GENERALIZED SUBSPACE RULES FOR ON-LINE PCA AND THEIR APPLICATION IN SIGNAL AND IMAGE COMPRESSION

Toshihisa Tanaka

Department of Electrical and Electronic Engineering,
Tokyo University of Agriculture and Technology (TUAT)
Nakacho, Koganei-shi, Tokyo, 184-8588, Japan. tanakat@cc.tuat.ac.jp

ABSTRACT

Weighted subspace (WS) algorithms developed by Oja and Xu for PCA are unified into generalized forms and theoretically analyzed. It is then proved that the generalized rules are stable at only the fixed point where the principal components are extracted. We moreover find the optimal parameter in terms of the preservation of orthogonality of estimated principal components during tracking. To understand the theoretical behavior, then, toy numerical examples are shown. Moreover, a possibility for the application of adaptive data compression is discussed, by showing examples of backward adaptation image coding.

1. INTRODUCTION

Principal component analysis (PCA), or equivalently called the Karhunen-Loève transform, is important and extensively used in the fields of signal compression, neural networks pattern recognition, and so on [1]. The principal components (PCs) can be obtained by solving an eigenvalue problem of the correlation matrix $\boldsymbol{R} = E_{\boldsymbol{X}}[\boldsymbol{x}\boldsymbol{x}^T]$, where $E_{\boldsymbol{X}}[\cdot]$ denotes the ensemble average corresponding to a stochastic vector $\boldsymbol{x}$ in the Euclidean space $\mathbb{R}^N$, and $\cdot^T$ denotes the transposition. We presume in this paper that $\boldsymbol{R}$ has full rank and distinct eigenvalues. Then, there exist the N eigenvectors $\boldsymbol{\phi}_i$, $i = 0, \ldots, N-1$ and the corresponding eigenvalues $\lambda_0 > \ldots > \lambda_{N-1} > 0$. The first K eigenvectors correspond to the K PCs and span the principal subspace of dimension K.

Estimation and tracking of the principal subspace/components by the gradient descent method can be accomplished by the update as

$$\boldsymbol{W}(n+1) = \boldsymbol{W}(n) + \beta_n \Delta \boldsymbol{W}(n) \tag{1}$$

where $\Delta\boldsymbol{W}(n)$ is the updating matrix, $\boldsymbol{W}(n)$ is the principal subspace/components estimate of size $N \times K$ at the nth iteration step, and β_n is a positive learning parameter. Several updating algorithms have been proposed. The well-known method for principal subspace tracking proposed by Oja [2] is given by

$$\Delta\boldsymbol{W}(n) = \boldsymbol{x}(n)\boldsymbol{y}^T(n) - \boldsymbol{W}(n)\boldsymbol{y}(n)\boldsymbol{y}^T(n), \tag{2}$$

where $\boldsymbol{y}(n) = \boldsymbol{W}^T(n)\boldsymbol{x}(n)$. By applying the direct differentiation to the cost function

$$J[\boldsymbol{W}] = E_{\boldsymbol{X}}\|\boldsymbol{x} - \boldsymbol{W}\boldsymbol{W}^T\boldsymbol{x}\|^2, \tag{3}$$

This work was partially supported by Japan Society for the Promotion of Science (JSPS), Grant-in-Aid for Scientific Research (C) (2), 15500101, 2003. He is also with Laboratory for Advanced Brain Signal Processing, RIKEN Brain Science Institute, Japan.

where $\boldsymbol{W}$ is of $N \times K$, we obtain the modified algorithm given as

$$\Delta\boldsymbol{W}(n) = 2\boldsymbol{x}(n)\boldsymbol{y}^T(n) - \boldsymbol{W}(n)\boldsymbol{y}(n)\boldsymbol{y}^T(n) - \boldsymbol{x}(n)\boldsymbol{y}^T(n)\boldsymbol{W}^T(n)\boldsymbol{W}(n), \tag{4}$$

which is proposed by Yang [3] and Xu [4], independently. This rule is sometimes referred to as normalized Oja's rule [5]. This rule also preserves orthogonality of $\boldsymbol{W}$ during learning. In the sense of the principal subspace tracking, it is reported in [5] that the learning rule given in (4) works better compared to Oja's rule as in (2). In many applications, however, adaptive rules which track not only a principal subspace but also PCs are desirable. For this purpose, Oja *et al* proposed in [6] a weighted version of the subspace rule given as

