Received: 4 March 2017 | Revised: 7 March 2018 | Accepted: 3 May 2018

DOI: 10.4218/etrij.2017-0304

ORIGINAL ARTICLE

WILEY ETRI Journal

Projection spectral analysis: A unified approach to PCA and ICA with incremental learning

Hoon Kang | **Hyun Su Lee**

School of Electrical and Electronics Engineering, Chung-Ang University, Seoul, Rep. of Korea.

Correspondence
Hoon Kang, School of Electrical and Electronics Engineering, Chung-Ang University, Seoul, Rep. of Korea.
Email: hkang@cau.ac.kr

Funding Information
This research was supported by the National Research Foundation of Korea (NRF), grant no. NRF-2017R1D1A1B03036450, and Chung-Ang University Graduate Research Scholarship in 2016.

Projection spectral analysis is investigated and refined in this paper, in order to unify principal component analysis and independent component analysis. Singular value decomposition and spectral theorems are applied to nonsymmetric correlation or covariance matrices with multiplicities or singularities, where projections and nilpotents are obtained. Therefore, the suggested approach not only utilizes a sum-product of orthogonal projection operators and real distinct eigenvalues for squared singular values, but also reduces the dimension of correlation or covariance if there are multiple zero eigenvalues. Moreover, incremental learning strategies of projection spectral analysis are also suggested to improve the performance.

KEYWORDS
independent component analysis, machine learning, neural network, principal component analysis, projection spectral analysis, singular value decomposition, spectral theorem

1 | INTRODUCTION

Projection spectral analysis [1] (PSA) is a useful tool that can be applied to unsupervised learning in the field of neural networks, where we deal with correlation or covariance matrices. PSA is mathematically a class of spectral decompositions [2] through which we may use the resolution of identity and obtain projections and/or nilponents for eigenvalues. In this paper, PSA will be extended to a unified paradigm of machine learning, including both principal component analysis (PCA) and independent component analysis (ICA), by exploiting singular value decomposition. Moreover, it is a generalized method of analyzing eigenstructures that are neither necessarily independent nor orthogonal, since multiple eigenvalues are considered, including singularities, in these matrices. Correlation or covariance matrices are widely used in the field of neural networks, especially in Hebbian learning [3], Hopfield memories [4], bidirectional associative memories [5], PCA [6], ICA [7,8], fast ICA [9], multilayer associative neural networks [10], Hinton's deep learning architectures [11,12], convolutional neural networks [13,14], and others [15]. Most of these neural networks are constructed in terms of correlation or covariance matrices, whose performance depends on the eigen-pairs, geometrically aligned to form localized minima of the energy terrains.

We begin by defining our network structure for PSA and then explain its mathematical foundations. The correlation or covariance matrix, the weight matrix A_K, of input-hidden layers may be represented by

$$A_K = \frac{1}{K}[YX^T] = \frac{1}{K}\sum_{i=1}^{K} y_i x_i^T \in \Re^{m\times n}, \tag{1}$$

where K is the number of data pairs, $x_i \in \Re^{n\times 1}$ is the ith input data, $y_i \in \Re^{m\times 1}$ is the ith hidden data, $X = [x_1 \cdots x_K] \in \Re^{n\times K}$, and $Y = [y_1 \cdots y_K] \in \Re^{m\times K}$ [1]. The hidden data y_i constitutes an orthogonal set of kernel functions, $y_i^T y_j = m\delta_{ij}$, ascribed to a Haar wavelet; however, it is possible to adopt other wavelet functions. Here, δ_{ij} is the Kronecker delta function. If the training output is $z_i \in \Re^{p\times 1}$, the weight matrix W_K of the output layer is described by

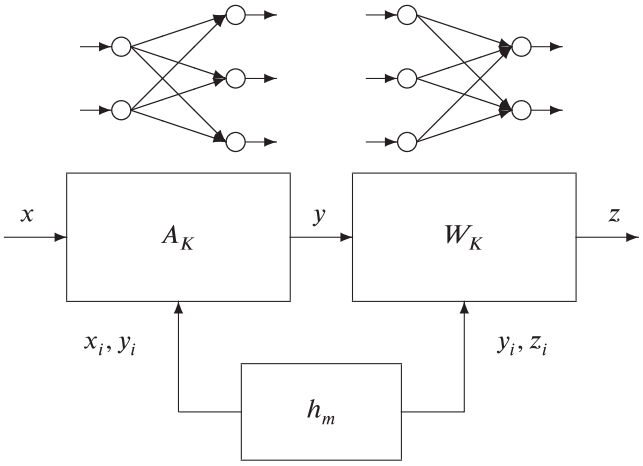

FIGURE 1 Block diagram of projection spectral analysis (PSA)

$$W_K = \frac{1}{m}\sum_{i=1}^{K} z_i y_i^T \in \Re^{p\times m}, \tag{2}$$

where $m = 2^{\text{ceil}(\log_2 K)}$ (ceil(a) is a function generating an integer larger than real a) since all the training data should have a one-to-one correspondence with distinct orthogonal vectors, y_i's. W_K is a special case of weights in Hebbian learning [3]. In the hidden layer, we have the following iterative matrix equation for the prescribed data of Haar wavelet packets:

$$h_1 = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}, \cdots, h_m = \begin{bmatrix} h_{m-1} & h_{m-1} \\ -h_{m-1} & h_{m-1} \end{bmatrix}. \tag{3}$$

Therefore, $h_m^T h_m = m \cdot I_m$, where I_m is the $m \times m$ identity matrix. Here, the hidden data y_i is chosen from either a row or a column vector of h_m because $K \le m$. The block diagram of PSA is shown in Figure 1. Now, we consider the singular value decomposition of A_K in the next section.

