3-1-2015

Integrating Data Transformation in Principal Components Analysis

Mehdi Maadooliat
Marquette University

Jianhua Z. Huang
Texas A & M University - College Station

Jianhua Hu
University of Texas M.D. Anderson Cancer Center

Accepted version. *Journal of Computational and Graphical Statistics*, Vol. 24, No. 1 (March 2015): 84-103. DOI.

Integrating Data Transformation in Principal Components Analysis

Mehdi Maadooliat, Jianhua Z. Huang and Jianhua Hu*

Abstract

Principal component analysis (PCA) is a popular dimension reduction method to reduce the complexity and obtain the informative aspects of high-dimensional datasets. When the data distribution is skewed, data transformation is commonly used prior to applying PCA. Such transformation is usually obtained from previous studies, prior knowledge, or trial-and-error. In this work, we develop a model-based method that integrates data transformation in PCA and finds an appropriate data transformation using the maximum profile likelihood. Extensions of the method to handle functional data and missing values are also developed. Several numerical algorithms are provided for efficient computation. The proposed method is illustrated using simulated and real-world data examples.

Keywords: Functional PCA; Missing data; PCA; Profile likelihood; Transformation model

1 Introduction

Principal component analysis (PCA) and its extensions are commonly used dimension reduction techniques that transform a collection of correlated variables into a small number of uncorrelated variables called principal components. Two common approaches of motivating the PCA are (a) finding a small number of linear combinations of the variables that account for most of the variance in the observed data (Hotelling, 1933); and (b) obtaining best low-rank matrix approximation of the data matrix (Pearson, 1901; Jolliffe, 2002, Section 3.5). These two

*Mehdi Maadooliat (mehdi@mscs.mu.edu) is Assistant Professor, Department of Mathematics, Statistics and Computer Science, Marquette University, WI; Jianhua Z. Huang is Professor, Department of Statistics, Texas A&M University, College Station, TX; and Jianhua Hu is Associated Professor, Department of Biostatistics, Division of Quantitative Sciences, The University of Texas MD Anderson Cancer Center, Houston, TX.

approaches boil down respectively to computation of the spectral decomposition of the sample covariance matrix and the singular value decomposition (SVD) of the data matrix, and give the same result. PCA has also been used as an important tool for unsupervised functional data analysis (Ramsay and Silverman, 2005). In the functional PCA (FPCA), some regularization is needed to take into account the underlying smoothness of the functional data. Rice and Silverman (1991) and Silverman (1996) presented two alternative formulations of FPCA by regularized variance maximization, and Huang et al. (2008) formulated the FPCA using penalized low-rank approximation.

It is known that the standard PCA may not be the suitable technique to apply when the data distribution is skewed or there are outliers. Several extensions of PCA have been developed to handle such situations; see, for example, Croux and Ruiz-Gazen (2005), Higuchi and Eguchi (2004), Hubert et al. (2002), Locantore et al. (1999), Maronna (2005) and Hubert et al. (2009). Alternatively, data transformations have been used prior to applying PCA and functional PCA (e.g., Hu et al., 2006; Huang et al., 2008). Data transformations have also been applied when investigating the structure of the covariance matrix (e.g., Zimmerman and Núñez Antón, 2009), which is highly related to PCA. However, the choice of the transformation usually comes from previous studies, prior knowledge, or trial-and-error. The goal of this paper is to develop an automatic, data-driven method for finding an appropriate data transformation and obtaining the principal components simultaneously.

We make use of the connection of PCA and a probabilistic model to obtain such an automatic procedure. Consider the following fixed effects model

