A Robust Incremental Principal Component Analysis for Feature Extraction from Stream Data with Missing Values

Daijiro Aoki, Toshiaki Omori, and Seiichi Ozawa

Abstract—In this paper, we propose a robust incremental principal component analysis (IPCA) for stream data that can handle missing values on an ongoing basis. In the proposed IPCA, a missing value is substituted with the value estimated from a conditional probability density function. The conditional probability density functions are incrementally updated when new data are given. In the experiments, we evaluate the performance for both artificial and real data sets through the comparison with the two conventional approaches to handling missing values. We first investigate the estimation errors of missing values. The experimental results demonstrate that the proposed IPCA gives lower estimation errors compared to the other approaches. Next, we investigate the approximation accuracy of eigenvectors. The results show that the proposed IPCA has relatively good accuracy of eigenvectors not only for major components but also for minor components.

I. INTRODUCTION

Principal Component Analysis (PCA) is one of the most popular and powerful feature extraction methods in pattern recognition. In PCA, a set of eigenvectors is obtained by solving an eigenvalue problem such that input data are represented by a low-dimensional subspace (eigenspace). However, since PCA needs all the data to calculate an eigenspace, with regards to the memory and computational costs, it is not suitable for learning large-scale high-dimensional stream data which are given continuously over time. For incremental learning purposes, several approaches to Incremental PCA (IPCA) have been proposed so far [1]-[8].

IPCA algorithms are divided into two large groups: iterative approaches [1]-[3] and eigendecomposition approaches [4]-[8]. In the first approach, several principal components are estimated without keeping a covariance matrix through iterative computations. Although this approach is very efficient in memory use, they often suffer from convergence problems especially when input data are given as large dimensional vectors. To solve this problem, Weng et al [2] have proposed a new covariance-free IPCA with good convergence properties based on efficient estimate. However, the computations of principal components in the first approaches are carried out one by one; hence, the approximation errors for high-order components tend to become large. On the other hand, in the second approach, accurate eigenvectors and eigenvalues tend to be obtained because the eigenvalue problem to be solved is incrementally updated by taking all the data given into consideration. Although the eigenvalue problem must be solved whenever new data are given, the computational costs are not heavy in many cases because the matrix size in the eigenvalue problem is restricted to the dimensions of a subspace, which are usually quite smaller than those of input data [4], [5].

On the other hand, in actual environments, sensor measurements could be missing by accident. Since the conventional IPCA cannot handle such missing values, they should be compensated with estimated values before applying to IPCA. It is known that there are three types of missing values: Missing Completely at Random (MCAR), Missing at Random (MAR), and Missing Not at Random (MNAR) [9]. In MCAR, missing values are generated completely at random; that is, the missing of a feature value happens independent of the values of other features. On the other hand, in MAR, missing values of a certain feature could happen depending on the value of another feature. However, eliminating the government by the different feature, missing values are considered to be generated at random. Finally, in MNAR, missing values of a certain feature happens not only by the government of other features but also depending on the value of the same feature.

In this paper, we assume only the MCAR-type of missing values. The conventional methods to handle this type of missing values are listwise deletion and pairwise deletion [10], [11]. In the listwise deletion, an entire data is eliminated from training data even if a single value is missing. Therefore, if there exist many training data with missing values, it becomes difficult to expect good generalization performance. On the other hand, the pairwise deletion only removes the specific missing values from the analysis (e.g. the calculation of the covariance matrix of training data). Since all available training data can be used in the pairwise deletion, this method is useful when the sample size is small or the number of missing values are large. However, if we apply this method to PCA, a covariance matrix has to be kept all the time. If we deal with high-dimensional data, this requires large memory costs. Another way to handle missing data is to replace missing values with the mean value of a variable, called mean substitution. Since missing values are replaced with the same mean, the estimated missing value could have large error from the true value. Therefore, it could lead to the accuracy deterioration in IPCA and the performance deterioration in recognition when using the eigenfeatures.

In this paper, we propose a robust IPCA algorithm in which an imputation method based on a conditional probability density function is introduced to handle missing values. For the notational convenience, let us call this robust IPCA as IPCA with Imputation (IPCA-IM). In IPCA-IM,
missing values are estimated from other variables based on the conditional probability density functions [12], [13], and a missing value is replaced with the expected value obtained from all other variables. In the calculation of a conditional probability density, a covariance matrix is usually needed. However, IPCA does not keep a covariance matrix in the learning of principal components. Therefore, we propose an online method to approximate a covariance matrix using an eigenspace model.