1.1 | Singular value decomposition of A_K in PSA

For a large number of K, we can simply assume that $m \ge n$, which is obviously an overdetermined case of singular value decomposition. Therefore, $A_K^T A_K$ becomes

$$\begin{aligned} A_K^T A_K &= \frac{1}{K^2}\left(\sum_{i=1}^{K} y_i x_i^T\right)^T \left(\sum_{j=1}^{K} y_j x_j^T\right) \\ &= \frac{1}{K^2}\sum_{i=1}^{K}\sum_{j=1}^{K}(y_i^T y_j)[x_i x_j^T] \\ &= \frac{1}{K^2}\sum_{i=1}^{K}\sum_{j=1}^{K} m\delta_{ij}[x_i x_j^T] \\ &= \frac{m}{K^2}\sum_{i=1}^{K}[x_i x_i^T] \end{aligned} \tag{4}$$

where $A_K^T A_K \in \Re^{n\times n}$ is symmetric. If the rank of $A_K^T A_K$ is n, n distinct positive real eigenvalues λ_i, and the associated orthonormal eigenvectors $v_i \in \Re^{n\times 1}$, or the right singular vectors of A_K, are obtained.

$$A_K^T A_K v_i = \lambda_i v_i. \tag{5}$$

If we define the singular value $\sigma_i = \sqrt{\lambda_i}$ and the following relationship,

$$A_K v_i = \sigma_i u_i \tag{6}$$

where $u_i \in \Re^{m\times 1}$ are the left singular vectors, then from (6), we can obtain

$$\begin{aligned} &A_K[v_1|\cdots|v_n] = [u_1|\cdots|u_n]\Sigma \\ &\text{where } \Sigma = \text{diag}_{i=1\sim n}\{\sigma_i\} \in \Re^{n\times n}. \end{aligned} \tag{7}$$

Now, the singular value decomposition [16] is found by rearranging the above equation,

$$A_K V = U\Sigma \text{ or } A_K = U\Sigma V^T \tag{8}$$

where $U = [u_1|\cdots|u_n] \in \Re^{m\times n}$ and $V = [v_1|\cdots|v_n] \in \Re^{n\times n}$. The above equation (8) shows reduced singular value decomposition. Equation (4) is defined by $A^* \in \Re^{n\times n}$,

$$A^* = A_K^T A_K = \frac{m}{K^2}\sum_{i=1}^{K}[x_i x_i^T] \tag{9}$$

where A^* is a symmetric and positive semi-definite matrix, but nonsymmetric cases will be also taken into account.

On the other hand, in underdetermined cases ($m < n$), it is assumed that $A_K A_K^T$ has a full rank m,

$$A_K A_K^T u_i = \lambda_i u_i. \tag{10}$$

Similarly, the singular value is defined, and if the following relationship holds,

$$A_K^T u_i = \sigma_i v_i \tag{11}$$

it is concluded that

$$A_K^T U = V\Sigma, \text{ or } A_K = U\Sigma V^T. \tag{12}$$

Therefore, equations (8) and (12) are the same, but the sizes of U,V and Σ are different in their reduced version.

1.2 | Spectral theorem of A^* in PSA

We consider a real-value and square matrix A^* of a correlation or a covariance, which is not necessarily symmetric. From the spectral theorem [2], A^* is decomposed into projections and nilpotents. Let $P_i \in \Re^{n\times n}$ be a projection, and let $N_i \in \Re^{n\times n}$ be a nilpotent; then, A^* can be described by

$$A^* = \sum_{i=1}^{\kappa}(\lambda_i P_i + N_i), \tag{13}$$

where $\sum_{i=1}^{\kappa} m_i = n$, and λ_i is the ith eigenvalue of A^* with its multiplicity m_i, which is the root of the characteristic polynomial $\Delta_{A^*}(\lambda) = 0$. Here, we consider the multiplicities

of factors $(\lambda - \lambda_i)$'s in the characteristic equation, which becomes

$$\Delta_{A^*}(\lambda) = \prod_{i=1}^{\kappa} (\lambda - \lambda_i)^{m_i}. \tag{14}$$

The generalized eigenvalues for multiplicities can be found by a standard Jordan canonical form [16]. However, we try to obtain a projection, P_i, from the reciprocal of the characteristic equation in terms of a partial fraction expansion.

1.3 | Projection operators in PSA

In PSA, we can decompose A^* into projections and nilpotents, in accordance with the spectral theorem in (13). Let A^*, which is not symmetric, have distinct eigenvalues, and each projection operator becomes the product of an eigenvector v_i and some unit vector r_i such that

$$P_i = v_i \cdot r_i^T \tag{15}$$

and if the original covariance matrix A_K is square ($m = n$), it can be represented by a dyadic form,

$$A_K = U \Sigma V^T = \sum_{i=1}^{n} \sigma_i u_i v_i^T. \tag{16}$$

Let A^* be symmetric, then nilpotent operators become zero and it is straightforward to determine $r_i = v_i$, an orthonormal eigenvector. PSA will simply become PCA, solving for eigenvalue-eigenvector pairs. In general, if A^* is not symmetric but square, the eigenvectors are not necessarily orthogonal, but those composed of linearly independent and dependent components. The mathematical background of PSA will be presented in the next section.