$$\Delta\boldsymbol{W}(n) = \boldsymbol{x}(n)\boldsymbol{y}^T(n) - \boldsymbol{W}(n)\boldsymbol{y}(n)\boldsymbol{y}^T(n)\boldsymbol{D}, \tag{5}$$

where $\boldsymbol{D}$ is a diagonal matrix with positive entries in increasing order $0 < d_0 < \cdots < d_{K-1}$. Similarly, Xu proposed the weighted subspace rule given as

$$\Delta\boldsymbol{W}(n) = \boldsymbol{x}(n)\boldsymbol{y}^T(n)\boldsymbol{D}^{-1} - \boldsymbol{W}(n)\boldsymbol{D}^{-1}\boldsymbol{y}(n)\boldsymbol{y}^T(n), \tag{6}$$

and its normalized version given as

$$\begin{aligned}\Delta\boldsymbol{W}(n) \quad = \quad & 2\boldsymbol{x}(n)\boldsymbol{y}^T(n)\boldsymbol{D}^{-1} - \boldsymbol{W}(n)\boldsymbol{D}^{-1}\boldsymbol{y}(n)\boldsymbol{y}^T(n) \\ & - \boldsymbol{x}(n)\boldsymbol{y}^T(n)\boldsymbol{D}^{-1}\boldsymbol{W}^T(n)\boldsymbol{W}(n).\end{aligned} \tag{7}$$

As we have seen, the weighted learning rules for PCA summarized here can be categorized into two groups: weighted subspace (WS) rules, and the normalized weighted subspace (NWS) rules. Based on these observation, in this paper, we propose a unified approach of WS rules that allow individual PCs to be extracted. In the rest of the paper, firstly, we propose generalized forms for the two groups of weighted rules. Secondly, it is proved that the generalized rules are stable at only the point where the PCs are extracted. It is finally shown that in special cases, orthogonality of $\boldsymbol{W}$ is preserved during the learning. It is also proved that in these cases, the normalized rule preserves orthogonality more strongly that the non-normalized rules. Simple numerical examples are provided for illustrating theoretically justified properties of the proposed rules. Moreover, we provide examples of the application to backward adaptation of transform image coding. We would understand theoretical results through the numerical simulations.

2. GENERALIZED WEIGHTED SUBSPACE RULES AND THEIR STABILITY

The generalized forms for non-normalized and normalized weighted subspace rules are derived in this section. By replacing $\boldsymbol{W}\boldsymbol{D}^{-1/2}$ by

$\boldsymbol{W}$ in the Xu's weighted rule as in (6), we have the modified form given as

$$\Delta \boldsymbol{W}(n) = \boldsymbol{x}(n)\boldsymbol{y}^T(n)\boldsymbol{D}^{-1} - \boldsymbol{W}(n)\boldsymbol{y}(n)\boldsymbol{y}^T(n) \quad (8)$$

In the same way as the above, the normalized version as in (7) becomes

$$\Delta \boldsymbol{W}(n) = \boldsymbol{x}(n)\boldsymbol{y}^T(n)\boldsymbol{D}^{-1} - \boldsymbol{W}(n)\boldsymbol{y}(n)\boldsymbol{y}^T(n) - \boldsymbol{x}(n)\boldsymbol{y}^T(n)\boldsymbol{W}^T\boldsymbol{W}. \quad (9)$$

Comparing Xu's rules (8) and (9) with Oja's rule (2), we may unify them and generalize into the following unified rules:

- Generalized WS (GWS) rule:

$$\Delta \boldsymbol{W}(n) = \boldsymbol{x}(n)\boldsymbol{y}^T(n)\boldsymbol{D}^{-p} - \boldsymbol{W}(n)\boldsymbol{y}(n)\boldsymbol{y}^T(n)\boldsymbol{D}^{1-p}, \quad (10)$$

- Generalized NWS (GNWS) rule:

$$\Delta \boldsymbol{W}(n) = \boldsymbol{x}(n)\boldsymbol{y}^T(n)\boldsymbol{D}^{-p} - \boldsymbol{W}(n)\boldsymbol{y}(n)\boldsymbol{y}^T(n)\boldsymbol{D}^{1-p} - \boldsymbol{x}(n)\boldsymbol{y}^T(n)\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{D}^{1-p}. \quad (11)$$

where p is a scalar constant, which will change the learning behavior.

