

An Incremental Principal Component Analysis for Chunk Data

Seiichi Ozawa, *Member, IEEE*, Shaoning Pang, *Sen. Member, IEEE*,
and Nikola Kasabov, *Sen. Member, IEEE*

Abstract—This paper presents a new algorithm of dynamic feature selection by extending the algorithm of Incremental Principal Component Analysis (IPCA), which has been originally proposed by Hall and Martin. In the proposed IPCA, a chunk of training samples can be processed at a time to update the eigenspace of a classification model without keeping all the training samples given so far. Under the assumption that L of training samples are given in a chunk, first we derive a new eigenproblem whose solution gives us a rotation matrix of eigen-axes, then we introduce a new algorithm of augmenting eigen-axes based on the accumulation ratio. We also derive the one-pass incremental update formula for the accumulation ratio. The experiments are carried out to verify if the proposed IPCA works well. Our experimental results demonstrate that it works well independent of the size of data chunk, and that the eigenvectors for major components are obtained without serious approximation errors at the final learning stage. In addition, it is shown that the proposed IPCA can maintain the designated accumulation ratio by augmenting new eigen-axes properly. This property enables a learning system to construct an informative eigenspace with minimum dimensionality.

I. INTRODUCTION

Recently, the learning under dynamic environments, where the property of data source is being changed over time, has received a great attention in the computational intelligence community [1]-[4]. In realistic situations, it is not always assumed that a complete set of training samples are given in a batch to learn a system [5], [6]. Therefore, a system is required to test and improve the performance automatically on a on-going basis. This type of learning is often called *on-line learning* or *incremental learning*, and there have been proposed numerous works on this topic so far [7]-[11].

In pattern recognition and data mining, input data are often composed of many attributes. Hence, the informative input variables are first selected before the classification is carried out. This means that when constructing an adaptive classification system, we should consider not only the incremental learning of classifiers but also that of feature spaces. For this purpose, several incremental algorithms have been independently developed for the feature selection and the classifier learning. As for the feature selection, Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) have been extended to an incremental version [12]-[17]. Hall and Martin have developed a smart method

to update eigenvectors and eigenvalues incrementally (i.e., the update of an eigenspace) called *Incremental Principal Component Analysis* (IPCA) [15]. And Ozawa et al. have extended this IPCA algorithm such that an eigen-axis is augmented based on the accumulation ratio in order to control the dimensionality of an eigenspace easily [18].

Recently, we have proposed a new scheme of incremental learning in which feature selection and classifier learning are simultaneously carried out on-line [17]-[20]. In our previous works, IPCA or ILDA are adopted as feature selection methods, and Evolving Clustering Method [21] or Resource Allocating Network with Long-Term Memory [20], [22] are adopted as classifiers. A distinctive feature of the proposed scheme is that the learning is conducted incrementally *one-pass*; here, one-pass means that training samples are passed through a system only once for learning purposes [21], [23]. It was verified that the classification accuracy was improved constantly even if only an insufficient number of training samples are provided initially [19], [20]. However, several problems still remain for this approach. The biggest problem is scalability. In our previous approach, a training sample must be learned one by one even if a chunk of training sample is provided at a time. This causes inefficiency in computations because eigenvalue decomposition in IPCA must be applied to each training sample.

In this paper, the IPCA algorithm is extended such that a chunk of training samples is trained at a time, then we propose an effective way to determine a set of new eigen-axes to be augmented for a chunk of training samples. In addition, we introduce the accumulation ratio as a criterion of the eigen-axis augmentation and derive its incremental update equation.

This paper is organized as follows. Section II gives a quick review on IPCA and its extended method. In Section III, we propose an extension of IPCA in which a chunk of training samples is learned at a time and new eigen-axes are efficiently selected based on the accumulation ratio. In Section IV, the proposed IPCA is evaluated with three standard datasets from the UCI Machine Learning Repository [24]. Finally, Section V summarizes this paper and addresses our future works.

II. INCREMENTAL PRINCIPAL COMPONENT ANALYSIS FOR A SINGLE TRAINING SAMPLE

A. Original Algorithm

Let us review the IPCA algorithm proposed by Hall and Martin [15] briefly.

Seiichi Ozawa is with Graduate School of Science and Technology, Kobe University, Kobe 657-8501, JAPAN (email: ozawasei@kobe-u.ac.jp).

Shaoning Pang and Nikola Kasabov are with the Knowledge Engineering and Discovery Research Institute, Auckland University of Technology, 581 Great South Rd, 1020 Auckland, New Zealand (email: spang@aut.ac.nz, nkasabov@aut.ac.nz).

Assume that N training samples $\mathbf{x}_i \in \mathcal{R}^n$ ($i = 1, \dots, N$) have been presented so far, and an eigenspace model $\Omega = (\bar{\mathbf{x}}, \mathbf{U}_k, \mathbf{\Lambda}_k, N)$ is constructed by calculating the eigenvectors and eigenvalues from the covariance matrix of \mathbf{x}_i , where $\bar{\mathbf{x}}$ is a mean vector of \mathbf{x}_i ($i = 1, \dots, N$), \mathbf{U}_k is an $n \times k$ matrix whose column vectors correspond to the eigenvectors, and $\mathbf{\Lambda}_k$ is a $k \times k$ matrix whose diagonal elements correspond to the eigenvalues. Here, k is the number of eigen-axes spanning the eigenspace (i.e., eigenspace dimensionality).

Now, assume that the $(N + 1)$ th training sample $\mathbf{y} \in \mathcal{R}^n$ is given. The addition of this new sample will lead to the changes in both mean vector and covariance matrix; therefore, the eigenvectors and eigenvalues should also be updated. The new mean input vector $\bar{\mathbf{x}}'$ is easily obtained as follows:

$$\bar{\mathbf{x}}' = \frac{1}{N+1}(N\bar{\mathbf{x}} + \mathbf{y}) \in \mathcal{R}^n. \quad (1)$$

The problem is how to update the eigenvectors and eigenvalues.