2 | MATHEMATICAL BACKGROUND OF PROJECTION SPECTRAL ANALYSIS

The characteristic polynomial of A^* is represented as

$$\Delta_{A^*}(\lambda) = \det(\lambda I - A^*) = \prod_{k=1}^{\kappa} (\lambda - \lambda_k)^{m_k} \tag{17}$$

where m_k is the multiplicity order. If λ is replaced by A^*, from the Cayley-Hamilton theorem [16], it is known that $\Delta_{A^*}(A^*) = 0$. Now, a polynomial $p_i(\lambda)$ is defined as

$$p_i(\lambda) \triangleq \prod_{k=1, k\neq i}^{\kappa} (\lambda - \lambda_k)^{m_k} \tag{18}$$

and the reciprocal of $\Delta_{A^*}(\lambda)$ is given by

$$\frac{1}{\Delta_{A^*}(\lambda)} = \sum_{i=1}^{\kappa} \frac{\alpha_i(\lambda)}{(\lambda - \lambda_i)^{m_i}} \tag{19}$$

from a partial fraction expansion, and $\alpha_i(\lambda)$ is the residual equation of the order less than that of the denominator polynomial. If we multiply $\Delta_{A^*}(\lambda)$ with both sides of (19), the following identity can be found:

$$\begin{aligned} 1 &= \sum_{i=1}^{\kappa} \alpha_i(\lambda) \Delta_{A^*}(\lambda) / (\lambda - \lambda_i)^{m_i} \\ &= \sum_{i=1}^{\kappa} \alpha_i(\lambda) \prod_{k=1}^{\kappa} (\lambda - \lambda_k)^{m_k} / (\lambda - \lambda_i)^{m_i} \\ &= \sum_{i=1}^{\kappa} \alpha_i(\lambda) p_i(\lambda). \end{aligned} \tag{20}$$

Here, we claim that (20) can be extended to a matrix equation, such that

$$I = \sum_{i=1}^{\kappa} \alpha_i(A^*) p_i(A^*) \tag{21}$$

where $p_i(A^*)$ is defined as

$$p_i(A^*) \triangleq \prod_{k=1, k\neq i}^{\kappa} (A^* - \lambda_k I)^{m_k}. \tag{22}$$

Therefore, the sum product of the residue $\alpha_i(A^*)$ and the polynomial $p_i(A^*)$ becomes an identity matrix. The following lemma also holds:

Lemma 1. **[Spectral Theorem 1]** *A^* is defined as a real-value square matrix, and let a projection matrix be $P_i \in \Re^{n\times n}$,*

$$P_i = \alpha_i(A^*) p_i(A^*) = \alpha_i(A^*) \prod_{k=1, k\neq i}^{\kappa} (A^* - \lambda_k I)^{m_k}. \tag{23}$$

Then, the spectral theorem (13) is satisfied for any real-value square matrix A^*,

$$A^* = \sum_{i=1}^{\kappa} (\lambda_i P_i + N_i) \tag{24}$$

with $N_i = N_i(A^*) = P_i(A^* - \lambda_i I)$, and (21) becomes

$$I = \sum_{i=1}^{\kappa} P_i = \sum_{i=1}^{\kappa} \alpha_i(A^*) p_i(A^*). \tag{25}$$

Proof. A^* can be represented as follows, by post-multiplying A^* with both sides of (21):

$$\begin{aligned} I \cdot A^* &= \sum_{i=1}^{\kappa} \alpha_i(A^*) p_i(A^*) \cdot A^* \\ &= \sum_{i=1}^{\kappa} \alpha_i(A^*) p_i(A^*) \cdot (\lambda_i I + A^* - \lambda_i I) \\ &= \sum_{i=1}^{\kappa} (\lambda_i P_i + N_i), \end{aligned} \tag{26}$$

$$N_i = P_i(A^* - \lambda_i I) = \alpha_i(A^*) p_i(A^*)(A^* - \lambda_i I), \tag{27}$$

because $P_i = \alpha_i(A^*) p_i(A^*)$. Now, equation (24) is satisfied for any square matrix A^*. □

Remark. If there is a multiplicity ($m_i \geq 2$), the fact that $N_i \neq 0$ is obvious because of the linear dependencies in m_i eigenvectors of λ_i.

Lemma 2. [Spectral Theorem 2] *If a square matrix A^* is real and symmetric, all nilpotents become zeros, $N_i = 0$, i. Now, we know that (24) simply becomes*

$$A^* = \sum_{i=1}^{\kappa} \lambda_i P_i. \tag{28}$$

Here, the set of all P_i's is called the resolution of identity, which satisfies (25).

Proof. It is easily found that N_i is also symmetric because $N_i = P_i(A^* - \lambda_i I)$. For any vectors x and y, $x^T N_i y = (N_i x)^T y$ is satisfied.

Now, suppose that $N_i^{m_i-1} \neq 0$, then $N_i^{m_i} = 0$, from the property of nilpotency. The squared norm of $N_i^{m_i-1}x$ becomes

$$\begin{aligned} \|N_i^{m_1-1}x\|^2 &= (N_i^{m_i-1}x)^T(N_i^{m_i-1}x) \\ &= (N_i^{m_i}x)^T(N_i^{m_i-2}x) = 0. \end{aligned} \tag{29}$$

Here, equation (29) contradicts the assumption that $N_i^{m_i-1} \neq 0$. Therefore, $N_i = 0$ is the only solution, and the spectral theorem states that $A^* = \sum_{i=1}^{\kappa} \lambda_i P_i$.