$$\mathbf{Y} = \mathbb{U}\mathbb{V}^\top + \mathbf{E}, \tag{1}$$

where $\mathbf{Y}$ is an $n \times m$ data matrix whose (i, j) entry y_{ij} is the i^{th} observation of the j^{th} variable, $\mathbb{U}$ and $\mathbb{V}$ are respectively $n \times d$ and $m \times d$ nonrandom matrices ($d \leq \min(m, n)$), and $\mathbf{E}$ is an $n \times m$ matrix of random errors. We assume that ϵ_{ij}'s, the entries of $\mathbf{E}$, are independent random variables from a normal distribution with mean 0 and constant variance σ^2. For identifiability, it is required that (a) $\mathbb{U}^\top\mathbb{U}$ is a diagonal matrix; and (b) $\mathbb{V}^\top\mathbb{V}$ is the identity matrix. Under model (1), the MLE of $\mathbb{U}$ and $\mathbb{V}$ minimizes the reconstruction error $\|\mathbf{Y} - \mathbb{U}\mathbb{V}^\top\|_F$ where $\|\cdot\|_F$ is the Frobenius norm. According to the formulation of PCA as seeking the best low-rank matrix approximation, the k^{th} column of $\mathbb{V}$ is the k^{th} principal component weight vector and the k^{th} column of $\mathbb{U}$ contains the corresponding principal component (PC) scores. Tipping and Bishop

(1999) proposed a closely related random effects model where $\mathbb{U}$ is assumed to be random and showed that up to a scale change, the MLE of their model is equivalent to the PCA. We adopt the fixed effect model (1) in this paper because it is exactly equivalent to the best low-rank approximation formulation of the PCA.

To incorporate data transformation, we modify the model (1) to

$$f(\mathbf{Y}|\boldsymbol{\eta}) = \mathbb{U}\mathbb{V}^\top + \mathbf{E}, \tag{2}$$

where $f(\cdot|\boldsymbol{\eta})$ is a monotonic function defined over the entries of $\mathbf{Y}$, and $\boldsymbol{\eta}$ is the unknown vector of transformation parameters. Here, we use the notational convention that when a function is applied to a matrix, it is an elementwise operation. The MLE will yield simultaneous estimation of the transformation parameters, the principal component weights and scores. In particular, the transformation parameters can be estimated using the maximum profile likelihood. This model can also easily incorporate functional data. When the rows of $\mathbf{Y}$ represent discretely sampled functions, we can introduce a roughness penalty on each column of $\mathbb{V}$ to ensure the desired smoothness on the functional principal component weight functions and apply the maximum penalized likelihood for parameter estimation.

Since missing observations are often encountered in real applications, another goal of the paper is to extend our integrated approach of PCA with data transformation to handle missing data. Having a probabilistic model, the solution is conceptually simple—we just need to focus on the observed data likelihood. However, computation of the profile likelihood of the transformation parameters is not straightforward. We developed two algorithms to facilitate the computation. One algorithm iteratively imputes the missing data and then resorts to the complete data procedures. It is essentially an implementation of the expectation-maximization (EM) algorithm. The other algorithm is an extension of the power iteration (e.g., Appendix A of Jolliffe, 2002). Both algorithms are also extended to deal with functional data.

The rest of the papers is organized as follows. The details of the proposed methods including computational algorithms are given in Sections 2 and 3, which treat the ordinary and functional data structure respectively. In Section 4, we use simulations and two real datasets to demonstrate the applicability of the proposed methods. Some concluding remarks are given in Section 5. The appendix contains detailed derivations of the algorithms presented in the main text.

2 Ordinary data structure

We present our methods in two consecutive sections; this section focuses on the ordinary data structure and the next section considers the functional data structure.

2.1 Integrating the data transformation to PCA by profile likelihood

Denote $\mathbf{\Theta} = (\boldsymbol{\eta}^\top, \mathrm{vec}(\mathbb{U}), \mathrm{vec}(\mathbb{V}), \sigma^2)^\top$. The log-likelihood function for the transformation model (2) is

$$\ell(\mathbf{\Theta}) \;=\; -\frac{1}{2\sigma^2}||f(\mathbf{Y}|\boldsymbol{\eta}) - \mathbb{U}\mathbb{V}^\top||^2 - \frac{nm}{2}\log(\sigma^2) + \sum_{i=1}^{n}\sum_{j=1}^{m}\big\{\log|f'(y_{ij}|\boldsymbol{\eta})|\big\}. \tag{3}$$

