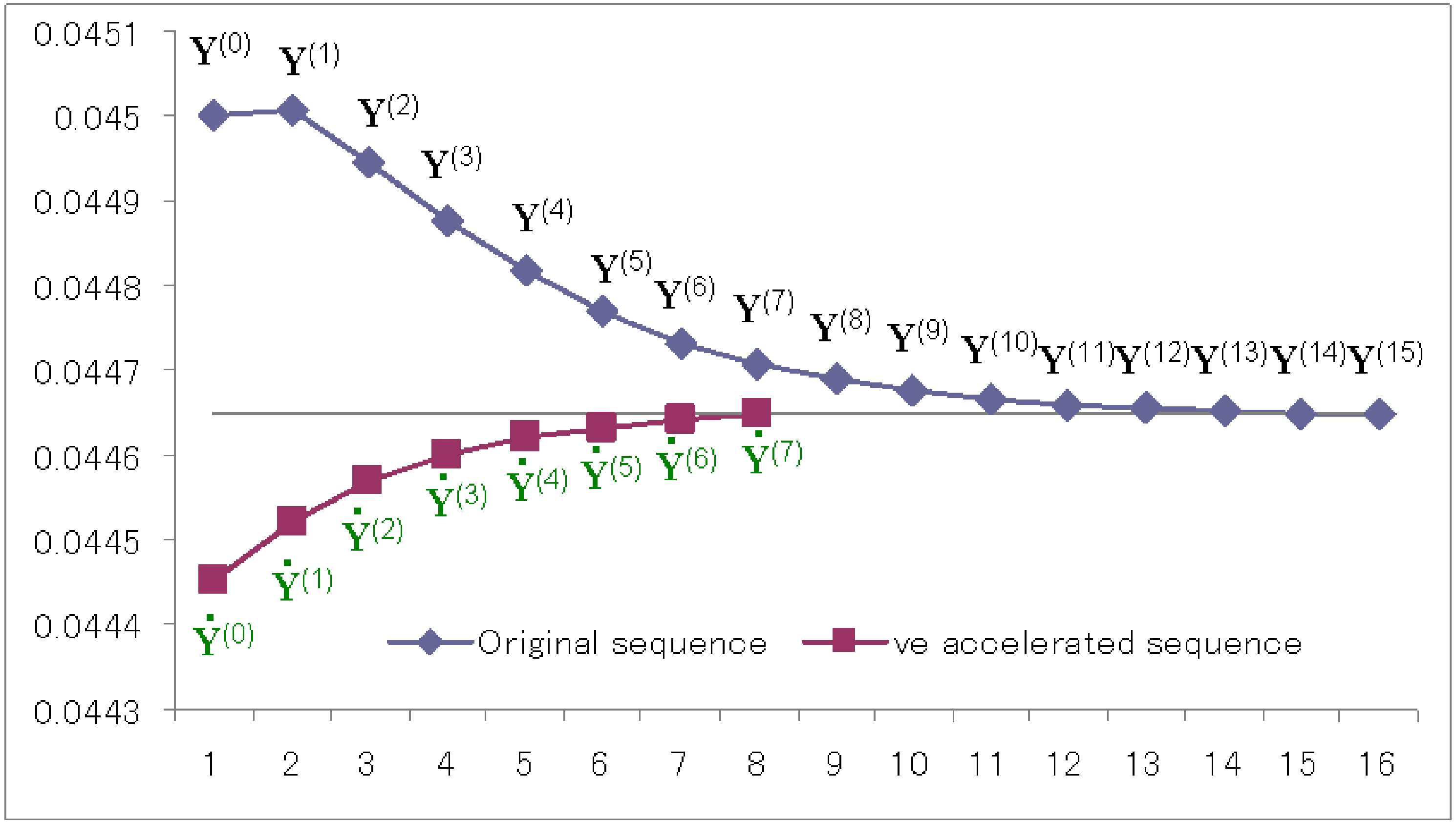
- $S \leq T$
- The accelerated sequence, $\dot{\mathbf{Y}}^{(t-1)}$ is obtained by the original sequence $(\mathbf{Y}^{(t-1)}, \mathbf{Y}^{(t)}, \mathbf{Y}^{(t+1)})$
- The v$\varepsilon$ accelerator does not depend on the statistical model $\{\mathbf{Y}^{(t)}\}_{t\geq 0}$. Therefore, when the v$\varepsilon$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