When updating the eigenspace model Ω , we need to check if the eigenspace should be enlarged in term of dimensionality. If the new sample includes almost all energy in the current eigenspace, the dimensionality does not need to be changed. However, if the eigenspace includes certain energy in the complementary eigenspace, the dimensional augmentation is needed, or crucial information on the new sample might be lost. In the original IPCA, the judgment of the eigenspace augmentation is made based on the norm of the following residue vector $\mathbf{h} \in \mathcal{R}^n$:

$$\mathbf{h} = (\mathbf{y} - \bar{\mathbf{x}}) - \mathbf{U}_k \mathbf{g} \quad (2)$$

where

$$\mathbf{g} = \mathbf{U}_k^T (\mathbf{y} - \bar{\mathbf{x}}). \quad (3)$$

Here, T means the transposition of vectors and matrices. When the norm of the residue vector \mathbf{h} is larger than a threshold value η , the dimensionality of the current eigenspace is increased from k to $k + 1$, and a new eigen-axis is added in the direction of \mathbf{h} . Otherwise, the dimensionality of the eigenspace remains the same.

It has been shown that the eigenvectors and eigenvalues are updated by solving the following intermediate eigenproblem [15]:

$$\left(\frac{N}{N+1} \begin{bmatrix} \mathbf{\Lambda}_k & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} + \frac{N}{(N+1)^2} \begin{bmatrix} \mathbf{g}\mathbf{g}^T & \gamma\mathbf{g} \\ \gamma\mathbf{g}^T & \gamma^2 \end{bmatrix} \right) \mathbf{R} = \mathbf{R}\mathbf{\Lambda}'_{k+1} \quad (4)$$

where $\gamma = \tilde{\mathbf{h}}^T (\mathbf{y} - \bar{\mathbf{x}})$, \mathbf{R} is a $(k+1) \times (k+1)$ matrix whose column vectors correspond to the eigenvectors obtained from the above intermediate eigenproblem, $\mathbf{\Lambda}'_{k+1}$ is the new eigenvalue matrix, and $\mathbf{0}$ is a k -dimensional zero vector. Using the solution \mathbf{R} , the new $n \times (k+1)$ eigenvector matrix \mathbf{U}'_{k+1} is calculated as follows:

$$\mathbf{U}'_{k+1} = [\mathbf{U}_k, \hat{\mathbf{h}}]\mathbf{R} \quad (5)$$

where

$$\hat{\mathbf{h}} = \begin{cases} \mathbf{h}/\|\mathbf{h}\| & \text{if } \|\mathbf{h}\| > \eta \\ \mathbf{0} & \text{otherwise.} \end{cases} \quad (6)$$

Here, η is a small threshold value which is set to zero in the original IPCA [15]. From Eq. (5), intuitively we can consider that \mathbf{R} gives a rotation from old eigen-axes to new ones; hence, let us call \mathbf{R} *rotation matrix* here. Note that if $\hat{\mathbf{h}} = \mathbf{0}$ (i.e., the case that the dimensional augmentation was not needed), \mathbf{R} degenerates into an $n \times k$ matrix, and \mathbf{U}'_{k+1} and $\mathbf{\Lambda}'_{k+1}$ also degenerate to the $n \times k$ matrix and the $k \times k$ matrix, respectively.

B. A Criterion for Eigen-axes Augmentation

As seen from Eq. (6), a new eigen-axis is augmented whenever the norm of a residue vector is larger than a threshold value η in the original IPCA. However, this is not a good criterion in practice because a suitable threshold can be varied depending on the magnitude of input values. If the threshold is too small, the dimensionality of a feature space could be excessively large and an efficient feature space with small dimensions is hard to be constructed; this may deteriorate both generalization performance and computational efficiency. On the other hand, if the threshold is too large, essential information on training samples is lost unexpectedly.

To reduce the dependency of the threshold on input values, the following accumulation ratio is often used as a criterion:

$$A(\mathbf{U}_k) = \frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^n \lambda_i} \quad (7)$$

where $\mathbf{U}_k = \{\mathbf{u}_1, \dots, \mathbf{u}_k\}$ is the eigenvector matrix whose column vectors span the k -dimensional feature space, λ_i ($i = 1, \dots, k$) is the eigenvalue of \mathbf{u}_i , and n is the dimensionality of the input space, respectively. By specifying an appropriate threshold value θ , the feature space dimensions are automatically determined by searching for a minimum k such that $A(\mathbf{U}_k) > \theta$ holds. In general, the update of Eq. (7) cannot be done without the training samples given previously. This is a serious problem when a one-pass incremental learning algorithm is developed. To overcome this problem, we need an incremental update algorithm of $A(\mathbf{U}_k)$ without keeping all the past training samples.

In [18], we derived the incremental update equation for the accumulation ratio $A'(\mathbf{U}_k)$ in Eq. (7), and it is given by

$$A'(\mathbf{U}_k) = \frac{N(N+1) \sum_{i=1}^k \lambda_i + N \|\mathbf{U}_k^T (\mathbf{y} - \bar{\mathbf{x}})\|^2}{N(N+1) \sum_{i=1}^n \lambda_i + N \|\mathbf{y} - \bar{\mathbf{x}}\|^2} \quad (8)$$

where $\mathbf{U}_k = \{\mathbf{u}_1, \dots, \mathbf{u}_k\}$. Note that no past samples are necessary for the incremental update of $A'(\mathbf{U}_k)$.

The eigen-axis augmentation is judged by using $A'(\mathbf{U}_k)$ as a criterion. Instead of Eq. (5), the update of \mathbf{U}' is carried out based on the following equation.

$$\mathbf{U}'_{k+1} = [\mathbf{U}_k, \hat{\mathbf{h}}]\mathbf{R} \quad (9)$$

where

$$\hat{\mathbf{h}} = \begin{cases} \mathbf{h}/\|\mathbf{h}\| & \text{if } A(\mathbf{U}_k) < \theta \\ \mathbf{0} & \text{otherwise.} \end{cases} \quad (10)$$

Here, θ is a threshold value.

III. INCREMENTAL PRINCIPAL COMPONENT ANALYSIS FOR A CHUNK OF TRAINING SAMPLES

As stated in Section I, the original IPCA is applied on one training sample at a time, and the intermediate eigenproblem must be solved repeatedly for every training sample. Therefore, the learning may get stuck in a deadlock if a large chunk of training samples is given to learn in a short term. To overcome this problem, we extend the original IPCA, so that the eigenspace model Ω can be updated with a chunk of training samples in a single operation. Let us call this extended algorithm *Chunk IPCA*.