We estimate the transformation parameter $\boldsymbol{\eta}$ by the maximum profile likelihood and then obtain the principal component weight vectors and scores using the MLE when fixing $\boldsymbol{\eta}$ at its MLE. The profile log-likelihood function of $\boldsymbol{\eta}$ has the expression

$$\ell_p(\boldsymbol{\eta}) \;=\; -\frac{1}{2\widehat{\sigma^2_{\boldsymbol{\eta}}}}||f(\mathbf{Y}|\boldsymbol{\eta}) - \widehat{\mathbb{U}}_{\boldsymbol{\eta}}\widehat{\mathbb{V}}_{\boldsymbol{\eta}}^\top||^2 - \frac{nm}{2}\log(\widehat{\sigma^2_{\boldsymbol{\eta}}}) + \sum_{i=1}^{n}\sum_{j=1}^{m}\big\{\log|f'(y_{ij}|\boldsymbol{\eta})|\big\}, \tag{4}$$

where $\widehat{\mathbb{U}}_{\boldsymbol{\eta}}$, $\widehat{\mathbb{V}}_{\boldsymbol{\eta}}$, and $\widehat{\sigma^2}_{\boldsymbol{\eta}}$ are the MLEs when fixing the transformation parameter at $\boldsymbol{\eta}$.

For fixed $\boldsymbol{\eta}$, the MLEs of $\mathbb{U}$ and $\mathbb{V}$ minimize the reconstruction error $\|f(\mathbf{Y}|\boldsymbol{\eta}) - \widehat{\mathbb{U}}_{\boldsymbol{\eta}}\widehat{\mathbb{V}}_{\boldsymbol{\eta}}^\top\|$. The solution can be found by applying the Eckart-Young theorem (or approximation theorem; see Stewart (1993) for its history). The explicit forms of the MLEs of $\mathbb{U}_{\boldsymbol{\eta}}$, $\mathbb{V}_{\boldsymbol{\eta}}$, and $\sigma^2_{\boldsymbol{\eta}}$ are

$$\begin{aligned} \widehat{\mathbb{U}}_{\boldsymbol{\eta}} &= \mathbf{U_d}\mathbf{\Sigma_d}, \qquad \widehat{\mathbb{V}}_{\boldsymbol{\eta}} \;=\; \mathbf{V_d}, \\ \widehat{\sigma^2_{\boldsymbol{\eta}}} &= \frac{1}{nm}||f(\mathbf{Y}|\boldsymbol{\eta}) - \widehat{\mathbb{U}}_{\boldsymbol{\eta}}\widehat{\mathbb{V}}_{\boldsymbol{\eta}}^\top||^2, \end{aligned} \tag{5}$$

where using the SVD of the $f(\mathbf{Y}|\boldsymbol{\eta}) = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, $\mathbf{U_d}$ and $\mathbf{V_d}$ are the first d columns of $\mathbf{U}$ and $\mathbf{V}$, respectively, corresponding to the d largest singular values, recorded in the diagonal matrix, $\mathbf{\Sigma_d}$. We do not need to compute the full SVD; an efficient algorithm for the truncated SVD can be used to speed up the calculation of the leading singular vectors (e.g., Wu and Simon, 2000).

The following is the algorithm for the proposed method of integrating the data transformation and the PCA. This algorithm is referred to as "PCA.t".

1. Start from an initial estimate of $\boldsymbol{\eta}$, denoted by $\boldsymbol{\eta}_0$ ($t = 0$). Usually we pick the initial estimate corresponding to the model with no transformation.

2. At the t^{th} iteration, let $\mathbf{X}_t$ be the $n \times m$ matrix with the $(i,j)^{\text{th}}$ entry $f(y_{ij}|\boldsymbol{\eta}_t)$. Obtain the rank-$d$ truncated SVD of $\mathbf{X}_t$ as $\mathbf{U_d}\mathbf{\Sigma_d}\mathbf{V_d}^\top$ and use equation (5) to obtain $\widehat{\mathbb{U}}_t$, $\widehat{\mathbb{V}}_t$, and $\widehat{\sigma_t^2}$, where $\widehat{\mathbb{U}}_t$, $\widehat{\mathbb{V}}_t$, and $\widehat{\sigma_t^2}$ are $\widehat{\mathbb{U}}_{\boldsymbol{\eta}}$, $\widehat{\mathbb{V}}_{\boldsymbol{\eta}}$, and $\widehat{\sigma_{\boldsymbol{\eta}}^2}$ at $\boldsymbol{\eta} = \boldsymbol{\eta}_t$.

