A Modified Incremental Principal Component Analysis for On-line Learning of Feature Space and Classifier

Seiichi Ozawa[*], Shaoning Pang[†], and Nikola Kasabov[†]

* Graduate School of Science and Technology, Kobe University
1-1 Rokko-dai, Nada-ku, Kobe 657-8501, Japan
Phone/Fax: +81 78 803 6466

† Knowledge Engineering & Discover Research Institute
Auckland University of Technology, Private Bag 92006, Auckland
1020, New Zealand
Phone: +64 9 917 9506, Fax: +64 9 917 9501

[**Contact Author**] Seiichi Ozawa (ozawasei@kobe-u.ac.jp)

[**Abstract**] We have proposed a new concept for pattern classification systems in which feature selection and classifier learning are simultaneously carried out on-line. To realize this concept, Incremental Principal Component Analysis (IPCA) and Evolving Clustering Method (ECM) was effectively combined in the previous work. However, in order to construct a desirable feature space, a threshold value to determine the increase of a new feature is properly given in the original IPCA. To alleviate this problem, we can adopt the accumulation ratio as its criterion. However, in incremental situations, the accumulation ratio must be modified every time a new sample is given. Therefore, in order to use this ratio as a criterion, we also need to develop a one-pass update algorithm for this ratio. In this paper, we propose an improved algorithm of IPCA in which the accumulation ratio as well as the feature space can be updated on-line without all the past samples. To see if correct feature construction is carried out by this new IPCA algorithm, the recognition performance is evaluated for some standard datasets when Evolving Clustering Method (ECM) is adopted as a prototype learning method in Nearest Neighbor classifier.

[**Content Areas**] Machine learning, Neural networks

1 Introduction

In many real-world applications such as pattern recognition and time-series prediction, we often confront difficult situations where a complete set of training samples is not given in advance. In face recognition tasks, for example, human faces have large variations depending on expressions, lighting conditions, make-up, hairstyles, and so forth. When a human is registered in a person identification system, it is difficult to consider all variations in face images in the first place [1]. Another difficulty in the realistic recognition problems lies in the uncertainty of data distribution; that is, we cannot know what training samples will appear in the future. Hence, it is quite difficult to extract essential features only from initially given training samples.

To solve these problems, we should select appropriate features on-line based on the property of an input data stream. This means that not only classifier but also feature space must be incrementally trained. For this purpose, a new concept of incremental learning have been proposed in which the feature selection and classifier learning are simultaneously carried out on-line [2, 3]. One of the great advantages in this concept is that classification systems can improve their performance constantly even if an insufficient number of training samples are given at the early stage, often resulting in inappropriate selection of features and poor classifier performance.

To realize the above two desirable characteristics in recognition systems, we have proposed a one-pass incremental learning scheme which consists of Incremental Principal Component Analysis (IPCA) [4] and Evolving Clustering Method (ECM) [5]. In order to construct a proper feature space, however, a suitable threshold value to determine the dimensional augmentation should be given in IPCA; this optimization often leads to annoying parameter search. This paper presents a remedy for this problem by introducing the accumulation ratio into IPCA as its criterion. Since the accumulation ratio is usually calculated from all the given samples, in order to develop one-pass incremental learning algorithm for feature space and classifier, we have to devise an incremental update algorithm for this ratio without keeping any past training samples.

In Section 2, we briefly review the original IPCA, and then we present a new criterion to determine eigenspace dimensionality and derive its incremental update algorithm. Section 3 describes the proposed learning scheme for both feature space and classifier. In Section 4, the proposed incremental learning scheme is evaluated for the three standard datasets in UCI Machine Learning Repository. Section 5 summarizes the facts we've done here and denotes our remaining future works.

2 Incremental Principal Component Analysis (IPCA)

2.1 Original IPCA Algorithm

Principal Component Analysis (PCA) is one of the most popular and powerful feature extraction techniques in pattern classification problems. Although the original PCA is not suited for incremental learning purposes, Hall and Martin have devised a method to update eigenvectors and eigenvalues in an incremental way [4].