The rest of this paper is organized as follows. Section II gives a brief review of IPCA. In Section III, we first explain the imputation of missing values using conditional probability density functions, and the learning algorithm of the proposed IPCA-IM is described. In Section IV, the estimation accuracy of missing values, the approximation accuracy of obtained vectors, and the recognition accuracy are evaluated. We conduct the experiments for both artificial data sets and real data sets which are selected from the UCI Machine Learning Repository [14].

II. INCREMENTAL PRINCIPAL COMPONENT ANALYSIS

PCA approximates the distribution of input features in a low-dimensional feature space by reducing and rotating axes based on the maximum variance criterion.

Let us consider that a \( k \)-dimensional feature vector \( \mathbf{x} = [x_1, \ldots, x_k]^T \) is transformed into a \( p \) \((p \leq k)\) dimensional feature vector \( \mathbf{y} = [y_1, \ldots, y_p]^T \) as follows:

\[
\mathbf{y} = \mathbf{U}_{kp}^T \mathbf{x}.
\]

Here, \( \mathbf{U}_{kp} = [\mathbf{u}_1, \ldots, \mathbf{u}_p] \) is a \( k \times p \) matrix whose row vector \( \mathbf{u}_i \) \((i = 1, \ldots, p)\) corresponds to an eigenaxis spanning a \( p \)-dimensional subspace.

Assume that a set of \( n \) data \( \mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \) is given. Then, the variance \( \hat{\sigma}^2(\mathbf{U}_{kp}) \) of feature vectors \( \mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n] \) is given by

\[
\hat{\sigma}^2(\mathbf{U}_{kp}) = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{y}_i - \bar{\mathbf{y}})^T (\mathbf{y}_i - \bar{\mathbf{y}}) = \text{tr}(\mathbf{U}_{kp}^T \mathbf{C}_{kk} \mathbf{U}_{kp}),
\]

where \( \bar{\mathbf{x}} \) and \( \bar{\mathbf{y}} \) are the mean vectors of \( \mathbf{X} \) and \( \mathbf{Y} \), respectively. \( \mathbf{C}_{kk} \) is the covariance matrix of \( \mathbf{X} \) whose definition is given by

\[
\mathbf{C}_{kk} = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T.
\]

To find \( \mathbf{U}_{kp} \) that maximizes the variance \( \hat{\sigma}^2(\mathbf{U}_{kp}) \) under the orthonormal constraint for \( \mathbf{U}_{kp} \), it is known that the following objective function \( J(\mathbf{U}_{kp}) \) should be maximized:

\[
J(\mathbf{U}_{kp}) = \text{tr}(\mathbf{U}_{kp}^T \mathbf{C}_{kk} \mathbf{U}_{kp}) - \text{tr}((\mathbf{U}_{kp}^T \mathbf{U}_{kp} - \mathbf{I}) \mathbf{A}_{pp}).
\]

Calculating the derivative of Eq. (4) and setting it to zero, the following equation is obtained.

\[
2\mathbf{C}_{kk} \mathbf{U}_{kp} - 2\mathbf{U}_{kp} \mathbf{A}_{pp} = 0
\]

\[
\iff \quad \mathbf{C}_{kk} \mathbf{U}_{kp} = \mathbf{U}_{kp} \mathbf{A}_{pp}
\]

Eigenaxes \( \mathbf{U}_{kp} \) are obtained by solving Eq. (5) and it is known that the following relation between the variances \( \hat{\sigma}^2(\mathbf{U}_{kp}) \) and the sum of eigenvalues \( \lambda_i \) is held:

\[
\max(\hat{\sigma}^2(\mathbf{U}_{kp})) = \max(\text{tr}(\mathbf{U}_{kp}^T \mathbf{C}_{kk} \mathbf{U}_{kp})) = \max(\text{tr} \mathbf{A}_{pp}) = \sum_{i=1}^{p} \lambda_i.
\]

As seen in Eq. (6), orthogonal vectors \( \mathbf{U}_{kp} \) that maximizes the variance of features are equivalent to the eigenvectors obtained by the eigenvalue decomposition of \( \mathbf{C}_{kk} \).