If we set in the GWS rule that $p = 0$ and $p = 1$, we obtain the Oja's WS rule (5) and the Xu's WS rule (6), respectively. Furthermore, if we set in the GNWS rule that $p = 1$, we obtain the Xu's NWS rule (9).

In order to analyze the stochastic behaviors of the algorithms, as done, for instance, in [1, 4, 6], we introduce the corresponding ordinary differential equations (ODE) given as

$$\frac{d\boldsymbol{W}}{dt} = \boldsymbol{R}\boldsymbol{W}\boldsymbol{D}^{-p} - \boldsymbol{W}\boldsymbol{W}^T\boldsymbol{R}\boldsymbol{W}\boldsymbol{D}^{1-p}, \quad (12)$$

for the GWS rule and

$$\frac{d\boldsymbol{W}}{dt} = 2\boldsymbol{R}\boldsymbol{W}\boldsymbol{D}^{-p} - \boldsymbol{W}\boldsymbol{W}^T\boldsymbol{R}\boldsymbol{W}\boldsymbol{D}^{1-p} - \boldsymbol{R}\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{D}^{1-p}, \quad (13)$$

for the GNWS rule.

First of all, we find the fixed points of the gradients (10) and (11). Assume that $\boldsymbol{R}$ has nonzero distinct eigenvalues $\lambda_0 > \lambda_1 > \cdots > \lambda_{N-1}$. Let $\boldsymbol{\Lambda}$ be a diagonal matrix given as $\boldsymbol{\Lambda} = \mathrm{diag}[\lambda_{\pi(0)}, \ldots, \lambda_{\pi(N-1)}]$, where π is a permutation of $\{0, \ldots, N-1\}$, that is, $\{\pi(0), \ldots, \pi(N-1)\} = \{0, \ldots, N-1\}$, and diag$[\cdot]$ denotes a diagonal matrix. Then, an eigenvalue decomposition may be given as $\boldsymbol{R} = \boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P}^T$, where $\boldsymbol{\Lambda} = \mathrm{diag}[\lambda_{\pi(0)}, \ldots, \lambda_{\pi(N-1)}]$. Now, we present the following result.

Theorem 1 *Let $\bar{\boldsymbol{P}}$ be an $N \times K$ matrix whose columns correspond to eigenvectors of $\boldsymbol{R}$ in arbitrary order without duplication. The fixed point is then given by $\boldsymbol{W} = \bar{\boldsymbol{P}}\boldsymbol{D}^{-1/2}$.*

Proofs in this paper are omitted due to lack of space. All the proofs are provided in [7].

Keep in mind that the theorem means that at any fixed point, $\boldsymbol{W}$ is in the manifold S.

2.1. Stability of the Fixed Points

We investigate in this section the stability of the fixed points. Firstly, we show the main result as follows.

Theorem 2 *The rules (10) and (11) are stable if and only if $\boldsymbol{W}$ is at the fixed point $\boldsymbol{P}^*\boldsymbol{D}^{-1/2}$, where $\boldsymbol{P}^*$ is the $M \times K$ matrix whose columns are the K PCs in decreasing order of the eigenvalues, that is, $\boldsymbol{P}^* = [\boldsymbol{\phi}_0, \ldots, \boldsymbol{\phi}_{K-1}]$.*

2.2. Stability on the Manifold

We investigate the behavior of the dynamics of $(\boldsymbol{W}^T\boldsymbol{W})(t)$ in this subsection. It is shown that in some particular cases, the manifold S is stable under the rules (10) and (11). The stability of the manifold is a desirable property for adaptive algorithms, since no additional normalization, which would make the system more complicated, is needed.