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

Let us consider the case that the $(N+1)$th training sample $\boldsymbol{y}$ is presented. The addition of this new sample will lead to the changes in both of the mean vector and covariance matrix; therefore, the eigenvectors and eigenvalues should also be recalculated. The mean input vector $\bar{\boldsymbol{x}}$ is easily updated as follows:

$$\bar{\boldsymbol{x}}' = \frac{1}{N+1}(N\bar{\boldsymbol{x}} + \boldsymbol{y}). \tag{1}$$

The problem is how to update the eigenvectors and eigenvalues.

When the eigenspace model Ω is reconstructed to adapt to a new sample, we must check if the dimensions of the eigenspace should be changed or not. If the new sample has almost all energy in the current eigenspace, the dimensional augmentation is not needed in reconstructing the eigenspace. However, if it has some energy in the complementary space to the current eigenspace, the dimensional augmentation cannot be avoided. This can be judged from the norm of the following residue vector $\boldsymbol{h}$:

$$\boldsymbol{h} = (\boldsymbol{y} - \bar{\boldsymbol{x}}) - \boldsymbol{U}\boldsymbol{g} \tag{2}$$

where

$$\boldsymbol{g} = \boldsymbol{U}^T(\boldsymbol{y} - \bar{\boldsymbol{x}}). \tag{3}$$

Here, T means the transposition of vectors and matrices. When the norm of the residue vector $\boldsymbol{h}$ is larger than a threshold value η, it must allow the number of dimensions to increase from k to $k+1$, and the current eigenspace must be expanded in the direction of $\boldsymbol{h}$. Otherwise, the number of dimensions remains the same.

It has been shown that the eigenvectors and eigenvalues should be updated based on the solution of the following intermediate eigenproblem [4]:

$$\left(\frac{N}{N+1}\begin{bmatrix} \boldsymbol{\Lambda} & \boldsymbol{0} \\ \boldsymbol{0}^T & 0 \end{bmatrix} + \frac{N}{(N+1)^2}\begin{bmatrix} \boldsymbol{g}\boldsymbol{g}^T & \gamma\boldsymbol{g} \\ \gamma\boldsymbol{g}^T & \gamma^2 \end{bmatrix}\right)\boldsymbol{R} = \boldsymbol{R}\boldsymbol{\Lambda}' \tag{4}$$

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

$$\boldsymbol{U}' = [\boldsymbol{U},\ \hat{\boldsymbol{h}}]\boldsymbol{R} \tag{5}$$

where

$$\hat{\boldsymbol{h}} = \begin{cases} \boldsymbol{h}/\|\boldsymbol{h}\| & \text{if } \|\boldsymbol{h}\| > \eta \\ \boldsymbol{0} & \text{otherwise.} \end{cases} \tag{6}$$

Here, η is a small threshold value which is set to zero in the original IPCA [4]. As you can see from Eq. (5), $\boldsymbol{R}$ operates to rotate the eigenvectors; hence, let us call $\boldsymbol{R}$ a rotation matrix in the following. Note that if $\hat{\boldsymbol{h}} = \boldsymbol{0}$, $\boldsymbol{R}$ degenerates into a $n \times k$ matrix; that is, the dimensions of the updated eigenspace remains the same as those of the previous eigenspace.

2.2 A New Criterion for Increasing Eigenspace Dimensionality

As seen in Eq. (6), the dimensional augmentation is carried out whenever the norm of a residue vector is larger than a threshold value η. 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, we cannot get an efficient feature space with small dimensions; this may result in deteriorating generalization performance and computational efficiency.

To reduce this dependency in determining appropriate feature space dimensions, the following accumulation ratio is often used as its criterion:

$$A(k) = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{j=1}^{n} \lambda_j} \tag{7}$$

where λ_i is the ith largest eigenvalue, k and n are the numbers of dimensions of the current feature space and input space, respectively. By specifying an appropriate threshold value θ, we can determine the feature space dimensions by searching for a minimum k such that $A(k) > \theta$ holds. In general, the update of Eq. (7), cannot be done without the training samples given previously. This is a serious problem when we device a one-pass incremental learning algorithm. To solve this problem, we propose an incremental update algorithm of $A(k)$ without keeping all the past training samples.