# Acceleration of PCA.ALS by the vector $\varepsilon$ accelerator



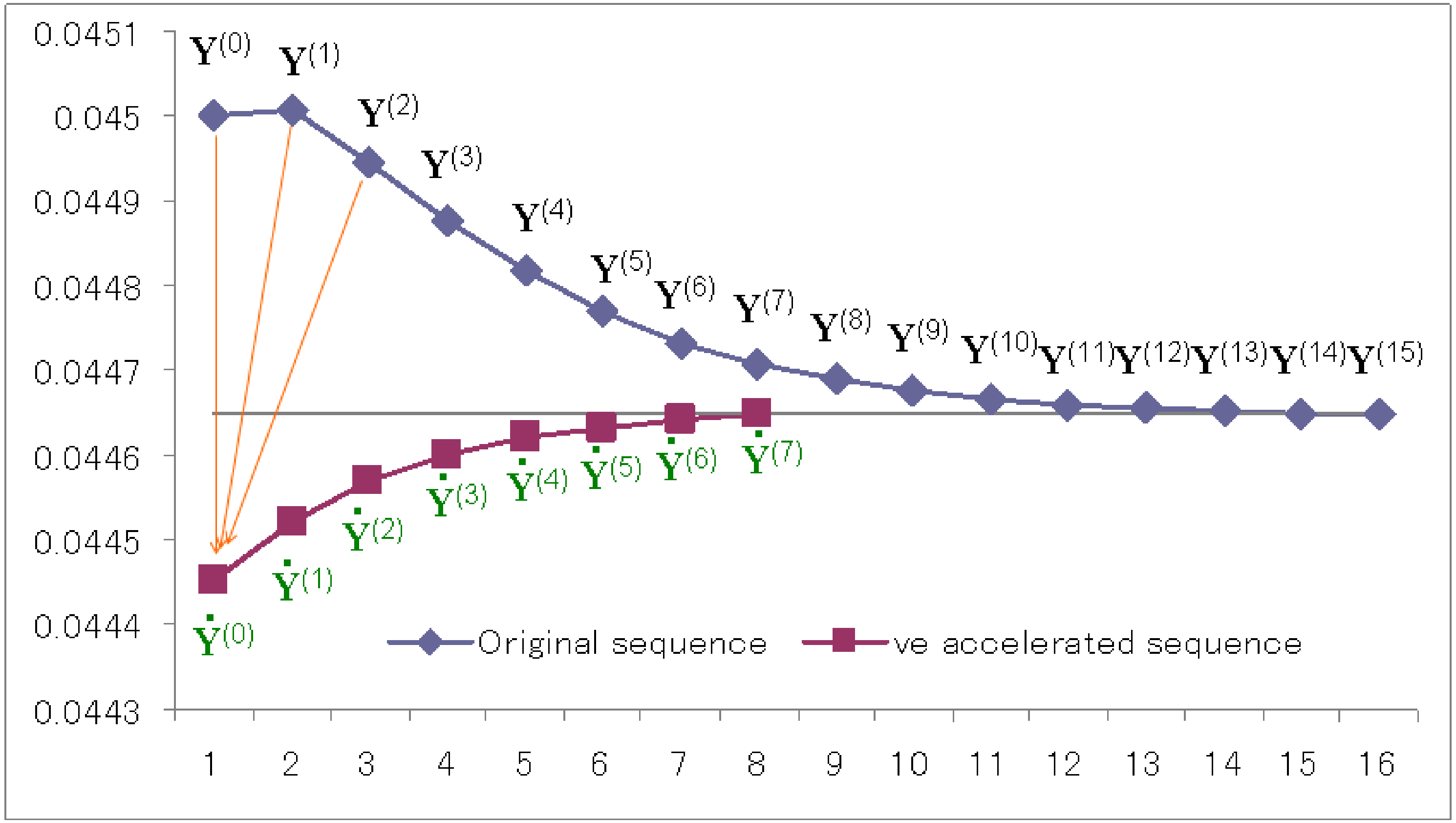
Acceleration by the vector $\varepsilon$ algorithm ($\#$ of iterations 1)

# Acceleration of PCA.ALS by the vector $\varepsilon$ accelerator



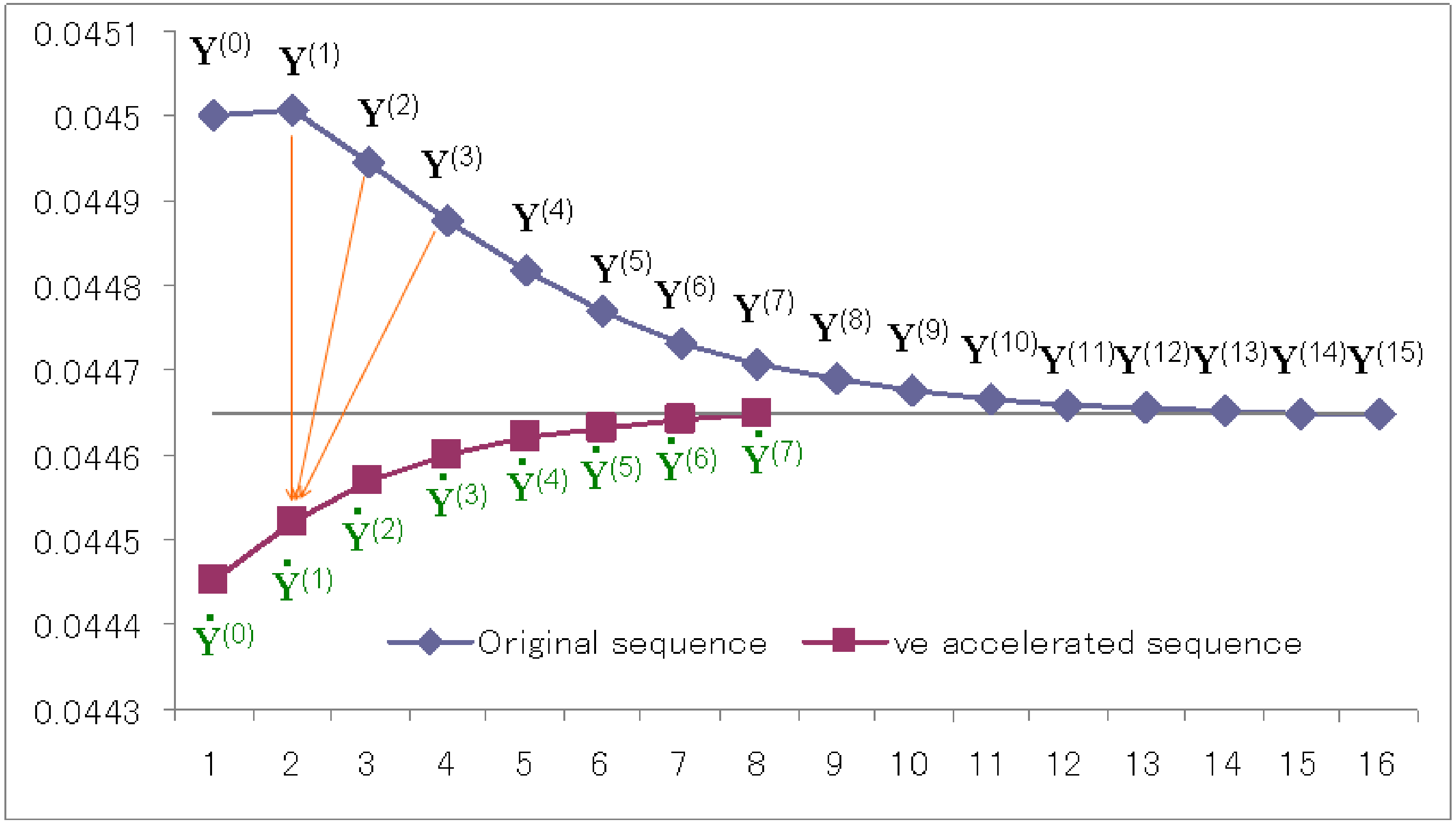
Acceleration by the vector $\varepsilon$ algorithm (# of iterations 2)