Before the stability analysis on the manifold S, we find the special cases where the manifold S is stable. Assume that at the nth iteration, $\boldsymbol{W}(n)$ is in the manifold S, that is, $\boldsymbol{W}^T(n)\boldsymbol{W}(n) = \boldsymbol{D}^{-1}$. Then, the GNWS rule (11) becomes identical to the GWS rule (10). From (1) and (10), we have

$$\begin{aligned}\boldsymbol{W}^T(n+1)\boldsymbol{W}(n+1) &= \boldsymbol{D}^{-1} + \beta\Big[(\boldsymbol{D}^{-p}\boldsymbol{W}^T\boldsymbol{R}\boldsymbol{W} + \boldsymbol{W}^T\boldsymbol{R}\boldsymbol{W}\boldsymbol{D}^{-p}) \\ &\quad - (\boldsymbol{D}^{1-p}\boldsymbol{W}^T\boldsymbol{R}\boldsymbol{W}\boldsymbol{D}^{-1} + \boldsymbol{D}^{-1}\boldsymbol{W}^T\boldsymbol{R}\boldsymbol{W}\boldsymbol{D}^{1-p})\Big] + O(\beta^2), \quad (14)\end{aligned}$$

where we omit the iteration index n in the right hand side. It should be here noted that the signal is not necessarily stationary. In order that $\boldsymbol{W}^T(n+1)\boldsymbol{W}(n+1)$ is still in the manifold S under the assumption that β^2 is negligible small, it must hold that

$$\boldsymbol{D}^{-p}\boldsymbol{W}^T\boldsymbol{R}\boldsymbol{W} + \boldsymbol{W}^T\boldsymbol{R}\boldsymbol{W}\boldsymbol{D}^{-p} = \boldsymbol{D}^{1-p}\boldsymbol{W}^T\boldsymbol{R}\boldsymbol{W}\boldsymbol{D}^{-1} + \boldsymbol{D}^{-1}\boldsymbol{W}^T\boldsymbol{R}\boldsymbol{W}\boldsymbol{D}^{1-p}. \quad (15)$$

If $\boldsymbol{W}^T\boldsymbol{R}\boldsymbol{W}$ is diagonal, the equality holds; however, the diagonality is guaranteed at the fixed points of $\boldsymbol{W}$. We have now the following result of the preservation of orthogonality:

Theorem 3 *By the update with a small parameter β, it holds that*

$$\boldsymbol{W}^T(n+1)\boldsymbol{W}(n+1) = \boldsymbol{D}^{-1} + O(\beta^2), \quad (16)$$

if and only if $K = 1$ or $p = 1$.

If we choose $K = 1$, $\boldsymbol{W}^T\boldsymbol{R}\boldsymbol{W}$ is always a scalar, and then the equality holds for arbitrary p. If $K \neq 1$, which is a more general case, p must be unity so that the equality holds.

If a signal is stationary, we can provide a precise discussion and a stronger result of the stability analysis on the manifold in this special case $K = 1$ or $p = 1$.

Theorem 4 *In the learning rules (10) and (11), the manifold is stable under perturbation when $K = 1$ or $p = 1$.*