First let us consider the numerator of Eq. (7). Using the fact that the total amount of eigenvalues is equivalent to the summation of variances σ_i^2, the numerator is given by

$$\sum_{i=1}^{k} \lambda_i = \sum_{i=1}^{k} \sigma_i^2 = \frac{1}{N} \sum_{i=1}^{k} \sum_{j=1}^{N} \{\boldsymbol{u}_i^T (\boldsymbol{x}^{(j)} - \bar{\boldsymbol{x}})\}^2 \tag{8}$$

where $\boldsymbol{u}_i$ is the ith column vector of $\boldsymbol{U}$.

Assume that a new sample $\boldsymbol{y}$ is given, the new mean $\boldsymbol{u}_i^T \bar{\boldsymbol{x}}'$ of feature values on $\boldsymbol{u}_i$ is calculated as follows:

$$\boldsymbol{u}_i^T \bar{\boldsymbol{x}}' = \frac{1}{N+1} \boldsymbol{u}_i^T (N\bar{\boldsymbol{x}} + \boldsymbol{y}) \tag{9}$$

From Eqs. (8) and (9), the total amount of new eigenvalues is given by

$$\begin{aligned}\sum_{i=1}^{k}\lambda'_i = \sum_{i=1}^{k}\sigma_i'^2 = \sum_{i=1}^{k}\frac{1}{N+1}\Big[\sum_{j=1}^{N}\{\boldsymbol{u}_i^T(\boldsymbol{x}^{(j)}-\bar{\boldsymbol{x}}')\}^2+\{\boldsymbol{u}_i^T(\boldsymbol{y}-\bar{\boldsymbol{x}}')\}^2\Big] \\ = \frac{N}{N+1}\sum_{i=1}^{k}\sigma_i^2+\frac{N}{(N+1)^2}\sum_{i=1}^{k}\{\boldsymbol{u}_i^T(\boldsymbol{y}-\bar{\boldsymbol{x}})\}^2 \\ = \frac{N}{N+1}\sum_{i=1}^{k}\lambda_i+\frac{N}{(N+1)^2}\|\boldsymbol{U}_k^T(\boldsymbol{y}-\bar{\boldsymbol{x}})\|^2 \end{aligned} \quad (10)$$

where $\boldsymbol{U}_k = \{\boldsymbol{u}_1, \cdots, \boldsymbol{u}_k\}$. In the similar manner, the denominator in Eq. (7) is also obtained as follows:

$$\sum_{i=1}^{k}\lambda'_i = \frac{N}{N+1}\sum_{i=1}^{n}\lambda_i+\frac{N}{(N+1)^2}\|\boldsymbol{y}-\bar{\boldsymbol{x}}\|^2. \quad (11)$$

Then, the following new accumulation ratio $A'(k)$ is calculated from Eqs. (10) and (11):

$$A'(k) = \frac{N(N+1)\sum_{i=1}^{k}\lambda_i + N\|\boldsymbol{U}_k^T(\boldsymbol{y}-\bar{\boldsymbol{x}})\|^2}{N(N+1)\sum_{i=1}^{n}\lambda_i + N\|\boldsymbol{y}-\bar{\boldsymbol{x}}\|^2} \quad (12)$$

Note that no past samples are needed for the incremental update of $A'(k)$ here.

In the proposed method, the dimensional augmentation is judged from the accumulation ratio $A(k)$. Hence, the new eigenvector matrix $\boldsymbol{U}'$ in Eq. (5) is modified as follows:

$$\boldsymbol{U}' = [\boldsymbol{U},\ \hat{\boldsymbol{h}}]\boldsymbol{R} \quad (13)$$

where

$$\hat{\boldsymbol{h}} = \begin{cases}\boldsymbol{h}/\|\boldsymbol{h}\| & \text{if A(k)} < \theta \\ \boldsymbol{0} & \text{otherwise.}\end{cases} \quad (14)$$

Here, θ is a threshold value.

3 Proposed Learning Scheme

3.1 Incremental Prototype Update for k-NN classifier

As stated in Section 2, IPCA is utilized for reducing the dimensions of input data and constructing an appropriate feature space (i.e., eigenspace) based on an incoming data stream. In IPCA, depending on input data, the following two operations are carried out: eigen-axes rotation and dimensional augmentation of a feature space. On the other hand, ECM can evolve the prototypes which correspond to the representative points in the feature space constructed by IPCA. Hence, when the rotation and dimensional augmentation are carried out, all prototypes must be modified so as to keep the consistency between the old and new eigenspaces.