## Acceleration of PCA.ALS by the vector $\varepsilon$ accelerator



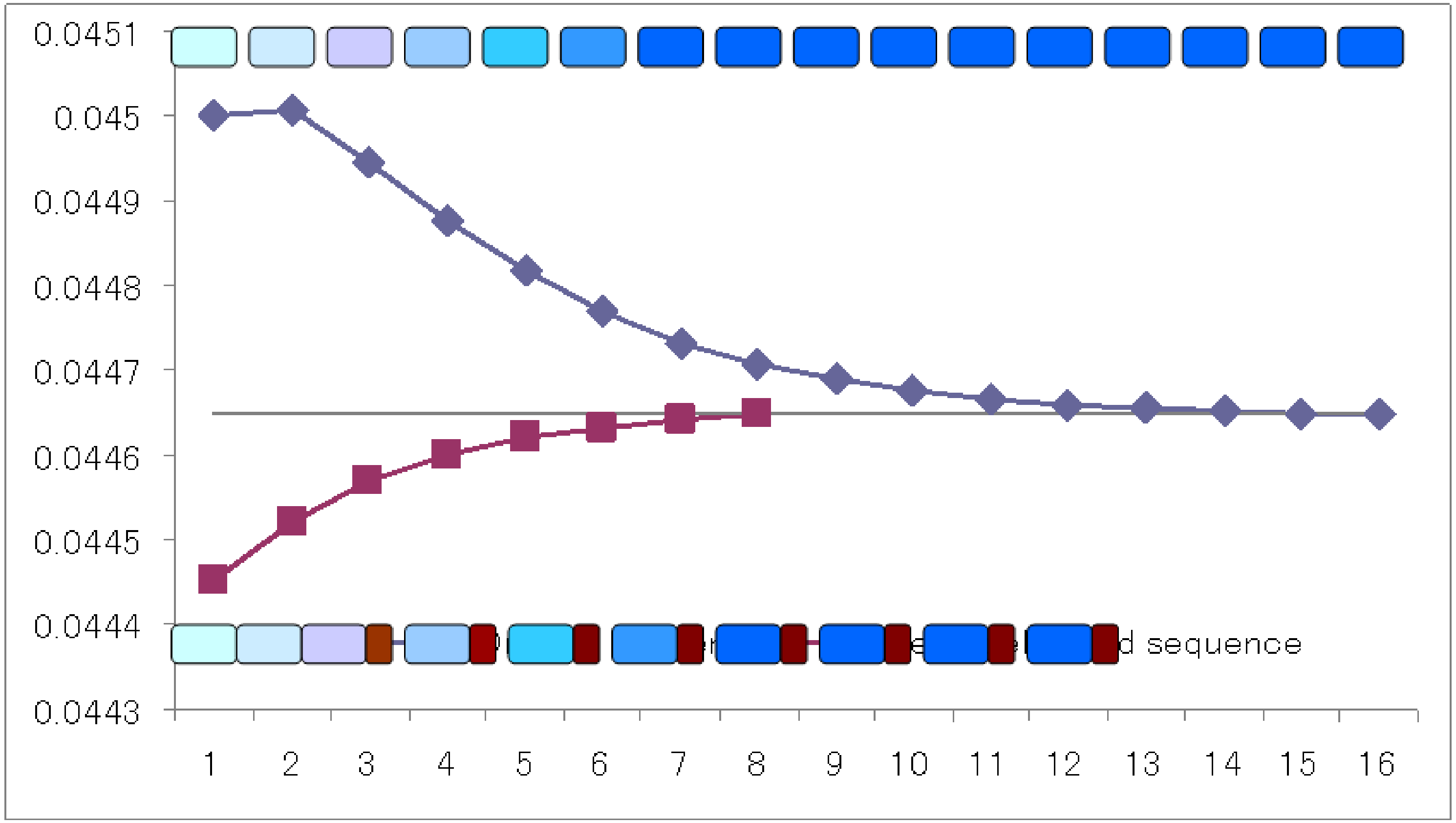
Acceleration by the vector $\varepsilon$ algorithm (# of iterations 3)