A. Update of Mean and Covariance Matrix

Let us assume that N training samples $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \in \mathcal{R}^{n \times N}$ have been given so far and they were already discarded. Instead of keeping actual training samples, we have an eigenspace model $\Omega = (\bar{\mathbf{x}}, \mathbf{U}_k, \mathbf{\Lambda}_k, N)$ where $\bar{\mathbf{x}}$, \mathbf{U}_k , and $\mathbf{\Lambda}_k$ are a mean input vector, an $n \times k$ eigenvector matrix, and a $k \times k$ eigenvalue matrix, respectively. Now, assume that a chunk of L training samples $\mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_L\} \in \mathcal{R}^{n \times L}$ is presented.

Without the previous training samples \mathbf{X} , the updated mean vector $\bar{\mathbf{x}}'$ is easily obtained as follows:

$$\bar{\mathbf{x}}' = \frac{1}{N+L} \left(\sum_{i=1}^N \mathbf{x}_i + \sum_{j=1}^L \mathbf{y}_j \right) = \frac{1}{N+L} (N\bar{\mathbf{x}} + L\bar{\mathbf{y}}). \quad (11)$$

On the other hand, the updated covariance matrix is defined as follows:

$$\mathbf{C}' = \frac{1}{N+L} \left[\sum_{i=1}^N (\mathbf{x}_i - \bar{\mathbf{x}}')(\mathbf{x}_i - \bar{\mathbf{x}}')^T + \sum_{i=1}^L (\mathbf{y}_i - \bar{\mathbf{x}}')(\mathbf{y}_i - \bar{\mathbf{x}}')^T \right]. \quad (12)$$

However, this definition is not suitable for incremental learning because the previous training samples \mathbf{X} are included. Considering that the following relation holds:

$$\begin{aligned} \mathbf{x}_i - \bar{\mathbf{x}}' &= \mathbf{x}_i - \frac{1}{N+L} (N\bar{\mathbf{x}} + L\bar{\mathbf{y}}) \\ &= (\mathbf{x}_i - \bar{\mathbf{x}}) - \frac{L}{N+L} (\bar{\mathbf{y}} - \bar{\mathbf{x}}) \end{aligned} \quad (13)$$

$$\begin{aligned} \mathbf{y}_i - \bar{\mathbf{x}}' &= \mathbf{y}_i - \frac{1}{N+L} (N\bar{\mathbf{x}} + L\bar{\mathbf{y}}) \\ &= \frac{1}{N+L} \{N(\mathbf{y}_i - \bar{\mathbf{x}}) + L(\mathbf{y}_i - \bar{\mathbf{y}})\}, \end{aligned} \quad (14)$$

the new covariance matrix \mathbf{C}' in Eq. (12) can be calculated without the previous training samples \mathbf{X} as shown below.

$$\begin{aligned} \mathbf{C}' &= \frac{1}{N+L} \left[N\mathbf{C} + \frac{NL^2}{(N+L)^2} (\bar{\mathbf{y}} - \bar{\mathbf{x}})(\bar{\mathbf{y}} - \bar{\mathbf{x}})^T \right. \\ &\quad + \frac{N^2}{(N+L)^2} \sum_{i=1}^L (\mathbf{y}_i - \bar{\mathbf{x}})(\mathbf{y}_i - \bar{\mathbf{x}})^T \\ &\quad \left. + \frac{L(L+2N)}{(N+L)^2} \sum_{i=1}^L (\mathbf{y}_i - \bar{\mathbf{y}})(\mathbf{y}_i - \bar{\mathbf{y}})^T \right]. \end{aligned} \quad (15)$$

B. New Eigenproblem

Suppose that l eigen-axes must be augmented to avoid the serious loss of essential input information when a chunk of L training samples \mathbf{Y} is provided; that is, the eigenspace dimensions are increased by l . Let us denote the augmented eigen-axes as follows:

$$\mathbf{H} = [\mathbf{h}_1, \dots, \mathbf{h}_l] \in \mathcal{R}^{n \times l}. \quad (16)$$

Then the updated eigenvector matrix \mathbf{U}'_{k+l} is represented by using the rotation matrix \mathbf{R} and the current eigenvector matrix \mathbf{U}_k .

$$\mathbf{U}'_{k+l} = [\mathbf{U}_k, \mathbf{H}] \mathbf{R}. \quad (17)$$

Hence a new eigenvalue problem to be solved is given by

$$\begin{aligned} \mathbf{C}' \mathbf{U}'_{k+l} &= \mathbf{U}'_{k+l} \mathbf{\Lambda}'_{k+l} \Rightarrow \\ [\mathbf{U}_k, \mathbf{H}]^T \mathbf{C}' [\mathbf{U}_k, \mathbf{H}] \mathbf{R} &= \mathbf{R} \mathbf{\Lambda}'_{k+l} \end{aligned} \quad (18)$$

where $\mathbf{\Lambda}'_{k+l}$ is a new eigenvalue matrix. Substituting Eqs. (15) and (16) into Eq. (18), the following intermediate eigenproblem for Chunk IPCA is obtained.

$$\begin{aligned} &\left(\frac{N}{N+L} \begin{bmatrix} \mathbf{\Lambda}_k & \mathbf{0} \\ \mathbf{0}^T & \mathbf{0} \end{bmatrix} + \frac{NL^2}{(N+L)^3} \begin{bmatrix} \bar{\mathbf{g}}\bar{\mathbf{g}}^T & \bar{\mathbf{g}}\bar{\gamma}^T \\ \bar{\gamma}\bar{\mathbf{g}}^T & \bar{\gamma}\bar{\gamma}^T \end{bmatrix} \right) \\ &\quad + \frac{N^2}{(N+L)^3} \sum_{i=1}^L \begin{bmatrix} \mathbf{g}'_i \mathbf{g}'_i{}^T & \mathbf{g}'_i \gamma'_i{}^T \\ \gamma'_i \mathbf{g}'_i{}^T & \gamma'_i \gamma'_i{}^T \end{bmatrix} \\ &\quad + \frac{L(L+2N)}{(N+L)^3} \sum_{i=1}^L \begin{bmatrix} \mathbf{g}''_i \mathbf{g}''_i{}^T & \mathbf{g}''_i \gamma''_i{}^T \\ \gamma''_i \mathbf{g}''_i{}^T & \gamma''_i \gamma''_i{}^T \end{bmatrix} \Big) \mathbf{R} = \mathbf{R} \mathbf{\Lambda}'_{k+l} \end{aligned} \quad (19)$$

where

$$\begin{aligned} \bar{\mathbf{g}} &= \mathbf{U}_k^T (\bar{\mathbf{y}} - \bar{\mathbf{x}}), \quad \mathbf{g}'_i = \mathbf{U}_k^T (\mathbf{y}_i - \bar{\mathbf{x}}), \quad \mathbf{g}''_i = \mathbf{U}_k^T (\mathbf{y}_i - \bar{\mathbf{y}}), \\ \bar{\gamma} &= \mathbf{H}^T (\bar{\mathbf{y}} - \bar{\mathbf{x}}), \quad \gamma'_i = \mathbf{H}^T (\mathbf{y}_i - \bar{\mathbf{x}}), \quad \gamma''_i = \mathbf{H}^T (\mathbf{y}_i - \bar{\mathbf{y}}). \end{aligned}$$