Let the jth prototype in the current eigenspace $\Omega = (\bar{\boldsymbol{x}}, \boldsymbol{U}, \boldsymbol{\Lambda}, N)$ be $\tilde{\boldsymbol{p}}_j$ $(j = 1, \cdots, L)$ and let the corresponding prototype in the original input space be $\boldsymbol{p}_j$. Here, L is the number of prototypes. For these two prototypes, the following relation holds:

$$\tilde{\boldsymbol{p}}_j = \boldsymbol{U}^T(\boldsymbol{p}_j - \bar{\boldsymbol{x}}). \tag{15}$$

Assume that the $(N+1)$th sample $\boldsymbol{y}$ is added and the eigenspace Ω is updated by $\Omega' = (\bar{\boldsymbol{x}}', \boldsymbol{U}', \boldsymbol{\Lambda}', N+1)$. Substituting Eqs. (1) and (5) into Eq. (15), the updated prototypes $\tilde{\boldsymbol{p}}'_j$ are given as follows [3]:

$$\tilde{\boldsymbol{p}}'_j = \boldsymbol{U}'^T(\boldsymbol{p}_j - \bar{\boldsymbol{x}}') = \boldsymbol{R}^T \begin{bmatrix} \tilde{\boldsymbol{p}}_j \\ \hat{\boldsymbol{h}}^T(\boldsymbol{p}_j - \bar{\boldsymbol{x}}) \end{bmatrix} + \frac{1}{N+1}\boldsymbol{U}'^T(\bar{\boldsymbol{x}} - \boldsymbol{y}). \tag{16}$$

When no dimensional augmentation is needed, $\hat{\boldsymbol{h}} = 0$ holds from Eq. (6). Then, Eq. (16) reduces to

$$\tilde{\boldsymbol{p}}'_j = \boldsymbol{R}^T\tilde{\boldsymbol{p}}_j + \frac{1}{N+1}\boldsymbol{U}'^T(\bar{\boldsymbol{x}} - \boldsymbol{y}) \tag{17}$$

where no information on $\boldsymbol{p}_j$ is needed in the prototype update. However, when the dimensional augmentation as well as the rotation occurs, the original prototypes $\boldsymbol{p}_j$ are necessary for the exact calculation of the new prototype $\tilde{\boldsymbol{p}}'_j$. That is to say, unless we keep the original prototypes in memory, it is impossible to carry out this prototype update.

To do that, we have proposed the approximation for the first term in the right hand side of Eq. (16):

$$\tilde{\boldsymbol{p}}'_j \simeq \boldsymbol{R}^T[\tilde{\boldsymbol{p}}_j^T,\ 0]^T + \frac{1}{N+1}\boldsymbol{U}'^T(\bar{\boldsymbol{x}} - \boldsymbol{y}) \tag{18}$$

where $[\tilde{\boldsymbol{p}}_j^T,\ 0]^T$ is a $(k+1)$-dimensional column vector which is given by adding a zero element to the current prototype $\tilde{\boldsymbol{p}}_j$. This approach is efficient in memory use, but we have to mind the approximation error when the accumulation ratio for the feature space is not so large.

3.2 Learning Algorithm

Let us assume that a small number of training samples are given in advance to form an initial eigenspace. Then, the proposed one-pass incremental learning algorithm is shown below:

Step 0: Calculate the eigenvector matrix $\boldsymbol{U}$ and eigenvalue matrix $\boldsymbol{\Lambda}$ from the covariance matrix of initial training samples. Calculate the projection of all the initial training samples $\boldsymbol{x}_i$ into the eigenspace to obtain the feature vectors $\tilde{\boldsymbol{x}}_i$. Apply ECM (see the details in [6]) to these feature vectors, and obtain the prototypes $\tilde{\boldsymbol{p}}_j$.

Step 1: Apply IPCA to the $(N+1)$th training sample $\boldsymbol{y}$ and update the current eigenspace model $\Omega = (\bar{\boldsymbol{x}}, \boldsymbol{U}, \boldsymbol{\Lambda}, N)$ as follows:

1. Solve an intermediate eigenproblem in Eq. (4) to obtain a rotation matrix $\boldsymbol{R}$ and an eigenvalue matrix $\boldsymbol{\Lambda}'$.
2. Update the accumulation ratio $A'(k)$ based on Eq. (12).
3. Update the mean input vector $\bar{\boldsymbol{x}}'$ and eigenvector matrix $\boldsymbol{U}'$ based on Eqs. (1) and (13), respectively.
4. Increase the total number of training samples N by one.