On the other hand, IPCA updates a feature subspace by rotating eigenaxes \( \mathbf{U}_{kp} \) and adding new eigenaxes if needed. The update of eigenaxes should not require to keep all data learned before. Let us consider the situation that \( n \) data were already trained and a new data \( \mathbf{x}_{n+1} \) is given. The residual vector \( \mathbf{h} \) of \( \mathbf{x}_{n+1} \) is calculated as follows:

\[
\mathbf{h} = (\mathbf{x}_{n+1} - \bar{\mathbf{x}}) - \mathbf{U}_{kp} \mathbf{g}
\]

where

\[
\mathbf{g} = \mathbf{U}_{kp}^T (\mathbf{x}_{n+1} - \bar{\mathbf{x}}).
\]

The new mean vector \( \bar{\mathbf{x}}' \) and the new covariance matrix \( \mathbf{C}_{kk}' \) are given by

\[
\bar{\mathbf{x}}' = \frac{n \bar{\mathbf{x}} + \mathbf{x}_{n+1}}{n + 1}
\]

\[
\mathbf{C}_{kk}' = \frac{n}{n + 1} \mathbf{C}_{kk} + \frac{n}{(n + 1)^2} \mathbf{x}_{n+1} \mathbf{x}_{n+1}^T
\]

where \( \mathbf{x}_{n+1}' = \mathbf{x}_{n+1} - \bar{\mathbf{x}}' \). Using \( \mathbf{C}_{kk}' \) and \( \mathbf{U}_{kp} \), Eq. (5) is rewritten by

\[
\mathbf{C}_{kp} \mathbf{U}_{kp}' = \mathbf{U}_{kp}' \mathbf{A}_{pp}'.
\]

If a new eigenaxis \( \mathbf{h} \) is added, new \( p + 1 \) eigenaxes \( \mathbf{U}_{k,p+1}' \) are obtained by

\[
\mathbf{U}_{k,p+1}' = [\mathbf{U}_{kp}, \hat{\mathbf{h}}] \mathbf{R}_{p+1,p+1}.
\]

Substituting Eqs. (10) and (12) into Eq. (11), we obtain the following eigenvalue problem to be solved in IPCA:

\[
[\mathbf{U}_{kp}, \hat{\mathbf{h}}]^T \left( \frac{n}{n + 1} \mathbf{C}_{kk} + \frac{n}{(n + 1)^2} \mathbf{x}_{n+1}' \mathbf{x}_{n+1}'^T \right) \mathbf{U}_{kp}, \hat{\mathbf{h}} \times \mathbf{R}_{p+1,p+1} = \mathbf{R}_{p+1,p+1} \mathbf{A}_{p+1,p+1}.
\]

This is the to be solved when a new eigenaxis is added. Namely, it reduces dimensions assuming that eigenvalues whose numbers are larger than \( \lambda_{p+2} \) can be ignored. Equation (13) can be transformed as:

\[
\left\{ \begin{array}{c}
\frac{n}{n + 1} \mathbf{A}_{pp} \\
0
\end{array} \right\} + \frac{n}{(n + 1)^2} \left[ \begin{array}{c}
\mathbf{g}^T \\
\gamma^T \\
\gamma^2
\end{array} \right] \times \mathbf{R}_{p+1,p+1} = \mathbf{R}_{p+1,p+1} \mathbf{A}_{p+1,p+1}
\]

450
where $\gamma = \hat{h}^T x_{n+1}'$. Here, we use the approximation

$$C_{kk} \approx U_{kp} A_{pp} U_{kp}^T$$

(15)

and the fact that $\hat{h}$ is orthogonal to all eigenaxes. By solving this eigenvalue problem, the rotation matrix $R_{p+1,p+1}$ and corresponding eigenvalues $\Lambda'_{(p+1)(p+1)}$ are obtained. Then, the update of $U_{kp}$ is carried out by Eq. (12).

The accumulation ratio is used to judge whether a new eigenaxis should be added. It is the ratio of amount of information between eigenaxes and the original feature space. It is defined by

$$A(p) = \frac{\sum_{i=1}^{p} \lambda_i}{\sum_{i=1}^{p+1} \lambda_i}$$

(16)

Using $\lambda_i = \tilde{\sigma}_i^2$, this can be rewritten as follows:

$$A(p) = \frac{\sum_{i=1}^{p} \tilde{\sigma}_i^2}{\sum_{i=1}^{p+1} \tilde{\sigma}_i^2}$$

(17)

where $\tilde{\sigma}_i^2$ is the variance of features on the $i$th eigenaxis.

Since no past data is assumed to be retained in IPCA, we get the accumulation ratio in following procedure. Let $x_j$ be $j$th training data and $y_j$ be its projection on eigenaxes. Then, the variance on the $i$th eigenaxis is calculated as follows:

$$\tilde{\sigma}_i^2 = \frac{1}{n} \sum_{j=1}^{n} (y_j - \bar{y})^2$$

$$= \frac{1}{n} \sum_{j=1}^{n} (u_i^T x_j - u_i^T \bar{x})^2.$$  

(18)