□

Remark. It is well-known that the eigenvalues are real and distinct for a symmetric matrix A^*, and the associated eigenvectors are orthonormal.

Theorem 3. [Spectral Theorem 3] *For any nonlinear function $f(\cdot)$, and for a symmetric matrix A^* that satisfies (28), $f(A^*)$ may be rewritten by the spectral decomposition theorem as follows:*

$$f(A^*) = \sum_{i=1}^{\kappa} f(\lambda_i) P_i. \tag{30}$$

λ_i is the ith real and distinct eigenvalue of A^*. $\mathrm{P_i}$ is described by $P_i = q_i q_i^T$, where $\mathrm{q_i}$ is the ith orthonormal eigenvector of A^* ($A^* v_i = \lambda_i v_i$).

Proof. The proof is given in [1].

□

Now, let us consider a generalized projection theorem that deals with nonsymmetric projection operators.

Theorem 4. [Projection Theorem] *If A^* is non-symmetric with distinct eigenvalues, and if P_i is composed of the product between a column vector $v_i \in \Re^{n\times 1}$ and a row vector $r_i^T \in \Re^{1\times n}$, such as*

$$P_i = v_i \cdot r_i^T \in \Re^{n\times n}. \tag{31}$$

then, $\mathrm{v_i}$ is the first principal component from the orthogonalization of $\mathrm{P_i}$ with $\|\mathrm{v_i}\|_2 = 1$. Here, the only non-zero eigenvalue μ of $\mathrm{P_i}$ is 1, and $r_i^T v_i = 1$. All $\mathrm{v_i}$'s are independent eigenvectors of A^*. Now, let $\mathrm{v_i}$ be the ith column of V, then $A^* \cdot V = V \cdot \Lambda$.

Proof. Refer to [1] for the full proof. □

Whenever some eigenvalues are negligible and approximated to zero, the rank of A^* can be reduced by using the following theorem:

Theorem 5. [Reduction Theorem] *Let a nonsymmetric matrix be A^*, with arbitrary eigenvalues. We define V, R, $\bar{V}$, and $\bar{R}$ as follows:*

$$\begin{aligned} V &= [v_1|\cdots|v_\kappa] \in \Re^{n\times\kappa},\ R = \begin{bmatrix} r_1^T \\ \cdots \\ r_\kappa^T \end{bmatrix} \in \Re^{\kappa\times n}, \\ \bar{V} &= [\bar{v}_1|\cdots|\bar{v}_\ell] \in \Re^{n\times\ell}, \bar{R} = \begin{bmatrix} \bar{r}_1^T \\ \cdots \\ \bar{r}_\ell^T \end{bmatrix} \in \Re^{\ell\times n} \end{aligned} \tag{32}$$

where $\ell = n - \kappa$. Then, from Lemma 1 and Theorem 4, A^* becomes

$$A^* = \underbrace{V \cdot \Lambda \cdot R}_{\text{lin. indep.}} + \underbrace{\bar{V} \cdot \bar{\Lambda} \cdot \bar{R} + N}_{\text{lin. depend.}} \ (R \cdot V = I_\kappa). \tag{33}$$

Here, $N = \sum_{i=1}^{\kappa} N_i$, and V is a transformation operator of the independent eigenvectors. The diagonal elements of $\bar{\Lambda}$ are the eigenvalues of multiplicities. If $\lambda_\kappa \cong 0$ with multiplicity ($\mathrm{m_\kappa}$), and if the other eigenvalues are distinct, then equation (33) becomes

$$\begin{aligned} A^* &\cong V \cdot \Lambda \cdot R + N_\kappa \\ &\cong V_r \cdot \Lambda_r \cdot R_r. \end{aligned} \quad (R_r \cdot V_r = I_r) \tag{34}$$

$\mathrm{N_\kappa}$ is a nilpotent matrix, corresponding to multiple $\lambda_\kappa = 0$. The reduced eigenvalue matrix $\Lambda_r \in \Re^{(\kappa-1)\times(\kappa-1)}$ may contain complex-conjugate eigenvalues, in which the κth row and column elements of Λ are eliminated. Similarly, $V_r \in \Re^{n\times(\kappa-1)}$ is the reduced V with the κth column eliminated, while $R_r \in \Re^{(\kappa-1)\times n}$ is a reduced version of R in which the κth row is eliminated. For symmetric A^*, it is easy to prove that $R_r = V_r^T$.

Proof. The proof is provided in [1]. □

Remark. It is well-known that $\ell = n - \kappa$ is the number of pure extra multiplicities. In other words, the number of independent components, κ, is subtracted from the number of total multiplicities. Therefore, the first term in (33) is composed of the linearly independent components of A^*, while the second term is composed of the linearly dependent components from the multiplicities or from the zero eigenvalues. Finally, the third term N is the sum of nilpotents from the multiplicities.

It will be discussed how PSA is a useful machine learning tool for encoding training data and decoding test data.