Step 2: If the dimensional augmentation is not needed in IPCA, update all the current prototypes $\tilde{\boldsymbol{p}}_j$ based on Eq. (17). Otherwise, update them based on Eq. (18).

Step 3: For the training sample $\boldsymbol{y}$, calculate the feature vectors $\tilde{\boldsymbol{y}}$ using the updated eigenvector matrix $\boldsymbol{U}'$ and mean vector $\bar{\boldsymbol{x}}'$ as follows:

$$\tilde{\boldsymbol{y}} = \boldsymbol{U}'^T(\boldsymbol{y} - \bar{\boldsymbol{x}}') \tag{19}$$

Step 4: Apply ECM to the feature vectors $\tilde{\boldsymbol{y}}$, and obtain the updated prototypes $\tilde{\boldsymbol{p}}_{j'}$.

Step 5: Go back to Step 1.

When a query input is presented for classification purpose, the distances to all the prototypes are calculated, and then the k nearest neighbor (k-NN) method can be applied to determine the class. Note that the classification process is carried out on-line during the training of the feature space and prototypes. However, we do not need any modification on the k-NN classifier even if the rotation and augmentation are carried out, because this classifier uses only the distance between a query input and a prototype.

4 Experiments

4.1 Experimental Setup

To investigate the effectiveness of the proposed incremental learning scheme, the performance is evaluated for the three standard datasets in UCI Machine Learning Repository [7]: Segmentation data, Vowel data, and Sonar data. The dataset information is summarized in Table 1.

In the Sonar dataset, the training and test samples are not divided. Hence, we split this dataset into two halves, and the evaluations for the test samples are conducted through two-fold cross-validation. The item 'accuracy' in Table 1 means the highest accuracy shown on the UCI web site [7].

Before the learning starts, first we construct an initial feature space (eigenspace) using a small portion of training samples; that is, these training samples are used for calculating eigenvectors and their eigenvalues through conventional PCA. While the incremental learning is carried out, training samples are randomly drawn from the rest of the training dataset one by one, then the eigenspace is updated by IPCA shown in Section 2. Since the events of incremental learning may not happen at regular time intervals, we use the term *incremental learning stages* instead of the usual time scale. Here, the number of learning stages is equivalent to the number of all training samples that are not used as the initial dataset.

Table 1. Evaluated UCI datasets. The item 'accuracy' means the highest accuracy shown on the UCI web site [7].

name	input dim.	class	train. data	test data	accuracy [%]
Segmentation	19	7	210	2100	-
Vowel	10	11	528	462	56
Sonar	60	2	208	-	83

4.2 Study on Threshold Value η

In the original IPCA, the threshold value η in Eq. (6) is set to zero. However, since the norms of residue vectors are rarely zero in practice, a small value is usually set to η to avoid generating a redundant feature space. As easily expected, if the value is too large, a compact feature space is acquired but the recognition performance may get worse due to the lost of useful information. Generally, it is not easy to find a suitable η and it may be varied depending on the magnitude of input data.

As a preliminary experiment, let us see the influence of η to the recognition performance. Here, Nearest Neighbor (NN) classifier is used for evaluating the recognition performance[1]. The prototypes for NN classifier are trained by ECM in which the same training dataset as in IPCA is used for the training. The training of the feature space and prototypes are conducted based on the procedure shown in 3.2.

Figure 1 shows typical time courses of recognition accuracy, accumulation ratio, and feature space dimensions at each learning stage. In these experiments, the threshold values η are varied from 0.1 to 1.2, and 10% of the entire training samples are used for obtaining initial eigen spaces; that is, the remaining 90% samples are trained one by one in the following incremental stages.