# Acceleration of PCA.ALS by the vector $\varepsilon$ accelerator



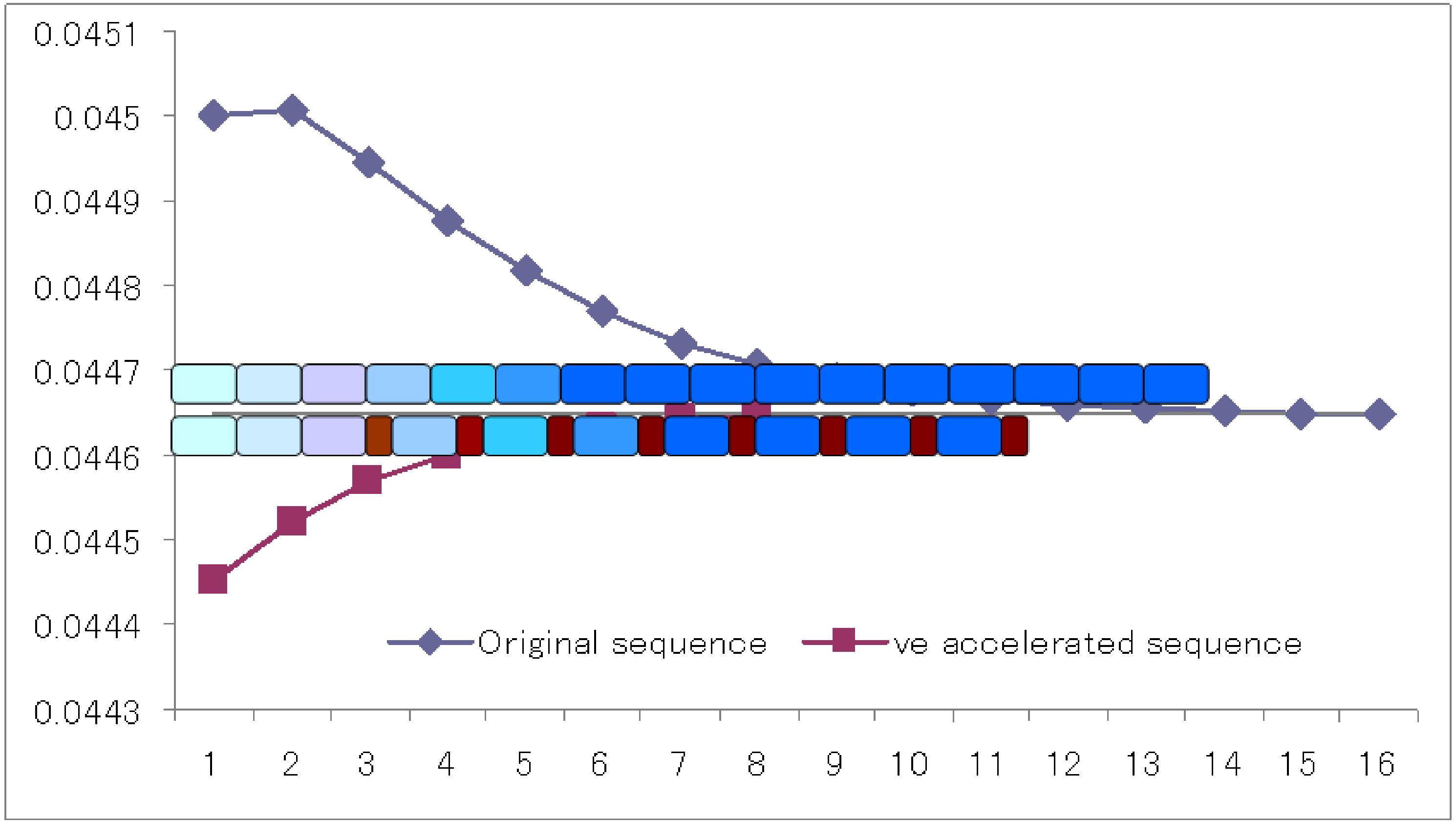
Acceleration by the vector $\varepsilon$ algorithm (# of iterations 4)

# Acceleration of PCA.ALS by the vector $\varepsilon$ accelerator



Acceleration by the vector $\varepsilon$ algorithm (time to convergence 1)

# Acceleration of PCA.ALS by the vector ε accelerator



Acceleration by the vector ε algorithm (time to convergence 2)

# Acceleration of PCA.ALS by the vector $\varepsilon$ accelerator

To accelerate PCA.ALS, we introduce the v$\varepsilon$ 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 $\{\dot{\mathbf{X}}^{*(t)}\}_{t\geq 0} = \{\dot{\mathbf{X}}^{*(0)}, \dot{\mathbf{X}}^{*(1)}, \cdots, \dot{\mathbf{X}}^{*(\infty)}\}$.**

## [ General procedure of v$\varepsilon$-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 v$\varepsilon$ algorithm:

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

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

## Acceleration of PCA.ALS by the vector $\varepsilon$ 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.

## Improvement of vε-PCA.ALS by using a restarting strategy

It may **not be needed to calculate** $\dot{\mathbf{X}}^{*(t)}$ in the *Acceleration step* within the **first several iterations**.