3 | LEARNING AND TESTING IN PROJECTION SPECTRAL ANALYSIS

PSA is described as an extended class of PCA or ICA in unsupervised learning, because the geometric structure of the eigenvectors is not necessarily orthogonal, and if not, A^* is divided into linearly independent and dependent components. Here, we consider general instances of correlation or covariance matrices. If $m \geq n$, $A^* = A_K^T A_K$, and the rank is determined by $\kappa = \min(n, K)$, whereas, if $m < n$, $A^* = A_K A_K^T$, and the rank is $\kappa = \min(m, K)$. This rank κ is further reduced by using the reduction Theorem 2 or by sorting the eigenvalues λ_i (the singular values σ_i) such that

$$\begin{aligned} &\lambda_1 \geq \cdots \geq \lambda_r > 0,\ \lambda_{r+1} \sim \lambda_n \cong 0 \\ &(\sigma_1 \geq \cdots \geq \sigma_r > 0,\ \sigma_{r+1} \sim \sigma_n \cong 0) \end{aligned} \tag{35}$$

where $r \leq \kappa$. In these rank-deficient cases, $A^* v_i = \lambda_i v_i$ $(i = 1 \sim r)$ and the reduced left singular vectors are found by $u_i = A_K v_i/\sigma_i$ if $m \geq n$, while $A^{*T} u_i = \lambda_i u_i$ $(i = 1 \sim r)$ and the reduced right singular vectors are obtained by $v_i = A_K^T u_i/\sigma_i$ if $m < n$. Therefore, A_K is represented by

$$\begin{aligned} A_K &= U_r \cdot \Sigma_r \cdot V_r^T, \\ U_r &= [u_1|u_2|\cdots|u_r] \in \Re^{m\times r}, \\ V_r &= [v_1|v_2|\cdots|v_r] \in \Re^{n\times r}. \end{aligned} \tag{36}$$

Now, the weight vectors ω_k or the weight matrix Ω are computed by

$$\begin{aligned} \omega_k &= V_r^T \cdot x_k \in \Re^{r\times 1}\ (k = 1 \sim K), \\ \Omega &= V_r^T \cdot X \in \Re^{r\times K}. \end{aligned} \tag{37}$$

The learning procedure of PSA is represented as follows:

Learning (Encoding) in PSA

1. Provide input-hidden-output data triples x_k, y_k, and z_k $(k = 1 \sim K)$, and find the weights in the hidden and the output layers, as follows:
$$\begin{aligned} A_k &= ((k-1)/k)A_{k-1} + (1/k)y_k x_k^T, \\ W_k &= ((k-1)/k)W_{k-1} + (1/k)z_k y_k^T. \end{aligned} \tag{38}$$
2. Use (19), (23), and (27) to find λ_i, P_i, and N_i, respectively, from the spectral theorem in (24), i.e.,
$$\begin{aligned} P_i &= \alpha_i(A^*) \prod_{k=1,k\neq i}^{L} (A^* - \lambda_k I)^{m_k}, \\ N_i &= P_i(A^* - \lambda_i I). \end{aligned}$$
3. Obtain v_i and r_i by solving $P_i = v_i \cdot r_i^T$ from the qr-decomposition and the reduction theorem (34) or from the reduced singular value decomposition (36) to obtain
$$A^* = V_r \cdot \Lambda_r \cdot R_r, \quad A_K = U_r \cdot \Sigma_r \cdot V_r^T.$$
4. Compute the weights $\{\omega_k\}$ in (37), and store Ω.

Therefore, W_K, the Haar wavelets y_i's, the eigen-structures U_r, Σ_r, V_r (V_r, Λ_r, R_r), and the weights ω_k's are stored in the memory after the encoding procedure. Now, the test procedure for PSA is represented as follows:

Testing (Decoding) in PSA

1. First, provide test input data $x \in \Re^n$
2. Obtain test weights, $\omega = V_r^T x \in \Re^{K\times 1}$
3. For $k = 1 \sim K$,
   - From the stored weights, ω_k, compute the Euclidean distance ρ_k,
$$\rho_k = \|\omega - \omega_k\|_2. \tag{39}$$
4. Find an index k^* of the minimum distance in $\{\rho_k\}$ and compute the output data z by using the weights in (2),
$$z = W_K \cdot y_{k^*} \tag{40}$$
5. Use softmax to obtain the probability values of z.

The Euclidean distance of ρ_k is the same as that of $\|x - x_k\|_2$ if the eigenvalues of A^* are distinct, since

$$\begin{aligned} \|x - x_k\|_2^2 &= (\omega - \omega_k)^T V_r^T R_r^T (\omega - \omega_k) \\ &= \|\omega - \omega_k\|_2^2 = \rho_k. \end{aligned} \tag{41}$$

Remark. Now, let A^* have distinct eigenvalues with multiple zero eigenvalues, then $R_r V_r = I_r$ and it is possible to search for the minimum index in a lower dimension.

4 | INCREMENTAL LEARNING OF PROJECTION SPECTRAL ANALYSIS

It is known that one drawback of learning in most neural networks, such as deep belief networks, convolutional neural networks, and backpropagation networks, is that, whenever it is necessary to add new training data, we have to perform the learning procedure again from the beginning of the old training data. This is the main reason for incremental learning of neural networks. We propose two methods of PSA incremental learning in this paper. The former is incremental learning for a single incoming data, while the latter is a similar incremental learning scheme for two training data sets with arbitrary numbers of training data.