Now a new data $x_{n+1}$ is given and we consider that $y_i = u_i' x_{n+1}$. Then, the mean value $\bar{y}_{n+1,i}$ and the variance $\tilde{\sigma}_i^2$, are respectively calculated as follows:

$$\bar{y}_{n+1,i} = \frac{ny_i + y_i}{n+1},$$

(19)

$$\tilde{\sigma}_i^2 = \frac{1}{n+1} \left\{ (y_i - \bar{y}_{n+1,i})^2 + \sum_{j=1}^{n} (y_{ij} - \bar{y})^2 \right\}.$$  

(20)

Substituting Eq. (19) into Eq. (20), we get

$$\tilde{\sigma}_i^2 = \frac{n}{n+1} \tilde{\sigma}_i^2 + \frac{n}{(n+1)^2} (y_i - \bar{y})^2.$$  

(21)

Then, the numerator of the accumulation ratio is given by

$$\sum_{i=1}^{p} \lambda_i' = \frac{n}{n+1} \sum_{i=1}^{p} \lambda_i + \frac{n}{(n+1)^2} \|y_{n+1}\|^2$$

(22)

In the same way, the denominator of the accumulation ratio is calculated as follows:

$$\sum_{i=1}^{k} \lambda_i' = \sum_{i=1}^{k} \tilde{\sigma}_i^2$$

$$= \frac{n}{n+1} \sum_{i=1}^{k} \lambda_i + \frac{n}{(n+1)^2} \|x_{n+1} - \bar{x}\|^2.$$  

(23)

Algorithm 1 IPCA

**Input:** A set of $n$ training data $X = [x_1, \cdots, x_n] \in R^{k \times n}$, threshold of accumulation ratio $\theta$.

1: Perform PCA for $X$ and calculate eigenvectors $U_{kk}$.
2: Obtain $U_{kp}$ such that $A_c(p) \geq \theta$ satisfies.
3: loop
4: Input: A new training data $x_{n+1}$.
5: $n \leftarrow n + 1$.
6: Update the accumulation ratio by Eq. (24).
7: if $A'(p) < \theta$ then
8: Add a new eigenaxis $h$ by Eq. (25).
9: $p \leftarrow p + 1$.
10: end if
11: Solve the eigenvalue problem in Eq. (14) to obtain the rotation matrix $R$ and the eigenvalue matrix $\Lambda'$.
12: Update eigenaxes $U_{kp}$ by Eq. (12).
13: end loop

Then, the denominator of the coefficient is obtained as follows:

$$A(p) = \frac{\sum_{i=1}^{p} \lambda_i + \frac{1}{n+1} \|y_{n+1}\|^2}{\sum_{i=1}^{p} \lambda_i + \frac{1}{n+1} \|x_{n+1} - \bar{x}\|^2}.$$  

(24)

Based on the accumulation ratio $A(p)$, the condition of adding a new eigenaxes is given by

$$\hat{h} = \left\{ \begin{array}{ll} \frac{h}{\|h\|} & \text{if } A(p) < \theta, \\ 0 & \text{otherwise} \end{array} \right.$$  

(25)

where $\theta$ is the threshold. The algorithm of IPCA is summarized in Algorithm1.

III. ESTIMATION OF MISSING VALUES BY IMPUTATION

As discussed in Section I, the list-wise deletion, the pair-wise deletion, and the mean substitution are not suitable for our purpose. In the proposed IPCA, to estimate missing values, we adopt the imputation approach based on the conditional probability density function (CPDF).

A. Preparation

Assume that a new data $x_{n+1}$ includes missing values. Without the loss of generality, we assume that the first $l$ values in $x_{n+1}$ are not missing and the remaining $m$ values are missing. Let $x_{n+1}^A$ be a sub vector with $l$ values and $x_{n+1}^B$ be a sub vector with $m$ missing values. Then, a new data $x_{n+1}$ is represented as follows:

$$x_{n+1} = [x_{n+1}^A, x_{n+1}^B]^T$$

(26)

where

$$x_{n+1}^A = [x_{n+1,1}^A, \cdots, x_{n+1,l}^A]^T$$

(27)

$$x_{n+1}^B = [x_{n+1,l+1}^B, \cdots, x_{n+1,m}^B]^T.$$  

(28)
Let \( \bar{x} \) be the mean vector of \( n \) data \( X = \{ x_j \}_{j=1}^n \). Then, \( \bar{x} \) is also divided into two subvectors \( \bar{x}^A \) and \( \bar{x}^B \) as follows:

\[
\bar{x} = [\bar{x}^A]^T, [\bar{x}^B]^T]^T,
\]

where \( \bar{x}^A \) is an \( l \)-dimensional subvector whose values come from the first \( l \) values of \( \bar{x} \) and \( \bar{x}^B \) is an \( m \)-dimensional subvector whose values come from the remaining \( l \) values of \( \bar{x} \).