$\Downarrow$

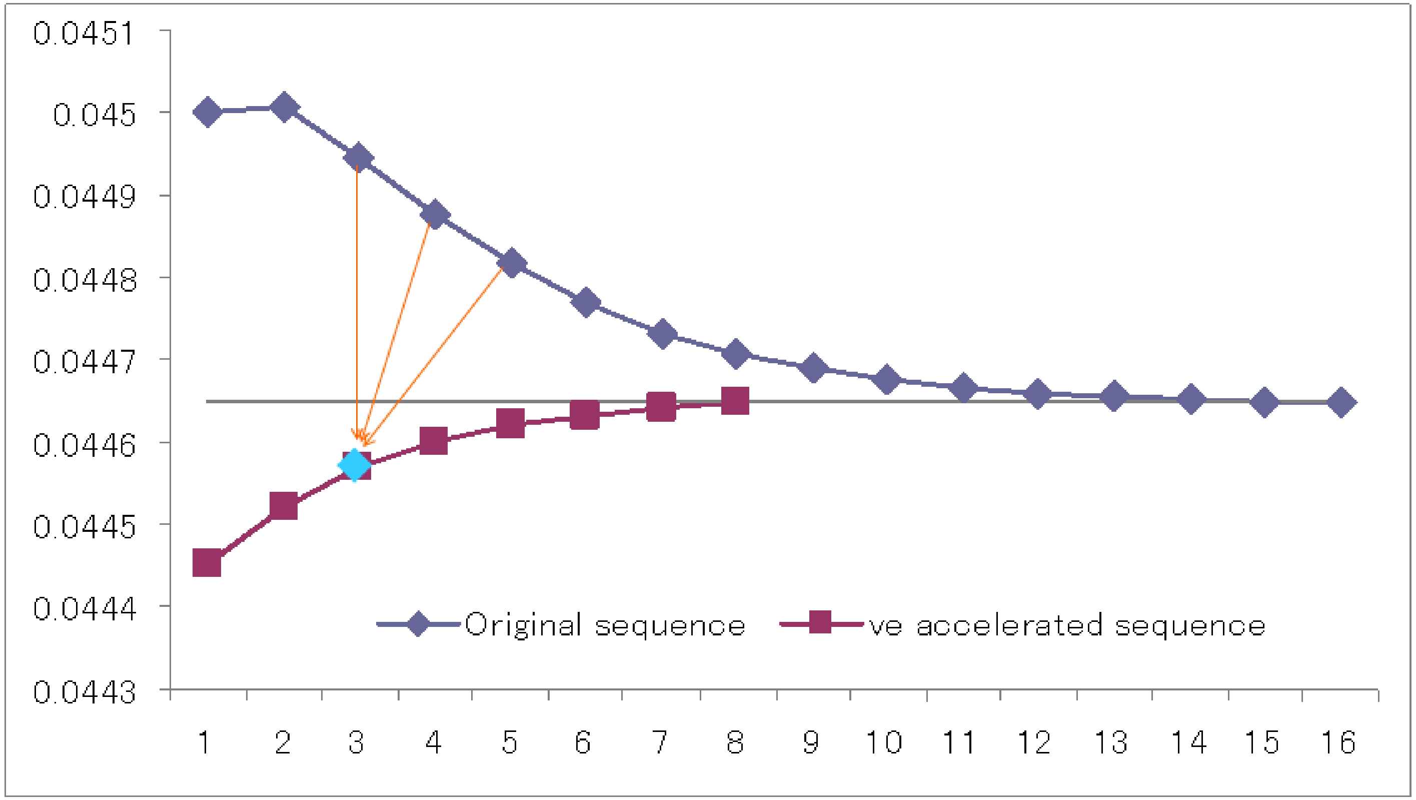
**New idea: Re-starting strategy**

- PCA.ALS iterations are continued until achieving restarting criteria,
- vε-PCA.ALS is re-started by using a new initial value of $\mathbf{X}^*$.

$\Downarrow$

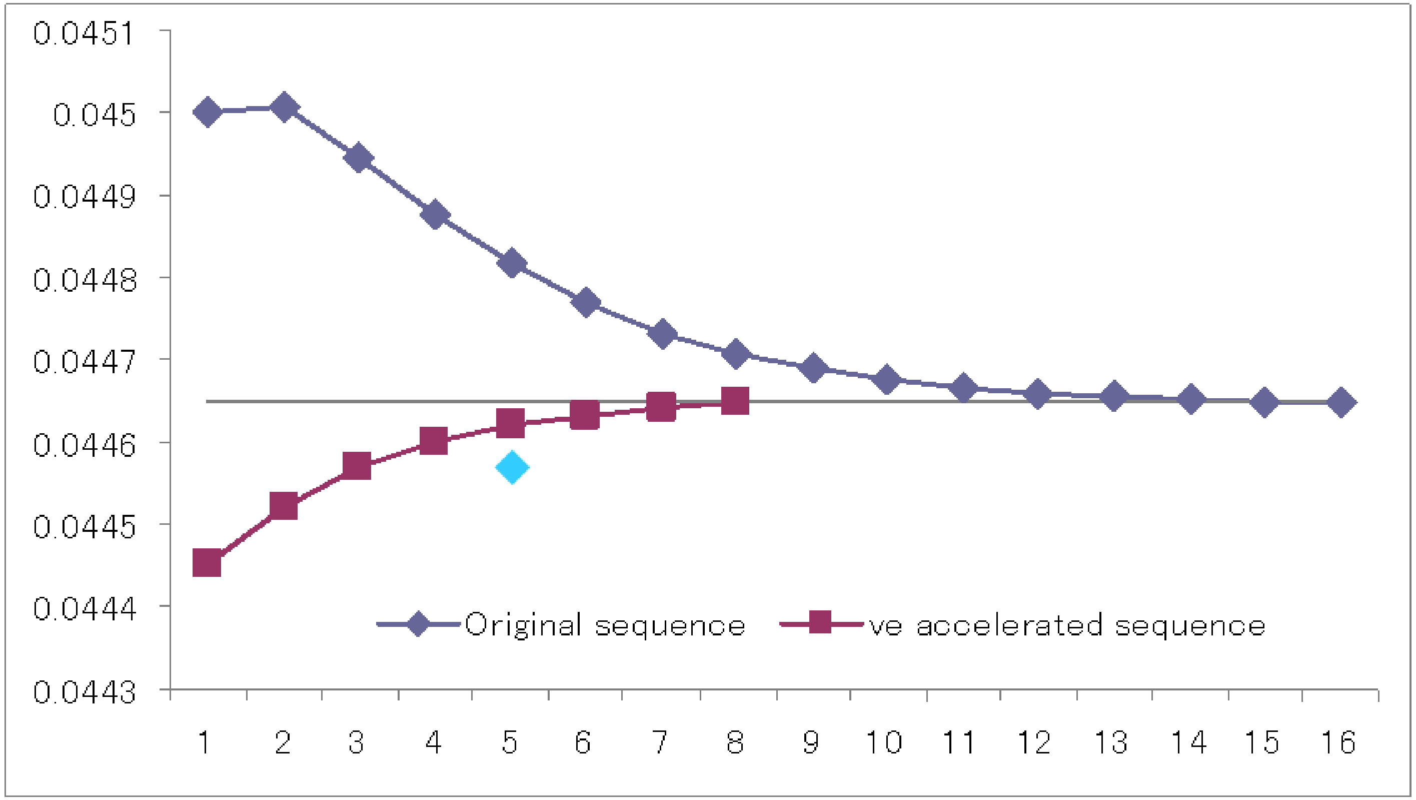
We decide the starting point of iteration of the *Acceleration step* and give the new initial value of $\mathbf{X}^*$.