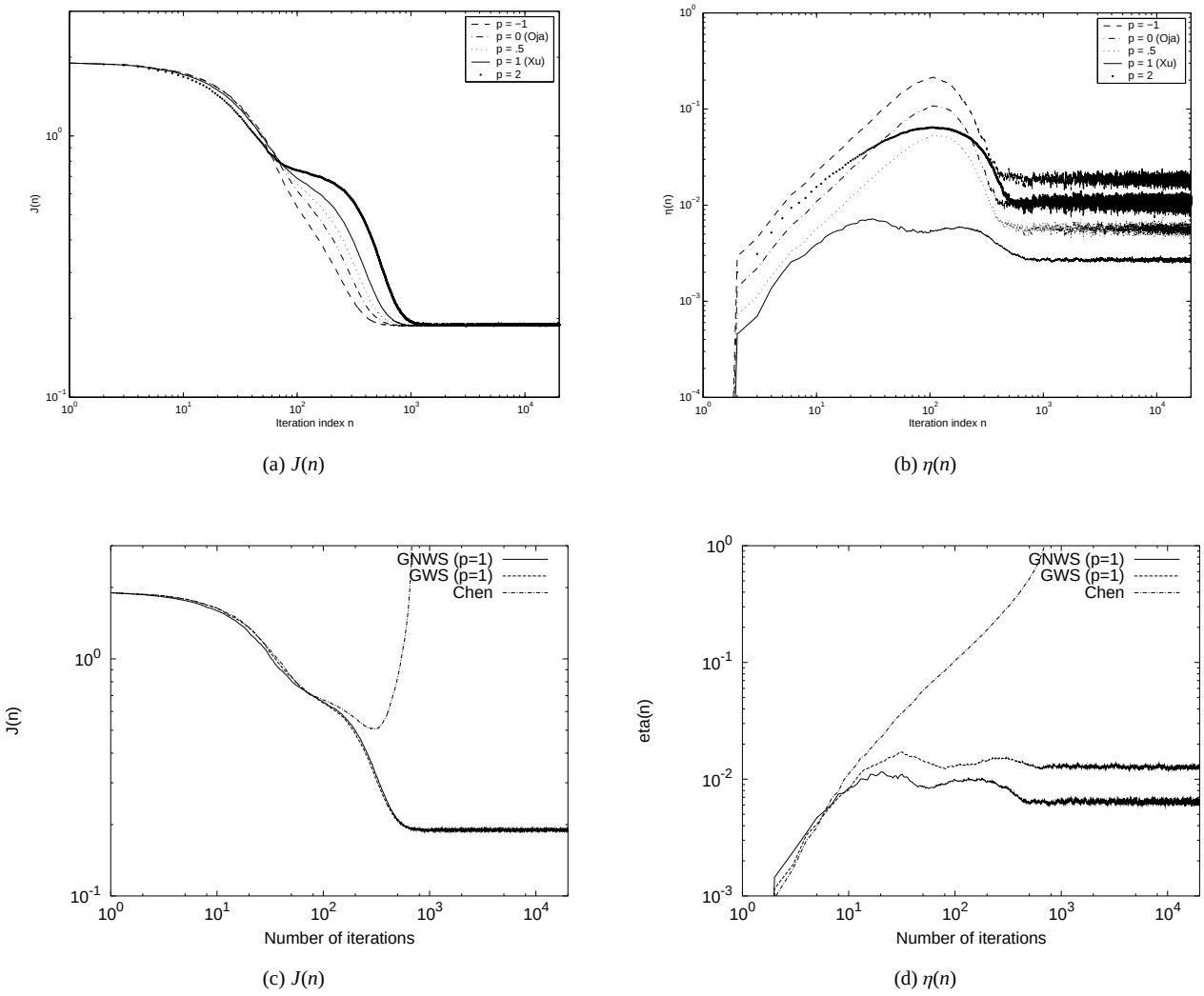
Furthermore, the perturbation in (11) decays to zero faster than that in (10).

3. NUMERICAL EXAMPLES

3.1. Random Vectors Generated from One Source

In this section, we show a simulation study of the proposed rules for a better understanding of the analytical results. We choose $\boldsymbol{x}(n)$ to be a sequence of independent jointly-Gaussian random vectors with the correlation matrix

$$\boldsymbol{R} = \begin{bmatrix} 0.9 & 0.4 & 0.7 & 0.3 \\ 0.4 & 0.3 & 0.5 & 0.4 \\ 0.7 & 0.5 & 1.0 & 0.6 \\ 0.3 & 0.4 & 0.6 & 0.9 \end{bmatrix}. \quad (17)$$



(a) $J(n)$ (b) $\eta(n)$

(c) $J(n)$ (d) $\eta(n)$

Fig. 1. Average behaviors for different values of p under the GNWS rule ((a) and (b)). Average behaviors of the GWS rule ($p = 1$), the GNWS rule ($p = 1$), and the Chen's rule [8] ((c) and (d)).

We choose $K = 2, \beta = 0.01$, and $\boldsymbol{D} = \text{diag}[d_0, d_1] = \text{diag}[0.51, 1.01]$. For evaluation of the algorithms, we compute the ensemble averages of the performance factors as follows:

$$J(n) = \text{tr}[(\boldsymbol{I} - \boldsymbol{P}(n)\boldsymbol{P}^T(n))\boldsymbol{R}(\boldsymbol{I} - \boldsymbol{P}(n)\boldsymbol{P}^T(n))^T], \quad (18)$$

$$\eta(n) = \|\boldsymbol{P}^T(n)\boldsymbol{P}(n) - \boldsymbol{I}\|_F^2 \quad (19)$$

for each algorithm using 100 independent runs, where $\boldsymbol{P}(n) = \boldsymbol{W}(n)\boldsymbol{D}^{1/2}$, $\|\cdot\|_F$ denotes the Frobenius norm, tr[·] is the trace, and $\boldsymbol{D}^{1/2}$ is a diagonal matrix with the entries $\sqrt{d_0}, \ldots, \sqrt{d_{K-1}}$.