As you can see from Fig. 1, the influences of η to the recognition accuracy and the generated feature space are quite different depending on the datasets. In Sonar data, it seems that the threshold values ($\eta = 0.1, 0.6, 1.2$) greatly influence to the construction of feature spaces. If $\eta = 1.2$, the small dimensional feature space is generated but the recognition accuracy is deteriorated due to the low accumulation ratio. If $\eta = 0.1$, the best accuracy is acquired but the dimensions of the feature space become very large. On the other hand, for Vowel data and Segmentation data, there are less influence of η to both accuracy and feature space dimensions. These results indicate that the threshold value η should be optimized for each dataset.

The proposed method mentioned in 3.2 can be adopted to avoid such a nuisance optimization. In the next experiment, the recognition performance and appropriateness of acquired feature spaces are evaluated for the modified IPCA using the above three UCI datasets.

[1] Although k-NN classifier can also be adopted here, NN classifier outperforms it in our preliminary experiments. Hence, we show only the result of NN classifier.

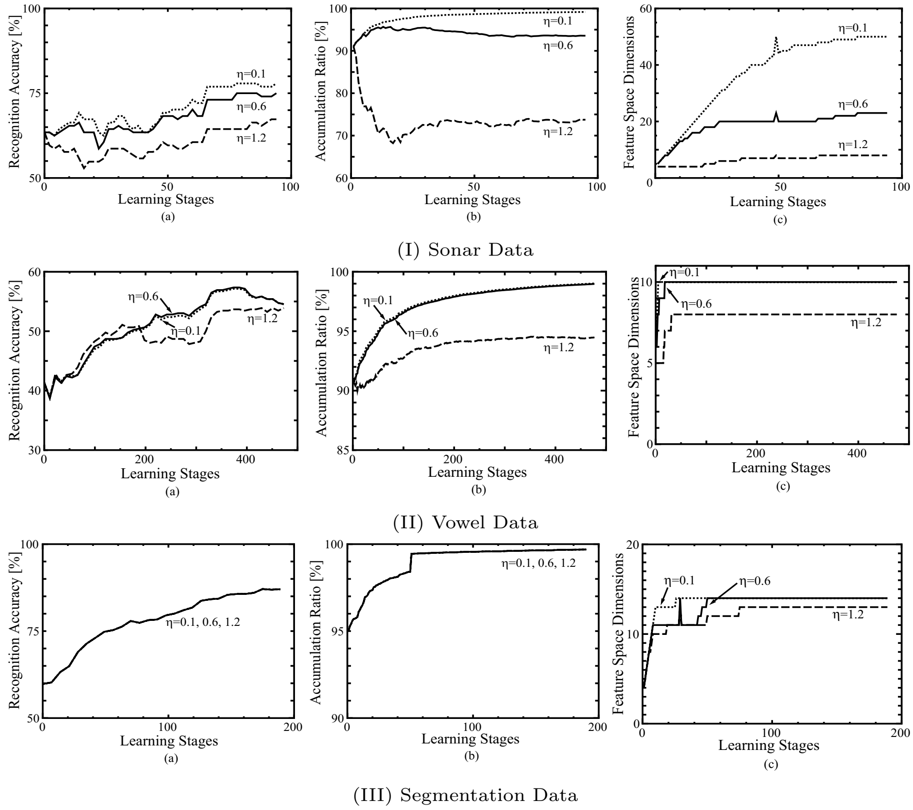

Fig. 1. Typical time courses of (a) recognition accuracy [%], (b) accumulation ratio [%], and (c) feature space dimensions for the three UCI datasets when the original IPCA is applied.

4.3 Evaluation of Proposed IPCA

Even in the proposed IPCA, a threshold value θ in Eq. (14) for the accumulation ratio must be properly given in order to specify how much signal energy should be retained to construct effective feature spaces. To find appropriate threshold values, θ is varied from 0.85 to 0.999 here. In general, the performance of incremental learning depends on the order of giving training samples. Hence, we shall evaluate the performance averaged over ten different learning conditions (i.e., ten different streams of training samples). To see the effectiveness of incremental feature construction, the evaluation for initial feature spaces is also carried out for comparative purposes. More concretely, the eigenvectors are selected only from an initial training set such that the accumulation ration is over 0.999, and the feature space spanned by these eigenvectors is fixed over the entire learning stages but the prototype learning is carried out by ECM.

Table 2. Recognition accuracy [%], accumulation ratio $A(k)$, and dimensions of feature space k at the final incremental learning stage for the three UCI datasets: (a) Sonar data, (b) Vowel, and (c) Segmentation. In PCA, the feature space is calculated only from an initial training set, and then it is fixed over the entire learning stages.