Note that the following approximation is introduced in the above derivation because we can assume that the previously given training samples do not have essential information in the new augmented subspace spanned by \mathbf{H}

$$[\mathbf{U}_k, \mathbf{H}]^T \mathbf{C} [\mathbf{U}_k, \mathbf{H}] \approx \begin{bmatrix} \mathbf{\Lambda}_k & \mathbf{0} \\ \mathbf{0}^T & \mathbf{0} \end{bmatrix}. \quad (20)$$

Solving this intermediate eigenproblem, a new rotation matrix \mathbf{R} and the eigenvalue matrix $\mathbf{\Lambda}'_{k+l}$ are obtained. Then, the corresponding new eigenvector matrix \mathbf{U}'_{k+l} is given by Eq. (17).

C. A New Criterion for Eigen-axes Augmentation

In the proposed Chunk IPCA, the number of eigen-axes to be augmented is determined by finding a minimum k such that the accumulation ratio $A(\mathbf{U}_k)$ in Eq. (7) satisfies the following condition: $A(\mathbf{U}_k) > \theta$. However, the number of augmented eigen-axes is not restricted to one when a chunk of training samples is learned at a time. Therefore, the update equation of $A(\mathbf{U}_k)$ in Eq. (8) must be modified such that it can be updated with a chunk of training samples in one-pass.

First, let us consider how to update the numerator of Eq. (7). Considering the fact that the total amount of eigenvalues is equivalent to the summation of variances σ_i^2 , the numerator is reduced to

$$\begin{aligned} \sum_{i=1}^k \lambda'_i &= \sum_{i=1}^k \sigma_i'^2 \\ &= \sum_{i=1}^k \frac{1}{N+L} \left[\sum_{j=1}^N \{\mathbf{u}_i^T(\mathbf{x}_j - \bar{\mathbf{x}}')\}^2 \right. \\ &\quad \left. + \sum_{j=1}^L \{\mathbf{u}_i^T(\mathbf{y}_j - \bar{\mathbf{y}}')\}^2 \right]. \end{aligned} \quad (21)$$

In the derivation of Eq. (21), we use the fact that the total energy of the training samples in the eigenspace spanned by the eigenvectors \mathbf{u}_i ($i = 1, \dots, k$) is equivalent to that in the eigenspace spanned by the updated eigenvectors \mathbf{u}'_i ($i = 1, \dots, k$); that is, only the axis-rotation does not change the energy.

In Eq. (21), the new mean $\mathbf{u}_i^T \bar{\mathbf{x}}'$ in the feature space is obtained from Eq. (11).

$$\mathbf{u}_i^T \bar{\mathbf{x}}' = \frac{1}{N+L} \mathbf{u}_i^T (N\bar{\mathbf{x}} + L\bar{\mathbf{y}}). \quad (22)$$

Using Eq. (22), the first term in the square brackets of Eq. (21) is calculated as follows:

$$\begin{aligned} &\sum_{j=1}^N \{\mathbf{u}_i^T(\mathbf{x}_j - \bar{\mathbf{x}}')\}^2 \\ &= \sum_{j=1}^N \left[\mathbf{u}_i^T \left\{ \mathbf{x}_j - \frac{1}{N+L} (N\bar{\mathbf{x}} + L\bar{\mathbf{y}}) \right\} \right]^2 \\ &= \sum_{j=1}^N \left[\mathbf{u}_i^T \left\{ (\mathbf{x}_j - \bar{\mathbf{x}}) + \frac{L}{N+L} (\bar{\mathbf{x}} - \bar{\mathbf{y}}) \right\} \right]^2 \\ &= \sum_{j=1}^N \{\mathbf{u}_i^T(\mathbf{x}_j - \bar{\mathbf{x}})\}^2 + \frac{NL^2}{(N+L)^2} \{\mathbf{u}_i^T(\bar{\mathbf{x}} - \bar{\mathbf{y}})\}^2 \\ &= N\lambda_i + \frac{NL^2}{(N+L)^2} \{\mathbf{u}_i^T(\bar{\mathbf{x}} - \bar{\mathbf{y}})\}^2. \end{aligned} \quad (23)$$

In the same way, the second term in the square brackets of Eq. (21) is reduced to

$$\sum_{j=1}^L \{\mathbf{u}_i^T(\mathbf{y}_j - \bar{\mathbf{y}}')\}^2$$

$$\begin{aligned} &= \sum_{j=1}^L \left[\mathbf{u}_i^T \left\{ (\mathbf{y}_j - \bar{\mathbf{y}}) - \frac{N}{N+L} (\bar{\mathbf{x}} - \bar{\mathbf{y}}) \right\} \right]^2 \\ &= \sum_{j=1}^L \{\mathbf{u}_i^T(\mathbf{y}_j - \bar{\mathbf{y}})\}^2 + \frac{N^2L}{(N+L)^2} \{\mathbf{u}_i^T(\bar{\mathbf{x}} - \bar{\mathbf{y}})\}^2. \end{aligned} \quad (24)$$

Substituting Eqs. (23) and (24) into Eq. (21), the numerator of Eq. (7) is given by

$$\begin{aligned} \sum_{i=1}^k \lambda'_i &= \frac{N}{N+L} \left[\sum_{i=1}^k \lambda_i + \frac{L}{N+L} \|\mathbf{U}_k^T(\bar{\mathbf{x}} - \bar{\mathbf{y}})\|^2 \right. \\ &\quad \left. + \frac{1}{N} \sum_{j=1}^L \|\mathbf{U}_k^T(\mathbf{y}_j - \bar{\mathbf{y}})\|^2 \right] \end{aligned} \quad (25)$$

where $\mathbf{U}_k = \{\mathbf{u}_1, \dots, \mathbf{u}_k\}$. In the similar manner, the denominator in Eq. (7) is calculated as follows:

$$\begin{aligned} \sum_{i=1}^n \lambda'_i &= \frac{N}{N+L} \left[\sum_{i=1}^n \lambda_i + \frac{L}{N+L} \|\bar{\mathbf{x}} - \bar{\mathbf{y}}\|^2 \right. \\ &\quad \left. + \frac{1}{N} \sum_{j=1}^L \|\mathbf{y}_j - \bar{\mathbf{y}}\|^2 \right]. \end{aligned} \quad (26)$$