Figures 1(a) and 1(b) illustrate the average behaviors of the cost functions $J(n)$ and $\eta(n)$ for different values p under the GNWS rule, respectively. We examine the cases where $p = -1$, 0, 0.5, 1, and 2. The GNWS rule when $p = 0.5$ is identical to the Xu's weighted rule (8). The difference of behavior is observed both in Figs. 1(a) and 1(b). Interestingly, it can also be observed from Fig. 1(b) that the GNWS rule with $p = 1$ preserves orthogonality more stronger than the GNWS rules with $p \neq 1$. Theorem 4 can theoretically justify this fact.

Then, we show the difference between the GWS and the GNWS rules in Fig. 1(c) and 1(d). The behavior of the Chen's rule [8] is also depicted. It can be observed that the Chen's rule diverges rapidly. The difference of behavior between the GWS and the GNWS rules is small; however, it should be noted that we can observe the distinct difference in the behavior of η as seen in Fig. 1(d). The GNWS rule keeps orthogonality more strongly than the GWS rule. This observation coincides with the theoretical result of Theorem 4.

3.2. Backward Adaptation in the Presence of Quantization

We next show an application of those rules to a backward adaptive transform system [9]. Block diagram of transform coding system with backward adaptive transform updates is illustrated in Fig. 2. In this adaptive system, the transform matrix $\boldsymbol{W}(n-1)$ at the $(n-1)$th iteration is updated with the quantized vector $\hat{\boldsymbol{y}}(n)$; therefore,

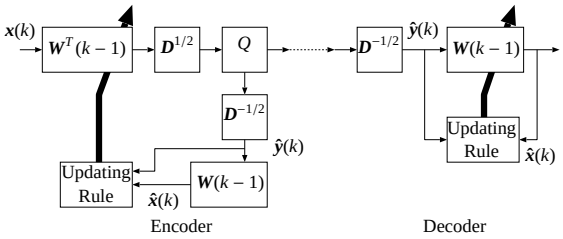


Fig. 2. Backward adaptive transform system.

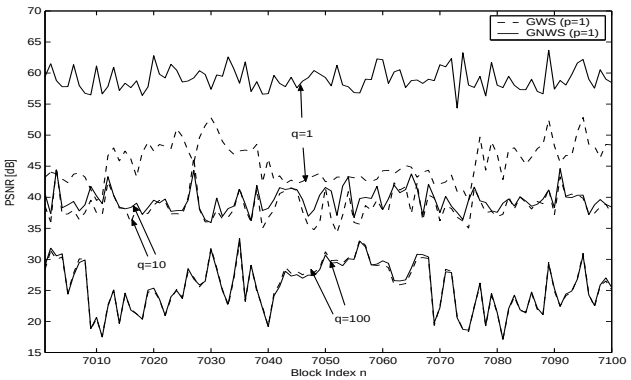


Fig. 3. PSNR at nth blocks in dB for "Lena."

the decoder can update the transform matrix in the same way as the encoder does. Specifically, the quantized vector $\hat{\boldsymbol{y}}(n)$ is obtained by the quantization operator (non-linear) defined as

$$\hat{\boldsymbol{y}} = \boldsymbol{D}^{-1/2} Q[\boldsymbol{D}^{1/2}\boldsymbol{y}] = q\boldsymbol{D}^{-p/2}\text{round}(1/q\boldsymbol{D}^{1/2}\boldsymbol{y}), \tag{20}$$

where round$(\cdot)$ is the rounding operator for vector elements and q is the step size of quantization. The update in (10) and (11) is performed by $\hat{\boldsymbol{y}}(n)$ and $\hat{\boldsymbol{x}}(n) = \boldsymbol{W}(n-1)\hat{\boldsymbol{y}}(n)$ instead of $\boldsymbol{y}(n)$ and $\boldsymbol{x}(n)$, respectively.