4.1 | Incremental learning for a single incoming data

When updating the weights A_K by using a single incoming data, we start with equation (38),

$$(k+1)A_{k+1} = kA_k + B,\ B = y_{k+1}x_{k+1}^T. \tag{42}$$

If we consider an overdetermined case ($m \geq n$), the singular value decompositions proceed as follows:

$$A_{k+1} = U\Sigma V^T, A_k V_1 = U_1\Sigma_1, BV_2 = U_2\Sigma_2. \tag{43}$$

Let us define $U_2 = U_1 P$ and $V_2 = V_1 Q$, where P and Q are appropriate coordinate transformation matrices. If we substitute U_2,V_2 for $BV_2 = U_2\Sigma_2$, then

$$\begin{aligned} BV_1 Q = U_1 P\Sigma_2 &\rightarrow BV_1 = U_1 P\Sigma_2 Q^T \\ &\rightarrow B = U_1 P\Sigma_2 Q^T V_1^T. \end{aligned} \tag{44}$$

Equation (42) now becomes

$$\begin{aligned} (k+1)A_{k+1} &= kU_1\Sigma_1 V_1^T + U_1 P\Sigma_2 Q^T V_1^T \\ &= U_1\{k\Sigma_1 + P\Sigma_2 Q^T\}V_1^T. \end{aligned} \tag{45}$$

The next step is to find the singular value decomposition of $(k\Sigma_1 + P\Sigma_2 Q^T)/(k+1)$ and rearrange (45) such that

$$\begin{aligned} (k\Sigma_1 + P\Sigma_2 Q^T)/(k+1) &= U_3\Sigma_3 V_3^T \\ A_{k+1} &= U_1 U_3\Sigma_3 V_3^T V_1^T. \end{aligned} \tag{46}$$

Now, comparing the above equation with $A_{k+1} = U\Sigma V^T$, the followings are obtained:

$$\begin{aligned} U = U_1 U_3,\ \Sigma = \Sigma_3, V = V_1 V_3 \\ \text{where } P = U_1^T U_2,\ Q = V_1^T V_2. \end{aligned} \tag{47}$$

It is noted that $P \in \Re^{m\times m}$ and $Q \in \Re^{n\times n}$ are generalized rotational transformation matrices since their column vectors are orthonormal to each other. In other representations, $P^T P = U_2^T U_1 U_1^T U_2 = I_m$ and $Q^T Q = V_2^T V_1\ V_1^T V_2 = I_n$. Therefore, by using (46) and (47), the singular value triples $\{U, \Sigma, V\}$ of $A_{k+1}(A_{k+1} = U\Sigma V^T)$ are found.

4.2 | Incremental learning for combining two incoming data sets

The next strategy is to combine two sets of training data and to update the weights when the number of the next data set B_{K_2} is different from that of the first set A_{K_1}. We now assume two sets of correlation or covariance matrices, A_{K_1} and B_{K_2}, the combined one $C_K(K = K_1 + K_2)$, and an interior division of A_{K_1} and B_{K_2}, as follows:

$$\begin{aligned} A_{K_1} = \frac{1}{K1}\sum_{i=1}^{K_1} y_i x_i^T,\ B_{K_2} = \frac{1}{K_2}\sum_{j=K_1+1}^{K_1+K_2} y_j x_j^T, \\ C_K = \frac{1}{K}\sum_{k=1}^{K} y_k x_k^T = \frac{1}{K}(K_1 A_{K_1} + K_2 B_{K_2}). \end{aligned} \tag{48}$$

Let the three singular value decompositions be $A_{K_1} = U_1\Sigma_1 V_1^T, B_{K_2} = U_2\Sigma_2 V_2^T$, and $C_K = U\Sigma V^T$. Similarly, P and Q are defined as $U_2 = U_1 P$ and $V_2 = V_1 Q$, respectively, which denote the generalized rotational transformation matrices. From (49),

$$\begin{aligned} K_1 A_{K_1} + K_2 B_{K_2} &= K_1 U_1\Sigma_1 V_1^T + K_2 U_2\Sigma_2 V_2^T \\ &= K_1 U_1\Sigma_1 V_1^T + K_2 U_1 P\Sigma_2 Q^T V_1^T \\ &= U_1(K_1\Sigma_1 + K_2 P\Sigma_2 Q^T)V_1^T \\ &= KC_K. \end{aligned} \tag{49}$$

Now, $(K_1\Sigma_1 + K_2 P\Sigma_2 Q^T)/K$ may be further applied to the singular value decomposition since $P\Sigma_2 Q^T$ is not diagonal in general, as follows:

$$\begin{aligned} (K_1\Sigma_1 + K_2 P\Sigma_2 Q^T)/K = U_3\Sigma_3 V_3^T, \\ C_K = U\Sigma V^T = U_1 U_3\Sigma_3 V_3^T V_1^T. \end{aligned} \tag{50}$$

Therefore, our new singular value triples are obtained,

$$\begin{aligned} U = U_1 U_3, \Sigma = \Sigma_3, V = V_1 V_3 \\ \text{where } P = U_1^T U_2, Q = V_1^T V_2 \end{aligned} \tag{51}$$

and it is possible to update the weights of PSA by using (50) and (51), even if the number of incoming training data is different from that of the existing one already stored.

5 | SIMULATION RESULTS OF BENCHMARK DBs

The experiments on PSA have also been conducted in this section, where benchmark DBs such as MNIST and CIFAR-10 are used to evaluate the performance of PSA, to which the reduction theorem and the incremental learning algorithm are applied. Here, a design parameter, the tolerance of zero eigenvalues, in the reduction procedure, is defined as ε, for example, $\varepsilon = 10^{-5}$. Therefore, each reduction eliminates the corresponding column vector in U_r (under-determined) or V_r(over-determined) in (36), when one diagonal element of Σ_r is reduced. The associated V_r(under-determined) or U_r(over-determined) is obtained by (6) or (11), respectively. With incremental learning from (50) and (51), the size of each partition for additional training data is selected as $K_2 = 1{,}000 \sim 10{,}000$ per evaluation. Therefore, 6 or 5 incremental procedures are required for training MNIST or CIFAR-10 data, respectively, and it is not necessary to recompute from the starting point, as there are more incoming data for learning.