In the same manner, the covariance matrix \( C_{kk} \) of \( X \) is divided into the following four matrices:

\[
C_{kk} = \begin{bmatrix}
C_{ll} & C_{lm} \\
C_{ml} & C_{mm}
\end{bmatrix}
\]

where

\[
C_{ll} = \frac{1}{n} \sum_{j=1}^{n} \bar{x}_j^A \bar{x}_j^A^T, \quad C_{lm} = \frac{1}{n} \sum_{j=1}^{n} \bar{x}_j^A \bar{x}_j^B^T
\]

\[
C_{ml} = \frac{1}{n} \sum_{j=1}^{n} \bar{x}_j^B \bar{x}_j^A^T, \quad C_{mm} = \frac{1}{n} \sum_{j=1}^{n} \bar{x}_j^B \bar{x}_j^B^T
\]

\[
\hat{x}_j^A = x_j^A - \bar{x}^A, \quad \hat{x}_j^B = x_j^B - \bar{x}^B.
\]

### B. Missing-Value Estimation Based on CPDF

The CPDF of \( x_{n+1}^B \) conditioned with \( x_{n+1}^A \) is calculated as follows:

\[
P(x_{n+1}^B | x_{n+1}^A) = \frac{P(x_{n+1}^B, x_{n+1}^A)}{P(x_{n+1}^A)}.
\]

Based on this equation, we estimate missing values. Assume that \( x_{n+1}^A \) is subject to a multi-dimensional normal distribution. Now we consider \( \tilde{z}_{n+1} = x_{n+1}^A - \bar{x} \). Then, the probability density function of \( x_{n+1}^A \) is given by

\[
q(x_{n+1}; \bar{x}, \hat{C}_{kk}) = \frac{1}{(2\pi)^{1/2}|C_{kk}|^{1/2}} \exp \left\{ -\frac{1}{2} \tilde{z}^T C_{kk}^{-1} \tilde{z} + \frac{1}{2} \tilde{z}^T \tilde{z} \right\}
\]

where \( \bar{x} \) and \( \hat{C}_{kk} \) are the real mean vector, the real covariance matrix, respectively. By solving the likelihood equation, maximum likelihood estimators of \( \bar{x} \) and \( \hat{C}_{kk} \) are obtained as follows:

\[
\bar{x}' = \frac{1}{n} \sum_{i=1}^{n} x_i,
\]

\[
\hat{C}_{kk}' = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x}')(x_i - \bar{x}')^T.
\]

They are equal to \( \bar{x} \) and \( \hat{C}_{kk} \). Let us consider \( \tilde{z}_{n+1} = x_{n+1}^A - \bar{x} \). Using \( \bar{x} \) and \( \hat{C}_{kk} \), Eq. (34) is reduced to

\[
q(x_{n+1}; \bar{x}, \hat{C}_{kk}) = \frac{1}{(2\pi)^{1/2}|\hat{C}_{kk}|^{1/2}} \exp \left\{ -\frac{1}{2} \tilde{z}^T \hat{C}_{kk}^{-1} \tilde{z} + \frac{1}{2} \tilde{z}^T \tilde{z} \right\}
\]

Let \( P_{kk} \) be an inverse matrix of \( C_{kk}^{-1} \). Then Eq. (37) is transformed as follows:

\[
q(x_{n+1}; \bar{x}, P_{kk}) = \frac{|P_{kk}|^{1/2}}{(2\pi)^{1/2}} \exp \left\{ -\frac{1}{2} \tilde{z}^T P_{kk} \tilde{z} + \frac{1}{2} \tilde{z}^T \tilde{z} \right\}.
\]

\( P_{kk} \) is a precision matrix that represents direct dependence among different variables of training data. We define a block matrix of \( P_{kk} \) as follows:

\[
P_{kk} = \begin{bmatrix}
P_{ll} & P_{lm} \\
P_{ml} & P_{mm}
\end{bmatrix} = \begin{bmatrix}
P_{AA} & P_{AB} \\
P_{BA} & P_{BB}
\end{bmatrix}.
\]

The following joint distribution of \( x_{n+1}^A \) is given by taking the integral of Eq. (38) with regards to \( x_{n+1}^B \).

\[
P(x_{n+1}^A) = \frac{|P_{kk}|^{1/2}|P_{BB}|^{-1/2}}{(2\pi)^{1/2}} \exp \left\{ -\frac{1}{2} \tilde{z}^T P_{kk} \tilde{z} + \frac{1}{2} \tilde{z}^T \tilde{z} \right\}.
\]

Calculating the integral, the following equation is obtained.

\[
P(x_{n+1}^A) = \frac{|P_{BB}|^{1/2}}{(2\pi)^{1/2}} \exp \left\{ -\frac{1}{2} \tilde{z}^T P_{BB} \tilde{z} \right\}.
\]

By Eq. (38) and (41), Eq. (33) is transformed as follows:

\[
P(x_{n+1}^A | x_{n+1}^B) = \frac{|P_{BB}|^{1/2}}{(2\pi)^{1/2}} \exp \left\{ -\frac{1}{2} \tilde{z}^T P_{BB} \tilde{z} \right\}.
\]

We use \( x_{n+1}^{B, CPDF} \) for the estimation of missing values.