(a) Sonar

	θ=0.85	θ=0.9	θ=0.95	θ=0.999	PCA
Accuracy [%]	77.8	79.4	80.0	79.4	76.2
$A(k)$ [%]	85.6	90.4	95.2	99.9	70.9
k	15.3	19.5	26.5	53.5	9

(b) Vowel

	θ=0.85	θ=0.9	θ=0.95	θ=0.999	PCA
Accuracy (%)	55.4	56.3	57.8	56.0	56.5
$A(k)$ (%)	87.2	92.3	96.6	100	100
k	6.1	7.2	8.2	10	10

(c) Segmentation

	θ=0.85	θ=0.9	θ=0.95	θ=0.999	PCA
Accuracy (%)	79.4	80.9	81.4	87.3	79.5
$A(k)$ (%)	94.3	96.8	97.8	100	86.4
k	4.4	4.6	4.8	8.6	6

Tables 2(a)-(c) show the recognition accuracy, accumulation ratio $A(k)$, and dimensions of feature space k at the final incremental learning stage for the three UCI datasets. In any case, the percentage of initial training samples is set to 10%.

As seen from the results, we can find some threshold values that gives better final recognition accuracy as compared with the results of PCA. Moreover, this final accuracy increases when a large threshold value θ is given. For Sonar data and Vowel data, it seems that there is an optimal value for θ around 0.95. Comparing the feature space dimensions k in these two cases with the previous results in Fig. 1 (see the cases of $\eta = 0.1$ or 0.6), we can see that high-performance compact feature spaces are constructed by the proposed IPCA[2]. It is considered that this result comes from the property of the proposed IPCA; that is, keeping the accumulation ratio at a specified value throughout the learning stages seems to be effective to construct efficient (i.e., low-dimensional) feature spaces.

For Segmentation data, on the other hand, the optimal θ is 0.999 and the accumulation ration $A(k)$ becomes 100%. This result shows that the optimal θ can be different depending on the datasets. However, since we know by experience that there is an optimal value around 0.95 in many cases, we can easily search for the optimal value using the cross-validation. This optimization process is much easier than the search for the optimal value of η in the original IPCA.

[2] Note that the dimensions in Table 2 are averaged over the ten runs, while the result in Fig. 1 is obtained in one of the ten runs.

5 Conclusions and Future Works

In our previous works [2, 3], 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.

In order to construct a proper feature space based on this approach, a suitable threshold value to determine the dimensional augmentation should be given in the IPCA algorithm. This optimization often needs a little annoying process; therefore, the accumulation ratio is introduced into IPCA as its criterion. To implement this approach, first we devised the incremental update algorithm for the accumulation ratio without the past training samples. Next, we presented a new incremental learning scheme for feature space and classifier. From several experiments using the three standard datasets in UCI machine learning repository, we verified that the proposed IPCA worked well without elaborating sensitive parameter optimization and its recognition accuracy outperforms that of the previously proposed learning scheme [3].

There are still several open problems. One is that the computation costs for feature space update could be expensive especially for large dimensional data because the current IPCA algorithm must be applied to each given training sample. To alleviate this problem, we should introduce a batch-mode learning strategy into IPCA. Another problem is that the eigen-features are not always effective for classification purposes. Recently kernel PCA is widely noticed as high-performance features; hence, the extension of incremental learning approach to kernel PCA should be our next research target.

References

1. 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)
2. S. Pang, S. Ozawa, and N. Kasabov, "One-pass Incremental Membership Authentication by Face Classification," *Proc Int. Conf. on Biometric Authentication 2004* (in press)
3. S. Ozawa, S. Pang, and N. Kasabov, "On-line Feature Selection for Adaptive Evolving Connectionist Systems," (submitted)
4. P. Hall and R. Martin, "Incremental Eigenanalysis for Classification," *Proc. British Machine Vision Conference*, vol. 1, pp. 286-295 (1998)
5. N. Kasabov, *Evolving Connectionist Systems: Methods and Applications in Bioinformatics, Brain Study and Intelligent Machines.* Springer-Verlag (2002)
6. 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)
7. http://www.ics.uci.edu/mlearn/MLRepository.html