Table 1 shows the recognition rates and standard deviations of PSA for MNIST and CIFAR-10 DBs. In the first row, the number of training data starts from 6,000 and 5,000 for MNIST and CIFAR-10, respectively, and the size of training data increases by 5,000 in both cases, up to the full size of 60,000 and 50,000, respectively. The ratio 5:1 holds for the sizes of training and test data. In each instance of experiments with 100 trials, test data are randomly chosen from the test batch, with size 10,000 in both cases.

Graphical representations of the experimental results of test data are shown in Figures 2 and 3, where the best recognition rates are 96.9% for MNIST and 35.4% for CIFAR-10, respectively. As shown in Figures 2 and 3, the mean values from test results of MNIST tend to be saturated asymptotically towards a limit, while those values are linearly increasing from test results of CIFAR-10, which means that the performance of PSA could be improved if more training data are available.

6 | CONCLUSIONS AND DISCUSSION

PSA is refined and upgraded in this paper by using the singular value decomposition of arbitrary input-output data with the prescribed hidden data, and the spectral theorems with the mathematical background of the eigenstructures, projections, and nilpotents. Even when it is applied to nonsymmetric correlation or covariance matrices, PSA is useful and effective with numerical stability and robustness. For symmetric matrices, PSA is equivalent to PCA, while it becomes equivalent to ICA for nonsymmetric cases. Therefore, PSA is a unified approach to both PCA and ICA.

Recent works on ICA include causal analysis in a structural equation model [17], group ICA for three-way data [18], and improved estimation of basic linear mixing [19]. When a nonsymmetric covariance is encountered, we can decompose A^* into geometrically meaningful components, the linearly independent component, and the other linearly dependent ones. If the eigenvalues are distinct except for multiple zero eigenvalues, the rank is further reduced by excluding the multiple zero eigenvalues.

Moreover, two methods of incremental learning are also investigated in order to subsequently continue to update the weights of PSA without losing the previously stored information. This incremental learning is also applied to the experiments of benchmark DBs, such as MNIST and CIFAR-10. Therefore, PSA is a useful tool of neural networks, and it may be not only combined with

TABLE 1 PSA test results: means and standard deviations in MNIST vs. CIFAR-10

MNIST			CIFAR-10		
#Train : #Test	**Mean**	**SD**	**#Train : #Test**	**Mean**	**SD**
6,000 : 1,000	93.79%	0.686%	5,000 : 1,000	28.35%	1.017%
12,000 : 2,000	94.99%	0.431%	10,000 : 2,000	29.36%	0.726%
18,000 : 3,000	95.47%	0.416%	15,000 : 3,000	30.17%	0.611%
24,000 : 4,000	95.87%	0.253%	20,000 : 4,000	30.99%	0.541%
30,000 : 5,000	96.25%	0.192%	25,000 : 5,000	31.68%	0.426%
36,000 : 6,000	96.41%	0.133%	30,000 : 6,000	32.50%	0.347%
42,000 : 7,000	96.74%	0.131%	35,000 : 7,000	33.14%	0.317%
48,000 : 8,000	96.75%	0.103%	40,000 : 8,000	33.91%	0.286%
54,000 : 9,000	96.77%	0.048%	45,000 : 9,000	34.67%	0.206%
60,000 : 10,000	96.91%	0.000%	50,000 : 10,000	35.40%	0.000%

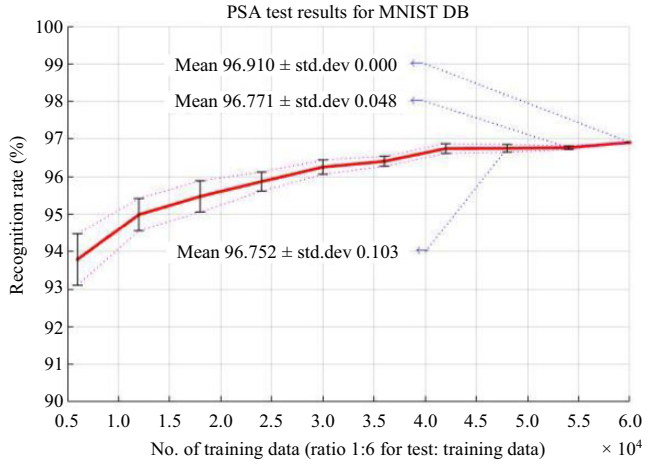

FIGURE 2 Simulation results of MNIST; means and standard deviations

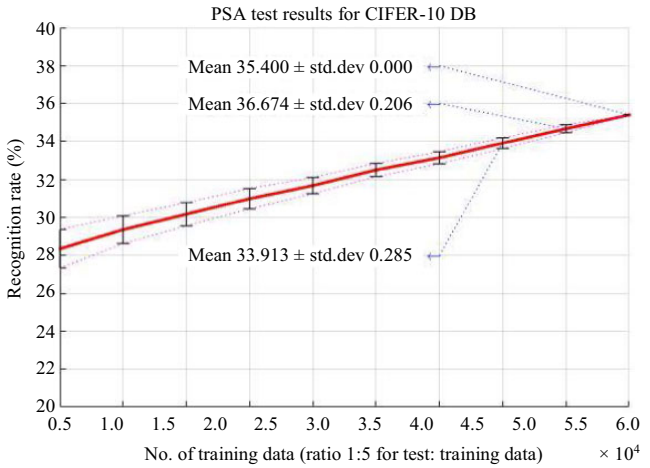

FIGURE 3 Simulation results of CIFAR-10; means and standard deviations

convolutional neural networks, but it could also be used for a number of practical applications, such as image detection, tracking, and recognition.