To calculate \( x_{n+1}^{B, CPDF} \), the covariance matrix \( C_{kk} \) is used. However, it is wasteful to retain \( C_{kk} \) from the point of view of memory cost, especially when \( k \) is large. Hence, we approximate it by Eq. (15). Let us denote this approximated covariance matrix by \( \hat{C}_{kk} \). In this case, the rank of \( \hat{C}_{kk} \) is lower than that of the real covariance matrix \( C_{kk} \). \( rank(\hat{C}_{kk}) \leq rank(C_{kk}) = k \). Therefore, \( det(\hat{C}_{kk}) = 0 \), and the approximated precision matrix \( \hat{P}_{kk} = \hat{C}_{kk}^{-1} \) cannot be calculated. We explain how to deal with this problem in the following.

When additional training data with missing values \( x_{n+1}^A \) is observed, IPCA-IM makes another eigenaxes \( U_{kk} = [U_{kp}, U_{kk-kp}] \), \( U_{kk-kp} \) is a set of axes that are orthogonal to eigenaxes \( U_{kk-kp} \), and their eigenvalues are all \( 1 \) (\( k_1, \ldots, k_{k-p} \)). By using \( U_{kk-kp} \) and \( A_{kk-kp} \), the approximation of the covariance matrix \( C_{kk} \) is calculated as follows:

\[
\hat{C}_{kk} = \hat{C}_{kk}' = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})'(x_i - \bar{x})'.
\]

After this, the approximation of the precision matrix \( \hat{P}_{kk} \) is given by

\[
\hat{P}_{kk} = \hat{C}_{kk}^{-1}.
\]

We divide \( \hat{P}_{kk} \) in Eq. (45) as we divide Eq. (39) into four small matrices and substitute this into Eq. (43). The algorithm of IPCA-IM is summarized in Algorithm 2.
Algorithm 2 IPCA-IM

Input: A set of \( n \) training data \( \mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{k \times n} \), threshold of accumulation ratio \( \theta \).
1: Perform PCA for \( \mathbf{X} \) and calculate eigenvectors \( \mathbf{U}_{kk} \).
2: Obtain \( \mathbf{U}_{kp} \) such that \( \mathbf{A}_i(p) \geq \theta \) satisfies.
3: loop
4: Input: A new training data \( \mathbf{x}_{n+1} \).
5: if \( \mathbf{x} \) includes missing values then
6: Substitute missing values by their estimations calculated by Eq. (43).
7: end if
8: \( n \leftarrow n + 1 \).
9: Update the accumulation ratio by Eq. (24).
10: if \( \mathbf{A}_i'(p) < \theta \) then
11: Add a new eigenaxis \( \mathbf{h} \) by Eq. (25).
12: \( p \leftarrow p + 1 \).
13: end if
14: Solve the eigenvalue problem in Eq. (14) to obtain the rotation matrix \( \mathbf{R} \) and the eigenvalue matrix \( \Lambda' \).
15: Update eigenaxes \( \mathbf{U}_{kp} \) by Eq. (12).
16: end loop

IV. EXPERIMENTS

As mentioned in Section I, in the proposed IPCA with IMputation (IPCA-IM), we assume that data are generated from a unimodal normal distribution. To see the robustness of IPCA-IM under practical situations, we evaluate the performance when data are not subject to a unimodal normal distribution. For this purpose, we generate artificial data sets whose input variables have different correlations and whose data distribution is subject to a multi-modal distribution. In addition, we evaluate the performance against practical data sets selected from the University of California at Irvine (UCI) Machine Learning Repository [14]. The performance is evaluated through the comparison to the following IPCA algorithms:

1) IPCA with List-wise deletion (IPCA-LW)
2) IPCA with Mean Substitution (IPCA-MS).

In IPCA-LW, data is eliminated from training data even if a single value is missing, while a missing value is replaced with the mean value of other valuables in IPCA-MS.