Then, the update equation for the accumulation ratio $A'(\mathbf{U}_k)$ is given by

$$\begin{aligned} A'(\mathbf{U}_k) &= \frac{\sum_{i=1}^k \lambda'_i}{\sum_{i=1}^n \lambda'_i} \\ &= \frac{\sum_{i=1}^k \lambda_i + \frac{L}{N+L} \|\mathbf{U}_k^T(\bar{\mathbf{x}} - \bar{\mathbf{y}})\|^2 + \frac{1}{N} \sum_{j=1}^L \|\mathbf{U}_k^T(\mathbf{y}_j - \bar{\mathbf{y}})\|^2}{\sum_{i=1}^n \lambda_i + \frac{L}{N+L} \|\bar{\mathbf{x}} - \bar{\mathbf{y}}\|^2 + \frac{1}{N} \sum_{j=1}^L \|\mathbf{y}_j - \bar{\mathbf{y}}\|^2}. \end{aligned} \quad (27)$$

As we can see from Eq. (27), no past samples \mathbf{x}_j are needed to update $A'(\mathbf{U}_k)$.

D. Selection of Eigen-axes

In IPCA, a new eigen-axis is selected so as to be perpendicular to the existing eigenvectors which are given by the column vectors of \mathbf{U}_k . A straightforward way to get new eigen-axes is to apply Gram-Schmidt orthogonalization technique to the given chunk of training samples [25]. If the chunk samples are represented by \tilde{L} linearly independent vectors, the maximum number of eigen-axes to be augmented is also \tilde{L} . However, the feature space spanned by all of the augmented eigen-axes is redundant in general; in addition, if the chunk size is large, the computation costs to solve the intermediate eigenproblem in Eq. (19) would be considerably expensive. Therefore, we need to select informative eigen-axes from the \tilde{L} eigen-axes efficiently.

Since the number of eigen-axes to be augmented is varied from 0 to \tilde{L} , the number of possible combinations of eigen-axes are represented by $\sum_{i=0}^{\tilde{L}} \binom{\tilde{L}}{i}$. If the chunk size is large, the computation costs would be large to find the optimal set of augmented eigen-axes. To avoid such an exhaustive search, we adopt a kind of greedy search based on the accumulation ratio in Eq. (27).

To construct a compact feature space, we should find a smallest set of augmented eigen-axes such that the eigenspace includes as much the energy of the given chunk data as possible. A straightforward way to find the set is to select an eigen-axis one by one, each of which gives a maximum accumulation ratio. Let us explain more concretely. First, the residue vectors for given L training samples are calculated based on Eq. (2) and they are put into a set $\mathcal{L} = \{\mathbf{h}_1, \dots, \mathbf{h}_L\}$. Next, a residue vector \mathbf{h}_1 is temporally added to the set of eigen-axes \mathbf{U}_k , and the accumulation ratio $A'([\mathbf{U}_k, \mathbf{h}_1])$ is calculated. This operation is carried out for every \mathbf{h}_l ($l = 1, \dots, L$). Then, the residue vector \mathbf{h}_{l^*} with the maximum $A'([\mathbf{U}_k, \mathbf{h}_{l^*}])$ is selected as the first eigen-axis, and \mathbf{h}_{l^*} is removed from \mathcal{L} . If $A'([\mathbf{U}_k, \mathbf{h}_{l^*}])$ is larger than a threshold value θ , the selection of eigen-axes is terminated. Otherwise, the same procedure is applied to the new set \mathcal{L} to acquire a next eigen-axis. The algorithm of the eigen-axis selection is summarized below.

[Algorithm of Eigen-axis Selection]

- i) Set $\mathbf{H} = \{ \}$ and calculate the accumulation ratio $A'(\mathbf{U}_k)$ in Eq. (27). If $A'(\mathbf{U}_k) > \theta$ (const.), terminate this algorithm. Otherwise, go to Step ii).
- ii) Obtain a set \mathcal{L} of the following residue vectors \mathbf{h}_l for the given training samples \mathbf{y}_l ($l = 1, \dots, L$).

$$\mathbf{h}_l = \frac{\mathbf{r}_l}{\|\mathbf{r}_l\|} \quad (28)$$

where

$$\mathbf{r}_l = (\mathbf{y}_l - \bar{\mathbf{x}}) - [\mathbf{U}_k, \mathbf{H}][\mathbf{U}_k, \mathbf{H}]^T(\mathbf{y}_l - \bar{\mathbf{x}}).$$

- iii) Find the following residue vectors \mathbf{h}_{l^*} which gives the maximum accumulation ratio $A'([\mathbf{U}_k, \mathbf{H}, \mathbf{h}_l])$:

$$l^* = \arg \max_l A'([\mathbf{U}_k, \mathbf{H}, \mathbf{h}_l]).$$

- iv) Update $\mathbf{H}' = [\mathbf{H}, \mathbf{h}_{l^*}]$ and remove \mathbf{h}_{l^*} from \mathcal{L} .
- v) If $A'([\mathbf{U}_k, \mathbf{H}']) > \theta$ (const.), terminate this algorithm. Otherwise, go to Step iii).

IV. EXPERIMENTS

A. Experimental Setup

We select three datasets from the UCI Machine Learning Repository [24] (see Table I). To construct an initial eigenspace, 5% of training samples are applied to the conventional PCA. The remaining 95% of training samples are sequentially provided as shown in Fig. 1. Although the proposed chunk IPCA algorithm can work even if the chunk

TABLE I
EVALUATED UCI DATASETS.

Database Name	#Attr.	#Class	#Train Data
Spambase	57	2	2301
Image Segmentation	19	7	210
Vowel	10	11	528

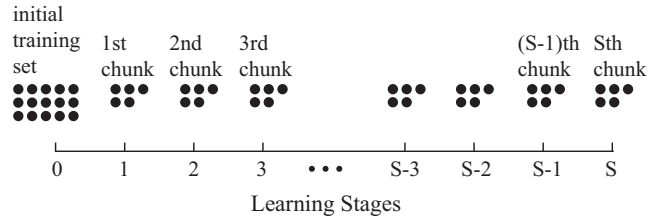


Fig. 1. The presentation of training samples.