We compare two rules in the backward adaptation system. For simplicity, we examine the one-dimensional case where an input image is raster-scanned horizontally and segmented into blocks of M samples. Figure 3 depicts the comparison of PSNRs for the test image "Lena," which is of size 512×512. In this figure, PSNRs of 7001st to 7100th blocks, i.e., $\text{PSNR}(n) = -20\log_{10}\dfrac{\|\boldsymbol{x}(n) - \hat{\boldsymbol{x}}(n)\|}{255 \times M}$ for $7001 \le n \le 7100$, are plotted. It is clearly observed that the GNWS provides better performance in PSNR that the GWS for $q = 1$ and $q = 10$. The difference is significant in the case $q = 1$. To understand the reason for this effect, we illustrate the plots of $\eta(n)$ for $1 \le n \le 9000$ and $q = 1$ in Fig. 4, which shows that the GWS is less orthogonal than the GNWS during the iteration. This fact may result in the difference of PSNRs between two rules.

4. CONCLUSIONS

In this paper, the framework for weighted subspace (WS) rules for PCA have been introduced. The proposed form includes the conventional WS rules. The stability of the algorithms have been analyzed. The special case of the proposed form such that orthogonality of the estimated matrix is preserved has been found. We conclude that the GNWS rule with the parameter $p = 1$ (Xu's normalized rule) is the most promising among the rules dealt with in the paper. Future work may involve the development of fast and orthogonal algorithms as done in [5, 10]

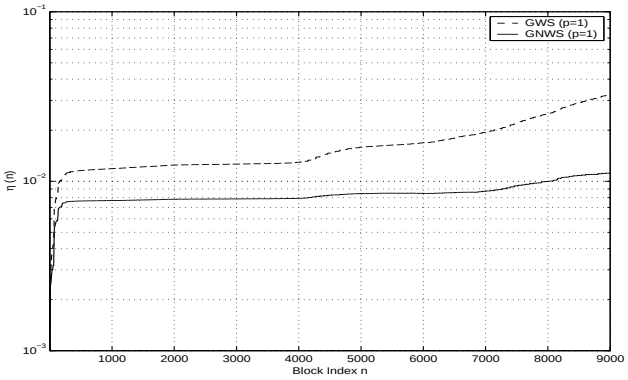


Fig. 4. Stability of orthogonality in the "Lena" image.

5. REFERENCES

[1] K. I. Diamantaras and S. Y. Kung, *Principal Component Neural Networks: Theory and Applications*. New York: Wiley, 1996.

[2] E. Oja, "Neural networks, principal components, and subspaces," *Int. J. Neural Systems*, vol. 1, pp. 61–68, Apr. 1989.

[3] B. Yang, "Projection approximation subspace tracking," *IEEE Trans. Signal Processing*, vol. 43, pp. 95–107, Jan. 1995.

[4] L. Xu, "Least mean square error reconstruction principle for self-organizing neural-nets," *Neural Networks*, vol. 6, pp. 627–648, 1993.

[5] S. Attallah and K. Abed-Meraim, "Fast algorithms for subspace tracking," *IEEE Signal Processing Letters*, vol. 8, pp. 203–206, July 2001.

[6] E. Oja, H. Ogawa, and J. Wangviwattana, "Principal component analysis by homogeneous neural networks, Part I: The weighted subspace criterion," *IEICE Trans. Inf. & Syst.*, vol. E75-D, pp. 366–375, May 1992.

[7] T. Tanaka, "Generalized weighted rules for principal components tracking," *to appear in IEEE Trans. Signal Processing*, 2004.

[8] T. Chen, S. Amari, and Q. Lin, "A unified algorithm for principal and minor components extraction," *Neural Networks*, vol. 11, no. 3, pp. 385–390, 1998.

[9] V. K. Goyal, J. Zhuang, and M. Vetterli, "Transform coding with backward adaptive updates," *IEEE Trans. Information Theory*, vol. 46, pp. 1623–1633, Feb. 1997.

[10] K. Abed-Meraim, S. Attallah, A. Chkeif, and Y. Hua, "Orthogonal Oja algorithm," *IEEE Signal Processing Letters*, vol. 7, pp. 116–119, May 2000.