A. Influence of Data Distributions on Estimation Error

In this subsection, we examine the influence of data distributions on the estimation errors of missing values. For this purpose, we artificially generate data from the two types of distributions: Multivariate Normal Distribution (MND) and Multivariate Normal Mixture Distribution (MNMD). The information on artificial data sets are shown in Table I. For MNMD, we further define two types of distributions: MNMD1 and MNMD2. The details of these distributions are mentioned below.

For MND, data are drawn from the following probability density function:

\[
f(\mathbf{x}; \bar{x}, \mathbf{C}) = \frac{1}{(2\pi)^{p/2}|\mathbf{C}|^{1/2}} \exp \left( -\frac{1}{2} (\mathbf{x} - \bar{x})^T \mathbf{C}^{-1} (\mathbf{x} - \bar{x}) \right)
\]  

(46)

where \( \bar{x} = \mathbf{x} - \bar{x} \) and \( \bar{x} \) is the mean vector of \( p \) input variables \( \mathbf{x} \). Here, \( \mathbf{C} \) is the following correlation matrix:

\[
\mathbf{C} = \begin{pmatrix}
1 & \cdots & \cdots & \cdots & c_{1p} \\
\vdots & \ddots & \cdots & \cdots & \vdots \\
\vdots & \cdots & c_{ij} & \cdots & \vdots \\
\vdots & \cdots & \cdots & \cdots & \vdots \\
c_{p1} & \cdots & \cdots & \cdots & 1
\end{pmatrix}
\]  

(47)

where \( c_{ij} (i \neq j) \) means the correlation between the \( i \)th and \( j \)th variables. In this experiment, we set \( p = 8 \) and the variable correlation \( c_{ij} \) is randomly selected within the different ranges [0, 0.1] through [0.9, 1.0].

For MNMD1 and MNMD2, data are drawn from the following multi-modal probability density function defined as a mixture of two normal distributions in Eq. (46):

\[
F(\mathbf{x}; \bar{x}_1, \bar{x}_2, \mathbf{C}) = \sum_{i=1}^{2} \pi_i f(\mathbf{x}; \bar{x}_i, \mathbf{C})
\]  

(48)

where \( \pi_i \) is a weight of \( f(\mathbf{x}; \bar{x}_i, \mathbf{C}) \); \( \bar{x}_1 \) and \( \bar{x}_2 \) are mean vectors. In this experiment, we set \( \pi_i = 0.5 \) and the distance between \( \bar{x}_1 \) and \( \bar{x}_2 \) is controlled to generate different multi-modal distributions. For MNMD1, the variable correlation \( c_{ij} \) is randomly selected within the different ranges [0, 0.1] through [0.9, 1.0] and the mean distance is fixed at 2. For MNMD2, \( c_{ij} \) is randomly selected within a fixed range [0.9, 1.0] and the mean distance is changed from 1 to 10. In all the data sets, a half of incremental data include missing values and the percentages of missing data are changed from 10 to 100 (%).

To evaluate the performance, we adopt the following mean absolute error as the estimation error of missing values:

\[
E = \frac{1}{M} \sum_{j=1}^{M} \| \hat{x}_j - x_j \|
\]  

(49)

where \( M \) is the number of data with missing values, \( \hat{x}_j \) is the \( j \)th data without missing values, and \( x_j \) is the data whose missing values are substituted with estimated values.

Figure 1 shows the estimation errors \( E \) of IPCA-MS and the proposed IPCA-IM when the variable correlations and the mean distances are changed in MND and MNMD. As
shown in Figs. 1 (a) and (b), the estimation error in IPCA-MS is not affected by the variable correlations, while the error in IPCA-IM is decreased for both MND and MNMD when the correlation becomes high. This result implies that the imputation of missing values in IPCA-IM works well by estimating a missing value based on the correlatedness to other variables. On the other hand, as seen in Fig. 1 (c), the mean distance gives a great influence on the estimation errors in IPCA-MS. However, the estimation error in IPCA-IM is not affected by the mean distance. This result also supports the robustness of the proposed IPCA-MS for multi-modal distributions.

B. Performance Evaluation for Real Data

In this subsection, four UCI data sets are used for the performance evaluation. The dataset information is shown in Table II. Randomly selected 10% of training data are used for initial training, and the rest of data are used for incremental learning except for Vowel-context. For Vowel-Context, 20% of training data are used for initial training and the rest of data are used for incremental learning. We assume that a half of training data include 10 to 100% missing values.