# Improvement of vε-PCA.ALS by using a restarting strategy



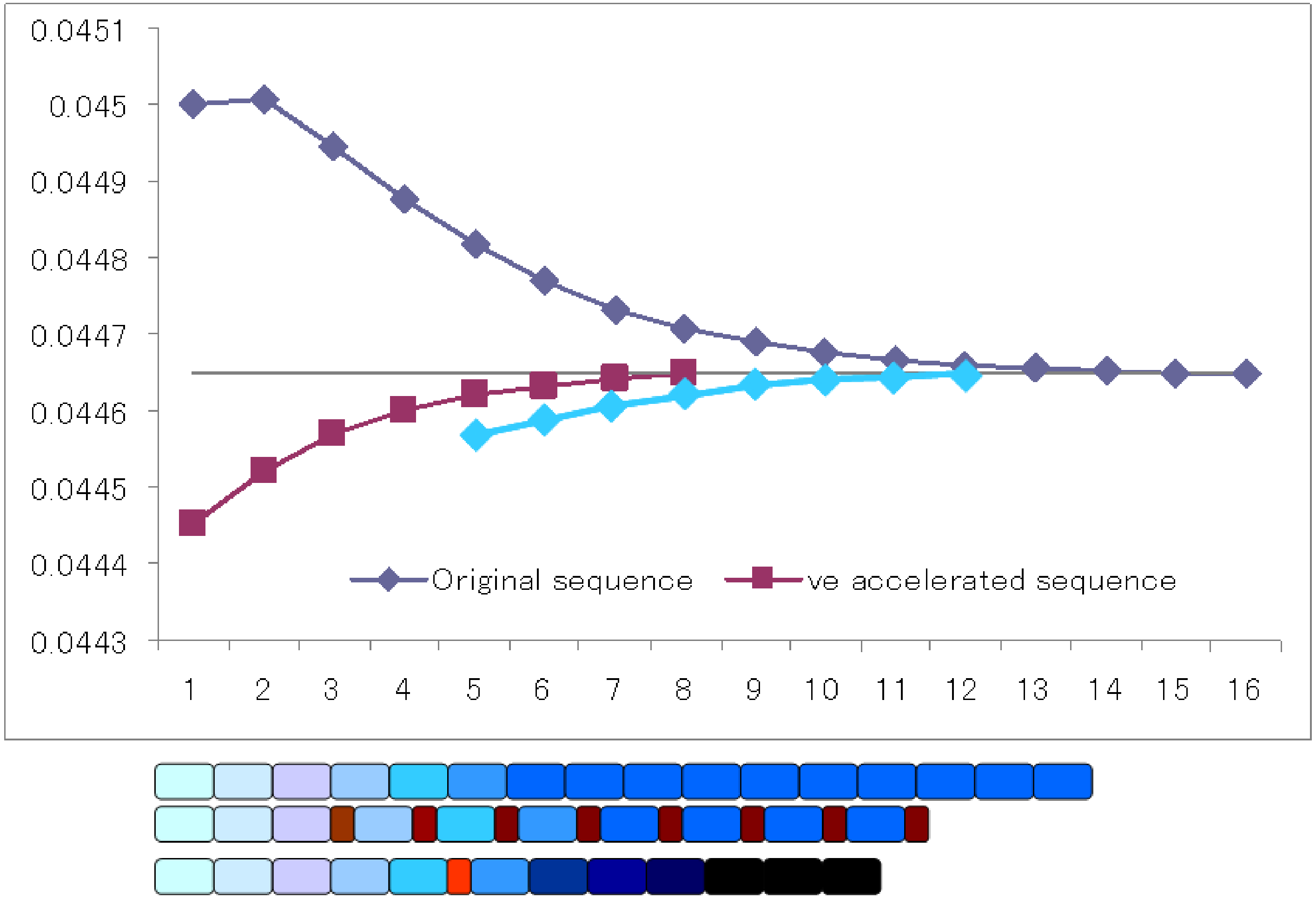
Re-starting strategy for vε algorithm

# Improvement of vε-PCA.ALS by using a restarting strategy



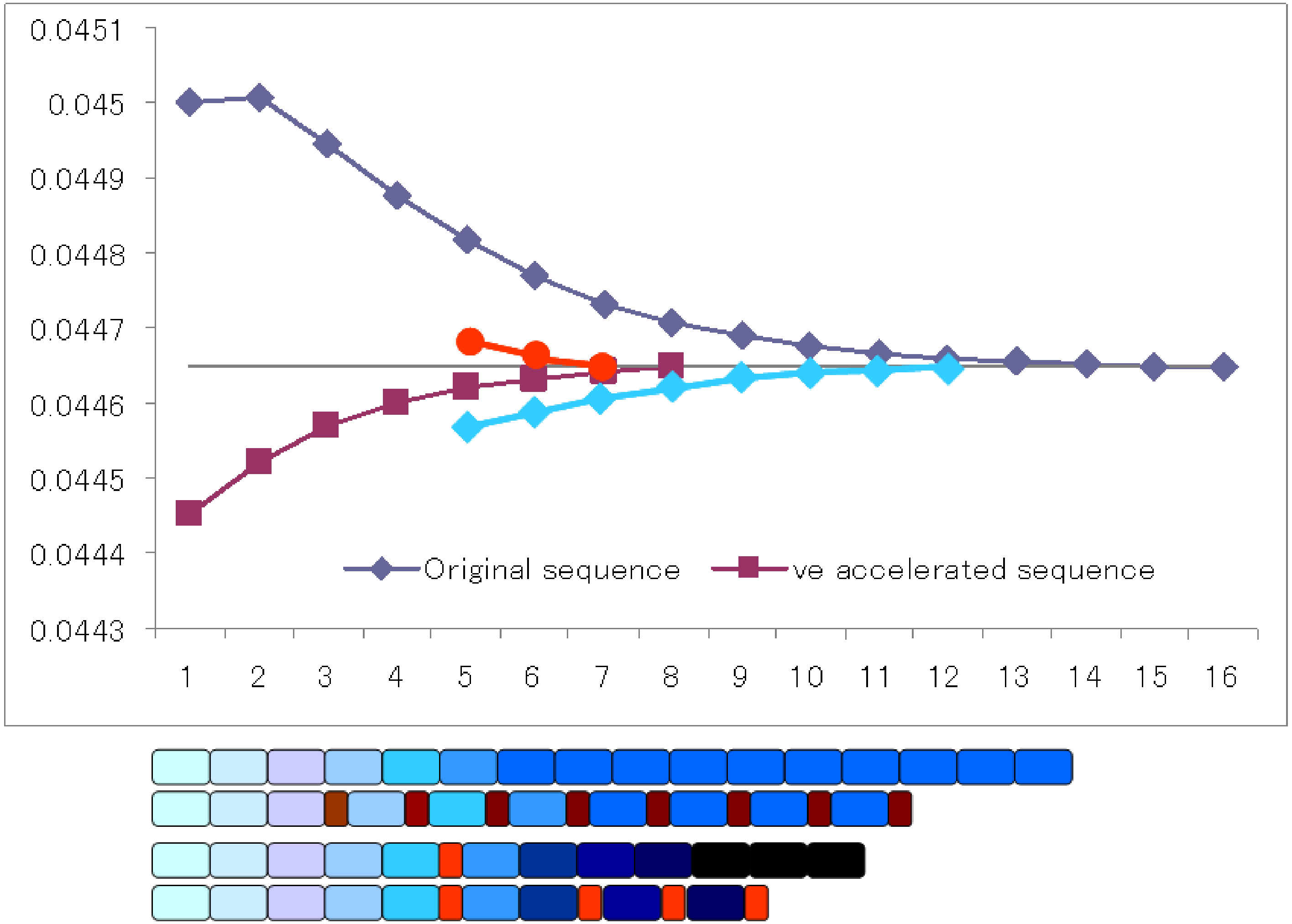
Re-starting strategy for vε algorithm

# Improvement of vε-PCA.ALS by using a restarting strategy



Re-starting strategy for vε algorithm (time to convergence 1)