3. Obtain the updated value of $\boldsymbol{\eta}_{t+1}$ via an optimization algorithm to increase the value of the profile log-likelihood function $\ell_p(\boldsymbol{\eta}_t)$ defined in equation (4).

4. Iterate between the last two steps until convergence is reached.

The algorithm may converge to a local maximum. It is advisable to rerun the algorithm several times with different initial values of $\boldsymbol{\eta}_0$ and use the one that gives the maximum value of the profile likelihood.

A variety of optimization techniques, such as the downhill simplex algorithm or gradient-based algorithms (e.g., Avriel, 1976) can be used in Step 3 to maximize the profile likelihood. While the algorithm is general enough to incorporate various transformation families, our implementation has focused on the widely used power (Box-Cox) transformation family (Box and Cox, 1964). The power transform is parameterized by a non-negative parameter β that includes the logarithm, square root, and multiplicative inverse as special cases. The Box-Cox transformation for each element of $\mathbf{Y}$ is defined as

$$f(y_{ij}|\beta) = \begin{cases} \dfrac{y_{ij}^{\beta} - 1}{\beta}\ , & \beta \neq 0, \\ \log(y_{ij})\ , & \beta = 0. \end{cases} \tag{6}$$

We used the L-BFGS-B algorithm (Byrd et al., 1994) in our implementation to maximize the profile likelihood for the univariate parameter in the power transformation family.

2.2 Handling missing data

We give extensions of the algorithm given in the previous subsection when there are missing data. We assume that missing values are "missing at random", that is, the cause of the missing data is unrelated to both the observed values and the missing values. In this case, the sample is an unbiased representation of the population of interest and observed data likelihood can be used for statistical inference (Daniels and Hogan, 2008).

Denote $\mathbf{Y}$ as the complete data matrix including observed and missing data. Define the indicator matrix $\mathbf{I}_m$ such that the $(i,j)^{\text{th}}$ element of $\mathbf{I}_m$ is set to be 1, if the data point has been observed and to be 0, otherwise. We define the indicator matrix $\mathbf{I}_m^c$ to be $\mathbf{1}_m\mathbf{1}_m^\top - \mathbf{I}_m$,

where $\mathbf{1}_m$ is a m-vector of ones. We use the symbol $\odot$ for the Schur product, which is defined as the entrywise product of two matrices of the same dimension. In the presence of missing observations, the observed data log-likelihood function has the following form:

$$\ell(\boldsymbol{\Theta}) = -\frac{1}{2\sigma^2}||\mathbf{I}_m \odot \big(f(\mathbf{Y}|\boldsymbol{\eta}) - \mathbb{U}\mathbb{V}^\top\big)||^2 - \frac{N(\mathbf{I}_m)}{2}\log(\sigma^2) + \sum_{(i,j):\mathbf{I}_m(i,j)=1}\big\{\log|f'(y_{ij}|\boldsymbol{\eta})|\big\}, \tag{7}$$

where $N(\mathbf{I}_m)$ is the number of actual observations. Equation (7) can be seen as an extension of (3) that excludes the missing data and focuses on the remaining observed-data points.

Now, we extend the profile likelihood approach in Section 2.1 to estimate $\boldsymbol{\eta}$. However, for a fixed $\boldsymbol{\eta}$, there is no direct solution like (5) to maximize the given log-likelihood function in the presence of missing observations. We propose two iterative algorithms for maximizing the profile log-likelihood of $\boldsymbol{\eta}$. The first algorithm iteratively imputes the missing data and then apply the SVD of a complete data matrix. The second algorithm avoids imputing the missing observations and updates the singular vectors and the singular values using modified power iterations.