In the following experiments, we adopt the three performance scales: the estimation errors $E$ of missing values in Eq. (49), the average similarities between eigenvectors, and the recognition accuracies. The performance is compared among the three IPCA algorithms: IPCA-LW, IPCA-MS, and IPCA-IM. The average similarities $d_i$ are calculated by the
Fig. 2. Average similarities between eigenaxes obtained by the three IPCA algorithms and usual IPCA with complete additional data: (a) results for Landsat data set (b) results for Adult data set (c) results for Vowel-context data set (d) results for Letter Recognition data set

following direction cosine $d_i$:

$$d_i = \frac{1}{L} \sum_{j=1}^{L} u_{ji}^T u_{ji}$$

(50)

where $L$ is the number of trials to obtain the average similarity; $u_{ji}$ and $u_{ji}^*$ are the $i$th eigenvectors for the data with and without missing values, respectively. The recognition accuracies are evaluated when the nearest neighbor classifier, where initial training data are used as prototypes, is adopted. The threshold of the accumulation ratio is automatically determined by the 10-fold cross validation for the initial training data.

Table III shows the estimation errors of missing values in IPCA-MS and IPCA-IM. Compared to IPCA-MS, IPCA-IM gives good estimation for missing values except for Adult data. The degradation in missing value estimation for Adult data may originate from the binominal property in the data distribution.

<table>
<thead>
<tr>
<th></th>
<th>IPCA-MS</th>
<th>IPCA-IM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Landsat</td>
<td>0.58 ± 0.03</td>
<td>0.16 ± 0.02</td>
</tr>
<tr>
<td>Adult</td>
<td>0.52 ± 0.03</td>
<td>0.63 ± 0.07</td>
</tr>
<tr>
<td>Vowel-context</td>
<td>0.59 ± 0.02</td>
<td>0.48 ± 0.02</td>
</tr>
<tr>
<td>Letter Recognition</td>
<td>0.55 ± 0.01</td>
<td>0.41 ± 0.01</td>
</tr>
</tbody>
</table>

Figure 2 show the average similarities between eigenvectors obtained by the above methods and IPCA. For the Landsat, IPCA-IM constructs an accurate eigenspace compared to IPCA-LW and IPCA-MS. For the Adult, the Vowel-context and the Letter Recognition, however, there are no difference between the eigenspaces obtained by IPCA-LW, IPCA-MS and IPCA-IM.

Table IV shows the recognition accuracy (%) of IPCA-LW, IPCA-MS, and IPCA-IM. From the results, we can say that the recognition accuracies are not seriously affected by
how missing values are dealt with; that is, the estimation of missing values do not contribute to the improvement in the recognition accuracy, even though the estimation errors of missing values are relatively improved. The reason for this is now investigated.

V. CONCLUSIONS AND FUTURE WORK

We propose a robust incremental principal component analysis (IPCA) that can handle missing values in stream data. In the proposed IPCA, an imputation approach to handling missing values is introduced, where the missing value is estimated from the conditional probability density function of variables. When new data are given, the conditional probability density function is also updated incrementally; therefore, the missing value estimation is conducted on an ongoing basis.

In the experiments, we first investigate the accuracy of missing value estimation for the artificial data, which are generated from both unimodal and multi-modal distributions. We control the shape of distributions by changing the variable correlations and the mean distance of two normal distributions. The result demonstrates that the proposed IPCA gives better estimation accuracy compared to the IPCA algorithms with the two conventional methods for handling missing values: IPCA with list-wise deletion (IPCA-LW), and IPCA with mean substitution (IPCA-MS).

We also conduct the experiments for real data sets. With regards to the estimation error of missing values, the proposed IPCA has better performance. Next, we investigate the approximation accuracy of eigenvectors obtained by the proposed IPCA. From the experimental results, we confirm that the proposed IPCA has relatively good approximation accuracy of eigenvectors not only for major components but also for minor components. Although the proposed IPCA gives better estimate of missing values, the recognition performance for the eigenfeatures extracted by the proposed IPCA has little difference from the performances for IPCA-LW and IPCA-MS. The reason for this is now investigating.

TABLE IV
RECOGNITION ACCURACY (%)

<table>
<thead>
<tr>
<th></th>
<th>IPCA-LW</th>
<th>IPCA-MS</th>
<th>IPCA-IM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Landsat</td>
<td>80.8 ± 1.9</td>
<td>80.7 ± 1.9</td>
<td>80.8 ± 1.9</td>
</tr>
<tr>
<td>Adult</td>
<td>74.8 ± 2.1</td>
<td>74.8 ± 2.1</td>
<td>74.8 ± 2.1</td>
</tr>
<tr>
<td>Vowel-context</td>
<td>47.5 ± 3.7</td>
<td>47.5 ± 3.7</td>
<td>47.5 ± 3.7</td>
</tr>
<tr>
<td>Letter Recognition</td>
<td>76.9 ± 0.71</td>
<td>76.9 ± 0.71</td>
<td>76.9 ± 0.71</td>
</tr>
</tbody>
</table>

REFERENCES