# Improvement of vε-PCA.ALS by using a restarting strategy



Re-starting strategy for vε algorithm (time to convergence 2)

# Improvement of vε-PCA.ALS by using a restarting strategy

## [ New acceleration algorithm: r-vε-PCA.ALS ]

- *Single PCA.ALS step:*

  Repeat the following computation until $|\theta^{*(t+1)} - \theta^{*(t)}| < \delta_0$.

  – Estimate model parameters $\mathbf{A}^{(t)}$ and $\mathbf{Z}^{(t)}$ and determine optimal scaling parameter $\mathbf{X}^{*(t+1)}$. Calculate $\theta^{*(t+1)}$.

- *New initial value computation*:

  Compute $\dot{\mathbf{X}}^{*(T-2)}$ from

$$\mathrm{vec}\dot{\mathbf{X}}^{*(T-2)} = \mathrm{vec}\mathbf{X}^{*(T-1)} + \left[\left[\mathrm{vec}(\mathbf{X}^{*(T-2)} - \mathbf{X}^{*(T-1)})\right]^{-1} + \left[\mathrm{vec}(\mathbf{X}^{*(T)} - \mathbf{X}^{*(T-1)})\right]^{-1}\right]^{-1},$$

  and set $\mathbf{X}^{*(T+0)} = \dot{\mathbf{X}}^{*(T-2)}$, where $T$ is the number of iterations of *Single PCA.ALS step*.

# Improvement of vε-PCA.ALS by using a restarting strategy

- vε-*PCA.ALS step*:

  Set $t = 0$.

  Alternate the following two steps by using $\mathbf{X}^{*(T+t)}$ as the starting value:

  – Obtain $\mathbf{X}^{*(T+t+1)}$ from *PCA.ALS step*.

  – Compute $\dot{\mathbf{X}}^{*(T+t-1)}$ using $\{\mathbf{X}^{*(T+t-1)}, \mathbf{X}^{*(T+t)}, \mathbf{X}^{*(T+t+1)}\}$ in *Acceleration step* and check the convergence by

$$\|\text{vec}(\dot{\mathbf{X}}^{*(T+t-1)} - \dot{\mathbf{X}}^{*(T+t-2)})\|^2 < \delta.$$

# Improvement of vε-PCA.ALS by using a restarting strategy

## Computational advantage

When a good initial value is obtained for $\mathbf{X}^{*(T+0)}$ in *New initial value computation*, the following advantage is expected:

- r-vε-PCA.ALS converges faster than vε-PCA.ALS in terms of both the computational time and the number of iterations.

## Key point in r-vε-PCA.ALS

The performance of r-vε-PCA.ALS depends on the value of re-starting criteria $\delta_0$.

$\Downarrow$

It is a serious problem to find a optimal value of $\delta_0$.

### Outline of the restarting strategy of vε-PCA.ALS: r-vε-PCA.ALS

- Given an initial value $\mathbf{X}^{*(0)}$, we continue taking PCA.ALS as long as $|\theta^{*(t+1)} - \theta^{*(t)}|$ is greater than restarting criteria $\delta_0$.
- When this condition is violated, we compute a new initial value of $\mathbf{X}^*$ and start vε-PCA.ALS.

## Numerical experiments

Compare

- the number of total iterations
- total CPU time and CPU time per iteration
- CPU time speed-up

[ Data 1 ]: Real data
– Data: Evaluation of a course
– The size of sample $(n)$ : 56
– The number of items $(p)$ : 13 items with 5 levels (from 1 to 5)

[ Data 2 ]: Artificial data
– Data: Random data
– Replication: 50 times
– The size of sample $(n)$ : 60
– The number of items $(p)$ : 40 items with 10 levels (from 1 to 10)

# Numerical experiments: Data 1

The numbers of iterations and CPU times of
PRINCIPALS, vε-PRINCIPALS and r-vε-PRINCIPALS
($r = 2$ and $\delta = 10^{-8}$)

| $r$ | PRINCIPALS | | vε-PRINCIPALS | | r-vε-PRINCIPALS | |
|---|---|---|---|---|---|---|
| | Iter. | Time | Iter. | Time | Iter. | Time |
| 1 | 9 | 0.25 | 4 | 0.167 | 2 (4) | 0.222 |
| 2 | 92 | 2.52 | 23 | 0.704 | 9 (6) | 0.469 |
| 3 | 28 | 0.59 | 9 | 0.231 | 4 (3) | 0.194 |
| 4 | 25 | 0.74 | 7 | 0.276 | 3 (5) | 0.210 |
| 5 | 28 | 0.58 | 10 | 0.248 | 5 (3) | 0.207 |
| 6 | 29 | 0.61 | 9 | 0.251 | 4 (4) | 0.210 |
| 7 | 28 | 0.79 | 9 | 0.330 | 3 (4) | 0.254 |
| 8 | 47 | 1.07 | 14 | 0.373 | 7 (5) | 0.324 |
| 9 | 45 | 1.30 | 13 | 0.433 | 6 (5) | 0.380 |
| 10 | 45 | 0.88 | 14 | 0.323 | 7 (5) | 0.279 |
| 11 | 33 | 0.65 | 10 | 0.236 | 5 (3) | 0.200 |
| 12 | 40 | 1.11 | 10 | 0.333 | 6 (3) | 0.309 |

Each value in ( ) of the sixth column is the number of iterations of the *Single PRINCIPALS step* under the restarting criteria $\delta_0 = 1$.