TABLE II
AVERAGE CPU TIME (SEC.) TO CALCULATE AN EIGENSPACE.

	Spam	Image	Vowel
Original IPCA	27.4	0.073	0.109
Chunk IPCA	0.9	0.023	0.058

size is varied at every learning stage, we assume that the size is fixed with L during the learning.

A chunk of training samples are randomly selected and it has no overlap with other chunks; hence all the training samples are presented only once. Thus, the number of learning stages S is given by $S = \lceil 0.95N/L \rceil$ where N is the total number of training samples. Note that the number of training samples in the last chunk can be less than L , and it is given by $N - L\lceil 0.95N/L \rceil$. Since the performance of incremental learning generally depends on the sequence of training samples, ten trials with different sequences of training samples are conducted to evaluate the average performance.

B. Computational Efficiency

The computation costs of the eigenspace update are estimated by measuring the CPU time using a Matlab function. Table II shows the average CPU time to calculate an eigenspace for the original IPCA and the proposed Chunk IPCA. Here, the chunk size L is set to 10. As you can see from Table II, the computation costs of Chunk IPCA is significantly reduced against Original IPCA especially for large datasets. The experimental results here suggest that the proposed method possesses excellent scalability.

Since the eigen-axes to be augmented are selected based on a greedy method (see III-D), the optimality of the selected eigen-axes is not ensured. Considering that more axes are generally added if the chunk size is larger, it is considered that the error of eigen-axes obtained by Chunk IPCA could increase when a large chunk of training samples are given. On the contrary, however, it is expected that the learning time can be shortened when the chunk size is large. Therefore, the evaluation of Chunk IPCA should be made for various chunk sizes in terms of both approximation accuracy and learning time.

Figure 2 shows the learning time when training samples are provided in various sizes of chunk. The relative CPU time is defined as the ratio of the CPU time to that for $L=1$. As seen from Fig. 2, the computation time is greatly reduced as the chunk size becomes large. It is considered that the

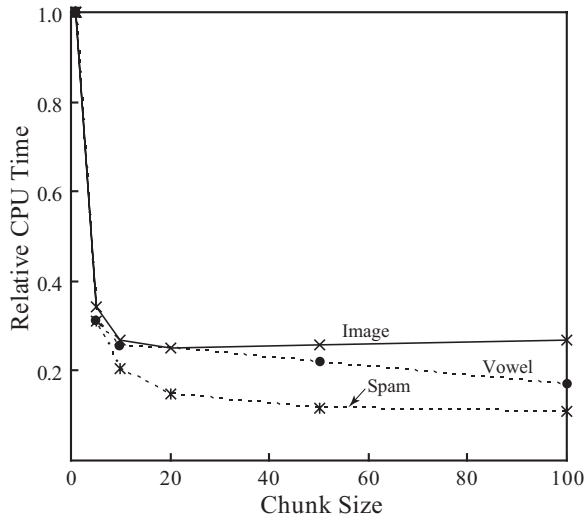


Fig. 2. Relative CPU time of Chunk IPCA when the chunk size L is set to 1, 2, 5, 10, 20, 50, and 100.

reduction in computation time is mainly attained by reducing the repeated times of solving the intermediate eigenproblem.

C. Accuracy of Updated Eigenvectors

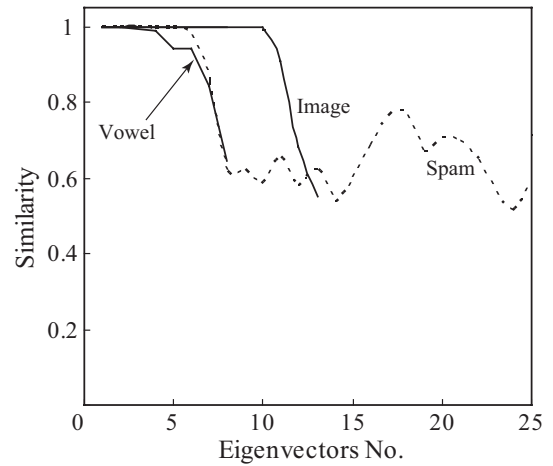
To see whether an appropriate feature space is constructed by Chunk IPCA, the similarity of eigenvectors obtained by Batch PCA and Chunk IPCA is examined. In Batch PCA, an eigen-feature space is updated at every learning stage by applying PCA to all the training samples given so far. Since Batch PCA always gives an accurate eigenspace, we adopt the eigenvectors obtained by Batch PCA as target eigenvectors of Chunk IPCA. Thus, the similarity is measured by the following directional cosine d_i :

$$d_i = \frac{1}{M} \sum_{j=1}^M \frac{\mathbf{u}_{ji}^{(b)T} \mathbf{u}_{ji}^{(c)}}{\|\mathbf{u}_{ji}^{(b)}\| \|\mathbf{u}_{ji}^{(c)}\|} \quad (29)$$

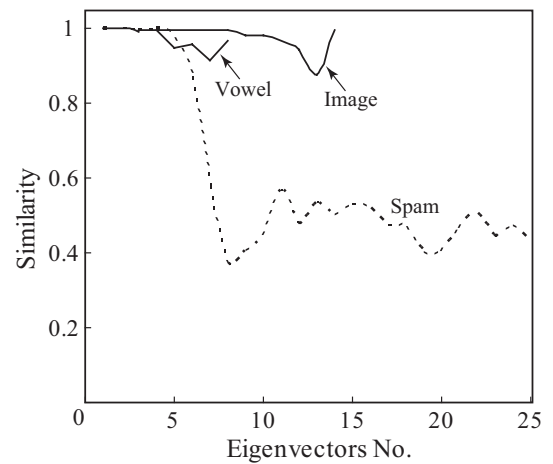
where $\mathbf{u}_{ji}^{(b)}$ and $\mathbf{u}_{ji}^{(c)}$ are respectively the i th eigenvector obtained by Batch PCA and Chunk IPCA in the j th trial, and M is the number of trials to average the similarity. Obviously, if the similarity is one, it means two eigenvectors are identical.