REFERENCES

1. H. Kang, *Projection spectral analysis*, Int. J. Control, Autom. Syst. **13** (2015), no. 6, 1530–1537.
2. A. W. Naylor and G. R. Sell, *Linear operator theory in engineering and science - Applied mathematical sciences*, vol. 40 Springer-Verlag, Inc., New York, 1982.
3. D. Hebb, *Organization of behavior*, Science Edition, Inc., New York, 1961.
4. J. J. Hopfield, *Neural networks and physical systems with emergent collective computational abilities*, Proc. Natl Acad. Sci. USA, Biophys. **79** (1982), 2554–2558.
5. B. Kosko, *Bidirectional associative memories*, IEEE Trans. System, Man, Cybern. **18** (1988), no. 1, 49–60.
6. M. Turk and A. Pentland, *Eigenfaces for recognition*, J. Cogn. Neurosci. **3** (1991), no. 1, 71–86.
7. A. Hyvärinen, J. Karhunen, and E. Oja, *Independent component analysis*, John Wiley and Sons, Inc., New York, 2001.
8. J. V. Stone, *Independent component analysis - A tutorial introduction*, The MIT Press, Cambridge, MA, 2004.
9. A. Hyvärinen, *Fast and robust fixed-point algorithms for independent component analysis*, IEEE Trans. Neural Netw. **10** (1999), no. 3, 626–634.
10. H. Kang, *Multilayered associative neural networks (m. a. n. n.): Storage capacity vs. perfect recall*, IEEE Trans. Neural Networks **5** (1994), no. 5, 812–822.
11. G. E. Hinton, S. Osindero, and Y. W. Teh, *A fast learning algorithm for deep belief nets*, Neural Comput. **18** (2006), no. 7 1527–1554.
12. Y. Bengio, *Learning deep architecture for AI*, Found. Trends Mach. Learn. **2** (2009), no. 1, 1–127.
13. Y. LeCunn, et al., *Gradient-based learning applied to document recognition*, Proc. IEEE **86** (1998), 1–46.
14. Y. LeCunn, K. Kavukcuoglu, and C. Farabet, Convolutional networks and applications in vision, *Proc. IEEE Int. Symp. Circuits Syst.*, Paris, France, 2010, pp. 253–256.
15. H. Kang, Associative cubes in unsupervised learning for robust gray-scale image recognition, *Proc. 3rd Int. Symposium on Neural Networks, Advances in Neural Networks - ISNN 2006*, Springer-Verlag, Berlin Heidelberg (J. Wang et al., ed.), vol. LNCS 3972, (2006), pp. 581–588.
16. C.-T. Chen, *Linear system theory and design*, Oxford University Press, Inc., New York, 1999.
17. S. Shimizu, et al., *A linear non-Gaussian acyclic model for causal discovery*, J. Mach. Learn. Res. **7** (2006), 2003–2020.
18. V. Calhoun, et al., *A method for making group inferences from functional MRI data using independent component analysis*, Hum. Brain Mapp. **14** (2001), 140–151.
19. M. Gutmann and A. Hyvärinen, Noise-constrastive estimation: A new estimation principle for unnormalized statistical models, *Proc. Int. Conf. Artif. Intell. Statistics* Sardinia, Italy, May 13–15, 2010, pp. 297–304.

AUTHOR BIOGRAPHIES

Hoon Kang was born in Seoul, Rep. of Korea, in 1959. He received his BS and MS degrees in electronic engineering from Seoul National University, Rep. of Korea, in 1982 and 1984, respectively. He earned his PhD degree and the CIMS certificate at the School of Electrical Engineering at Georgia Institute of Technology, Atlanta, USA, in 1989. From 1989 to 1991, he was first a postdoctoral fellow and then a research associate in the Georgia Tech Electrical Engineering Department. As he participated in a number of projects sponsored by the National Science Foundation, the Office of Naval Research, Ford

Motor Company, and Honeywell Inc., he developed new research ideas on fuzzy logic control, intelligent robotic control, and fault detection and identification. He also joined Automation Concepts and Systems, Inc., as a research fellow in 1991. He joined the School of Electrical and Electronics Engineering at Chung-Ang University, Seoul, Rep. of Korea in 1992, and became a full professor in 2000. He served as the department chair and the committee members of the Korean academic societies such as IIS, ICROS, and IEIE, where he served as an editorial board member, a financial secretary, and a general affairs director. His research interests include computational intelligence, such as fuzzy systems; neural networks; evolutionary computation; artificial life; and robotics and robot vision such as visual tracking, object recognition, human computer interfaces, intelligent robots, and humanoids.

Hyun Su Lee received his BS degree from the School of Electrical and Electronics Engineering, Chung-Ang University, Seoul, Rep. of Korea, in 2014. He is currently working toward an MS degree in intelligent robot at Chung-Ang University. Now, he joined Intelligent robot and vision lab, working on image detection and neural networks. His research interests are artificial intelligence, intelligent control, robot vision, and image processing.