## Numerical experiments: Data 2

CPU time speed-ups CPU time speed-ups from 50 simulated data ($r = 2$ and $\delta = 10^{-8}$).

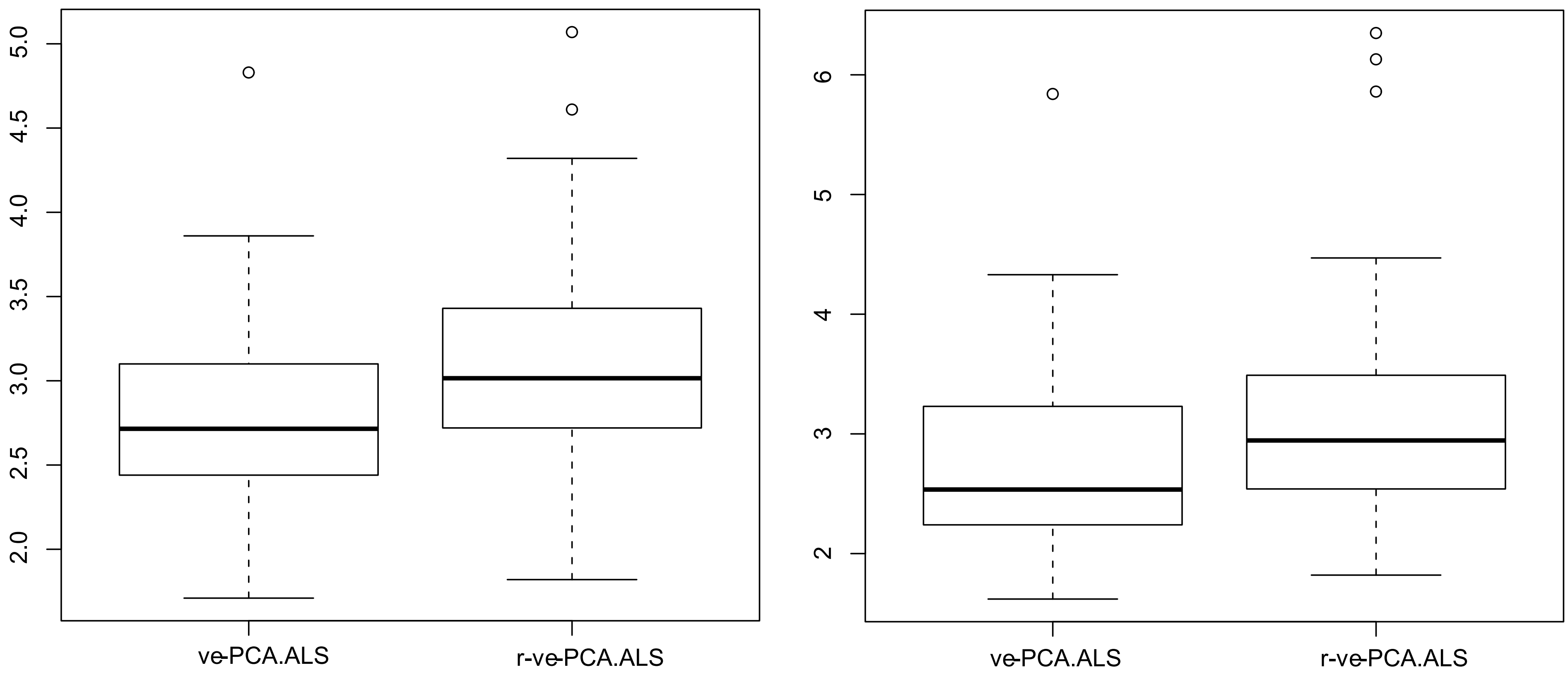
(a) $\delta_0 = 1.00$

| | | | Quantile | | |
|---|---|---|---|---|---|
| | Mean | [Min, Max] | 25% | 50% | 75% |
| vε-PCA.ALS | 2.79 | [1.71, 4.83] | 2.40 | 2.71 | 3.10 |
| r-vε-PCA.ALS | 3.08 | [1.82, 5.07] | 2.66 | 3.01 | 3.43 |

(b) $\delta_0 = 0.05$

| | | | Quantile | | |
|---|---|---|---|---|---|
| | Mean | [Min, Max] | 25% | 50% | 75% |
| vε-PCA.ALS | 2.74 | [1.62, 5.84] | 2.23 | 2.54 | 3.23 |
| r-vε-PCA.ALS | 3.14 | [1.82, 6.35] | 2.53 | 2.95 | 3.49 |

## Numerical experiments: Data 2



$\delta_0 = 1.0$

$\delta_0 = 0.05$

Boxplots of CPU time speed-ups from 50 simulated data

# Conclusion

From numerical experiments:

- Both two accelerated algorithms converge **3 to 4 times faster** than PRINCIPALS. Thus the new algorithm has the same performance of vε-PRINCIPAL in terms of the numbers of iterations.
- The computational times of r-vε-PRINCIPALS are **shorter** than those of vε-PRINCIPALS except $r = 1$.

$\Downarrow$

We can see that **the restating strategy works well** to reduce the computational time of vε-PRINCIPALS.

**[ Future problem ]**

In the experiments, the value of $\delta_0$ was decided roughly and thus it may not be optimal.

$\Downarrow$

It is a serious problem to find a optimal value of $\delta_0$ for large data sets.

$\Downarrow$

We intend to deduce criteria for $\delta_0$ systematically but not ad hoc.

References

# References

GIFI, A. (1989): Algorithm descriptions for ANACOR, HOMALS, PRINCIPALS, 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 epsilon algorithm. *Computational Statistics and Data Analysis 51, 1549-1561*.

KURODA, M., MORI, Y., IIZUKA, M. and SAKAKIHARA, M. (2008): Accerelation of convergence of the alternating least squares algorithm for principal component analysis. *Program & Abstracts IASC 2008, 172-172*.

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*.