Figure 3 shows the average similarity between the eigenvectors obtained by Batch PCA and Chunk IPCA for the three UCI datasets in Table I. The horizontal axis corresponds to the number of eigenvectors with the largest 25 eigenvalues. Since the number of attributes for Image Segmentation and Vowel datasets is less than 25, the lines for these two datasets terminate at the number of their attributes. Figures 3 (a) and (b) show the similarity when the chunk size L is 10 and 50, respectively.

As seen from Fig. 3, in almost all cases, the similarity is kept above 0.9 for the major eigenvectors. From the results in Fig. 3, we can say that the proposed Chunk IPCA gives a good approximation to major eigenvectors with large



(a) $L=10$



(b) $L=50$

Fig. 3. Average similarity between eigenvectors obtained by Batch PCA and Chunk IPCA for the three UCI datasets.

eigenvalues even if the eigen-axes are selected based on a greedy method.

On the other hand, the approximation to minor eigenvectors, whose normalized eigenvalues are almost zero, has a large error. The primary reasons for this is originated from the approximation error introduced in the derivation of the intermediate eigenproblem in Eq. (19). This approximation error could be small if the threshold for accumulation ratio θ is set properly. However, it is not easy to know a proper value of θ in advance because it depends on the sequence of training samples. In the one-pass learning situation assumed here, a chunk of training samples given sequentially is thrown away after the learning is done at every stage. Therefore, if the sample distributions at latter learning stages are largely varied from those at early stages, some crucial information would be lost during learning, and the loss would prevent from constructing an effective eigenspace. To overcome this problem, we should introduce an adaptive mechanism for θ , but this is left as a further work.

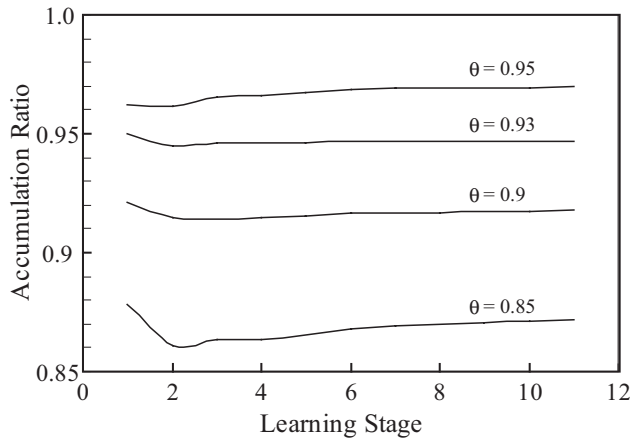


Fig. 4. Time course of average accumulation ratios for different threshold θ when Vowel dataset is used.

D. Controllability of Eigenspace Dimensions

Although proper selection of a threshold value θ is not easily done due to the reason mentioned above, we have to at least ensure that the accumulation ratio is always sustained above θ during the incremental learning so as not to lose crucial information unexpectedly. To check this out, we examine the average time course of accumulation ratio for different θ .

Figure 4 illustrates the transitions of average accumulation ratios over the entire learning stages for Vowel dataset. The time courses of the accumulation ratio are obtained by averaging over the ten trials. In this experiment, the chunk size L is set to 50, and the threshold θ is changed from 0.85 to 0.95. The accumulation ratio in Fig. 4 is calculated after the update of feature space; that is, it is calculated after the rotation and/or the augmentation of eigen-axes are done. On the other hand, the check of the accumulation ratio is carried out before the update of feature space. Hence, the time course in Fig. 4 should always be above a given threshold θ . As seen from Fig. 4, the proposed Chunk IPCA succeeds in sustaining the accumulation ratio above the threshold θ .

V. CONCLUSIONS

In our previous works [18], [19], we have proposed an adaptive evolving connectionist model in which Incremental Principal Component Analysis (IPCA) and Evolving Clustering Method (ECM) are effectively combined. This learning scheme gives a new concept for pattern recognition systems: feature selection and classifier learning are simultaneously carried out on-line. One drawback of this approach was scalability in terms of the number of data samples and the number of their attributes. This drawback comes from the limitation of the previous approach where a training sample must be applied one by one even if a chunk of training sample is given at a time.

To overcome this problem, we proposed a new algorithm of dynamic feature selection called Chunk IPCA, in which a chunk of training samples can be applied at a time to update

an eigenspace model incrementally. The novelty of this paper lies in the derivation of Chunk IPCA, the algorithm of eigen-axis selection, the introduction of accumulation ratio as a criterion of eigen-axis augmentation. In this paper, we did not mention about how to combine Chunk IPCA and classifier models (e.g., ECM and neural networks). However, it is not difficult to combine them by introducing the update method of prototype vectors, which has already been presented in our previous works [18], [20]. Therefore, it is expected that the proposed Chunk IPCA enables a classification system to learn more efficiently.

To verify if the proposed Chunk IPCA can work properly for any size of chunk data, we tested it on three standard datasets. The experimental results suggested that the proposed learning scheme worked quite well even when training samples were provided in various sizes of data chunks. Furthermore, we examined the approximation error of the eigenvectors obtained by Chunk IPCA and the controllability of eigenspace dimensionality. As a result, Chunk IPCA learned major eigenvectors without serious errors and could maintain the designated accumulation ratio by adding new eigen-axes properly.

There still remains several open questions. First, since the features are selected without considering the class separability in IPCA, optimal features are not always ensured. To alleviate this problem, recently we have proposed Incremental Linear Discriminant Analysis (ILDA) [17] in which only the axis rotation is carried out in an incremental way. Incorporating this ILDA algorithm into the proposed learning scheme is very promising. Hence, we need to extend it such that the feature space is automatically expanded in dimensions. Second, when constructing a classification system, the proposed Chunk IPCA must be combined with a classifier model. We have presented several attempts to combine with the k -nearest neighbor method and a variant model of Radial Basis Function network [17], [18], [20]. However, there are some other promising classifiers we can choose such as Dynamic Evolving Neuro-fuzzy Inference System (DENFIS) [26] and SVM Classification Tree [27], which were developed by some of the authors. In addition, the kernel method has received a great attention in pattern classification. There have been proposed several kernel-based feature selection methods such as Kernel PCA [28] and Kernel Discriminant Analysis [29]. Hence, pursuing the optimal combination of the feature space learning and the classifier learning must also be an interesting issue.

ACKNOWLEDGMENT

The authors would like to thank Prof. Shigeo Abe for his useful discussions and comments.

REFERENCES

- [1] Q. Zhu, "Pattern Classification in Dynamic Environments: Tagged Feature-class Representation and the Classifiers," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19, no. 5, pp. 1203-1209, 1989.