To describe the first algorithm, for fixed transformation parameter $\boldsymbol{\eta}$, define $\mathbf{X}_{\boldsymbol{\eta}}$ to be the complete $n \times m$ data matrix so that the observed elements are $\mathbf{I}_m \odot f(\mathbf{Y}|\boldsymbol{\eta})$ and the missing elements are $\mathbf{I}_m^c \odot \mathbf{X}_{\boldsymbol{\eta}}$. The algorithm starts from zero imputation (replacing the missing observations with zeros) and subsequently replaces the missing values with the associated values from the low-rank approximation of the complete data matrix from the previous iteration step.

Algorithm 1:

1. Set $t \leftarrow 0$, and let $\mathbf{X}_t = \mathbf{I}_m \odot \mathbf{X}_{\boldsymbol{\eta}}$.

2. Repeat the following steps until convergence.

   (a). Obtain the rank-d truncated SVD of the matrix $\mathbf{X}_t$ as $\mathbf{U_d}\mathbf{\Sigma_d}\mathbf{V_d}^\top$

   (b). Define $\mathbf{X}_{t+1} = \mathbf{I}_m \odot \mathbf{X}_{\boldsymbol{\eta}} + \mathbf{I}_m^c \odot (\mathbf{U_d}\mathbf{\Sigma_d}\mathbf{V_d}^\top)$.

   (c). Set $t \leftarrow t + 1$.

3. After convergence, record $\mathbf{U_d}$, $\mathbf{V_d}$, and $\mathbf{\Sigma_d}$ as the output of this algorithm.

This algorithm is not new; Hastie et al. (1999) showed that the maximizer of (7) is a fixed point of this algorithm. Beckers and Rixen (2003) used a similar algorithm to impute the missing observations in incomplete oceanographic data sets. The contribution here is to show this algorithm is essentially an EM algorithm (Dempster et al., 1977) to obtain the MLEs of

$\mathbb{U}$ and $\mathbb{V}$ in presence of missing observations for a fixed $\boldsymbol{\eta}$ (see Appendix for derivation). The Algorithm 1 has the advantage of obtaining the first d components of the SVD simultaneously based on maximizing the complete data log-likelihood function. However, similar to application of the EM in other contexts, the convergence of the algorithm could be very slow.

The second algorithm cyclically updates the singular vectors. In each cycle, we update the k^{th} singular vector (where $1 \leq k \leq d$) based on one iteration of the power algorithm with respect to the associated observed residual matrix (subtract the subspace spanned by the remaining $d-1$ singular vectors from the observed-data matrix). This algorithm does not require imputing missing data. Derivation of the algorithm can be found in the Appendix.

Algorithm 2:

1. Set $t \leftarrow 0$, and obtain the rank-d truncated SVD of the matrix $\mathbf{I}_m \odot \mathbf{X}_{\boldsymbol{\eta}}$ as $\mathbf{U_d}\mathbf{\Sigma_d}\mathbf{V_d}^\top$.

2. Repeat the following steps until convergence.

   (a). For $k = 1, \cdots, d$,

      i. Let $\mathbf{X}_{t,k} = \mathbf{I}_m \odot (\mathbf{X}_{\boldsymbol{\eta}} - \mathbf{U}_{-k}\mathbf{\Sigma}_{-k}\mathbf{V}_{-k}^\top)$.

      ii. Set $\boldsymbol{u}_{t,k} = \{\text{diag}(\mathbf{I}_m\mathbf{V}_{\cdot k}^2)\}^{-1}\mathbf{X}_{t,k}\mathbf{V}_{\cdot k}$, and $\boldsymbol{v}_{t,k} = \{\text{diag}(\mathbf{I}_m^\top\mathbf{U}_{\cdot k}^2)\}^{-1}\mathbf{X}_{t,k}^\top\mathbf{U}_{\cdot k}$.

      iii. Normalize $\boldsymbol{u}_{t,k}$ and $\boldsymbol{v}_{t,k}$ to norm one vectors.

      iv. Update $\mathbf{U}_{\cdot k} = \boldsymbol{u}_{t,k}$, $\mathbf{V}_{\cdot k} = \boldsymbol{v}_{t,k}$ and $\mathbf{\Sigma}_{kk} = \dfrac{(\boldsymbol{u}_{t,k}^\top\mathbf{X}_{t,k}\boldsymbol{v}_{t,k})}{(\boldsymbol{u}_{t,k}^{2\top}\mathbf{I}_m\boldsymbol{v}_{t,k}^2)}$.