- [2] B. Yamauchi and R. Beer, "Spatial Learning for Navigation in Dynamic Environments," *IEEE Transactions on Systems, Man and Cybernetics*, Part B, vol. 26, no. 3, pp. 496-505, 1996.
- [3] K. Tsumori and S. Ozawa, "Incremental Learning in Dynamic Environments Using Neural Network with Long-term Memory," *Proc. of Int. Joint Conf. on Neural Networks*, vol. 4, pp. 2583-2588, 2003.
- [4] N. Kubota, "A Spiking Neural Network for Behavior Learning of a Mobile Robot in a Dynamic Environment," *Proc. of IEEE Int. Conf. on Systems, Man and Cybernetics*, vol. 6, pp. 5783-5788, 2004.
- [5] J. Weng, C. H. Evans, and W.-S. Hwang, "An Incremental Learning Method for Face Recognition under Continuous Video Stream," *Proc. Fourth IEEE Int. Conf. on Automatic Face and Gesture Recognition*, pp. 251-256, 2000.
- [6] S. L. Toh and S. Ozawa, "A Face Recognition System Using Neural Networks with Incremental Learning Ability," *Proc. 8th Australian and New Zealand Conf. on Intelligent Information Systems*, pp. 389-394, 2003.
- [7] G. A. Carpenter and S. Grossberg, "The ART of Adaptive Pattern Recognition by a Self-organizing Neural Network," *IEEE Computer*, vol. 21, no. 3, pp. 77-88, 1988.
- [8] C. G. Atkeson, A. W. Moore, and S. Schaal, "Locally Weighted Learning," *Artificial Intelligence Review*, vol. 11, pp. 75-113, 1997.
- [9] S. Thrun and L. Pratt, *Learning to Learn*, Kluwer Academic Pub., 1998.
- [10] R. Polikar, L. Upda, S. S. Upda, and V. Honavar, "Learn++: an Incremental Learning Algorithm for Supervised Neural Networks," *IEEE Trans. on Systems, Man and Cybernetics*, Part C, vol. 31, no. 4, pp. 497-508, 2001.
- [11] G.-B. Huang, P. Saratchandran, and N. Sundararajan, "An Efficient Sequential Learning Algorithm for Growing and Pruning RBF (GAP-RBF) Networks," *IEEE Trans. on Systems, Man and Cybernetics*, Part B, vol. 34, no. 6, pp. 2284-2292, 2004.
- [12] E. Oja and J. Karhunen, "On Stochastic Approximation of the Eigenvectors and Eigenvalues of the Expectation of a Random Matrix," *J. Math. Analysis and Application*, vol. 106, pp. 69-84, 1985.
- [13] T. D. Sanger, "Optimal Unsupervised Learning in a Single-layer Linear Feedforward Neural Network," *IEEE Trans. Neural Networks*, vol. 2, pp. 459-473, 1989.
- [14] J. Weng, Y. Zhang, and W.-S. Hwang, "Candidate Covariance-Free Incremental Principal Component Analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 1034-1040, 1993.
- [15] P. Hall and R. Martin, "Incremental Eigenanalysis for Classification," *Proc. British Machine Vision Conference*, vol. 1, pp. 286-295, 1998.
- [16] J. Yan, Q.-S. Cheng, Q. Yang, and B. Zhang, "An Incremental Subspace Learning Algorithm to Categorize Large Scale Text Data," *Prof. of 7th Asia-Pacific Web Conference*, pp. 52-63, 2005.
- [17] S. Pang, S. Ozawa, and N. Kasabov, "Incremental Linear Discriminant Analysis for Classification of Data Streams," *IEEE Trans. on Systems, Man, and Cybernetics*, Part B, vol. 35, no. 5, pp. 905-914, 2005.
- [18] S. Ozawa, S. Pang, and N. Kasabov, "A Modified Incremental Principal Component Analysis for On-line Learning of Feature Space and Classifier," in C. Zhang, H. W. Guesgen, and W. K. Yeap (Eds.), *PRICAI 2004: Trends in Artificial Intelligence* LNAI, Springer-Verlag, pp. 231-240, 2004.
- [19] S. Pang, S. Ozawa, and N. Kasabov, "One-pass Incremental Membership Authentication by Face Classification," in Biometric Authentication, D. Zhang and A. K. Jain (Eds.), *Lecture Notes in Computer Science*, Springer-Verlag, pp. 155-161, 2004.
- [20] S. Ozawa, S. L. Toh, S. Abe, S. Pang, and N. Kasabov, "Incremental Learning of Feature Space and Classifier for Face Recognition," *Neural Networks*, vol. 18, no. 5-6, pp. 575-584, 2005.
- [21] N. Kasabov, *Evolving Connectionist Systems: Methods and Applications in Bioinformatics, Brain Study and Intelligent Machines*, Springer-Verlag, 2002
- [22] M. Kobayashi, A. Zamani, S. Ozawa, and S. Abe, "Reducing Computations in Incremental Learning for Feedforward Neural Network with Long-term Memory," *Proc. Int. Joint Conf. on Neural Networks*, vol. 3, pp. 1989-1994, 2001.
- [23] H. G. Loos, "Parity Madeline: a Neural Net with Complete Boolean Repertoire Capable of One-pass Learning," *Proc. of Int. Joint Conf. on Neural Networks*, vol. 2, pp. 111-118, 1989.
- [24] <http://www.ics.uci.edu/mllearn/MLRepository.html>
- [25] P. Hall, D. Marshall, and R. Martin, "Merging and Splitting Eigenspace Models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 9, pp. 1042-1049, 2000.
- [26] N. Kasabov and Q. Song, "DENFIS: Dynamic Evolving Neuro-fuzzy Inference System and Its Application for Time-series Prediction," *IEEE Trans. on Fuzzy systems*, vol. 10, no. 2, pp. 144-154, 2002.
- [27] S. Pang, D. Kim, and S. Y. Bang, "Face Membership Authentication Using SVM Classification Tree Generated by Membershipbased LLE Data Partition," *IEEE Trans. on Neural Network*, vol. 16, no. 2, pp. 436-446, 2005.
- [28] B. Schölkopf, A. Smola, and K. -R. Müller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," *MPI Technical Report*, no. 44, 1996.
- [29] G. Baudat and F. Anouar, "Generalized Discriminant Analysis Using a Kernel Approach," *Neural Computation*, vol. 12, pp. 2385-2404, 2000.