   (b). Set $t \leftarrow t + 1$

3. After convergence, record $\mathbf{U_d}$, $\mathbf{V_d}$, and $\mathbf{\Sigma_d}$ as the output of this algorithm.

In this algorithm, $\mathbf{\Sigma}_{-k}$ is a submatrix of $\mathbf{\Sigma_d}$ after removing the k^{th} row and column, $\mathbf{\Sigma}_{kk}$ denotes the $(k,k)^{\text{th}}$ element of the matrix $\mathbf{\Sigma}$, $\mathbf{U}_{\cdot k}$ stands for the k^{th} column of $\mathbf{U}$, $\mathbf{U}_{-k}$ is the submatrix formed by removing the k^{th} column of $\mathbf{U_d}$, and we define $\mathbf{U}_{\cdot k}^2$ as $\mathbf{U}_{\cdot k} \odot \mathbf{U}_{\cdot k}$. We consider similar definitions for $\mathbf{V}_{\cdot k}$, $\mathbf{V}_{-k}$, $\mathbf{V}_{\cdot k}^2$, $\boldsymbol{v}_{t,k}^2$ and $\boldsymbol{u}_{t,k}^2$. Moreover, the operator $\text{diag}(\cdot)$ is defined on vectors, and the output is a diagonal matrix whose diagonal elements equal to the input vector. The latter algorithm is the generalization of the power algorithm based on the observed values to obtain the first d components of SVD sequentially. It is much faster than the first algorithm; mainly due to the fact of omitting the calculation of the truncated SVD in each iteration. Note that, Algorithm 2 can not be easily extended to simultaneously obtain the singular vectors. The reason is that the QR decomposition, which is necessary at the end of

each power iteration (Trefethen and Bau, 1997), does not take into account the missing data structure.

Let $\mathbf{U_d}$, $\mathbf{V_d}$, and $\mathbf{\Sigma_d}$ be the output of either Algorithm 1 or 2 in presence of missing values. Following the model (2) and the log-likelihood function (7), for a fixed parameter $\boldsymbol{\eta}$, the MLEs for the rest of parameters are

$$\begin{aligned} \widehat{\mathbb{U}}_{\boldsymbol{\eta}} &= \mathbf{U_d}\mathbf{\Sigma_d}, & \widehat{\mathbb{V}}_{\boldsymbol{\eta}} &= \mathbf{V_d}, \\ \widehat{\sigma^2_{\boldsymbol{\eta}}} &= \frac{1}{N(\mathbf{I}_m)}||\mathbf{I}_m \odot \big(f(\mathbf{Y}|\boldsymbol{\eta}) - \widehat{\mathbb{U}}_{\boldsymbol{\eta}}\widehat{\mathbb{V}}_{\boldsymbol{\eta}}^\top\big)||^2. \end{aligned} \tag{8}$$

The derivation of (8) is given in the Appendix. As a direct consequence of (8), we obtain the profile log-likelihood function of $\boldsymbol{\eta}$ as

$$\ell_p(\boldsymbol{\eta}) = -\frac{1}{2\widehat{\sigma^2_{\boldsymbol{\eta}}}}||\mathbf{I}_m \odot \big(f(\mathbf{Y}|\boldsymbol{\eta}) - \widehat{\mathbb{U}}_{\boldsymbol{\eta}}\widehat{\mathbb{V}}_{\boldsymbol{\eta}}^\top\big)||^2 - \frac{N(\mathbf{I}_m)}{2}\log(\widehat{\sigma^2_{\boldsymbol{\eta}}}) + \sum_{(i,j):\mathbf{I}_m(i,j)=1}\big\{\log|f'(y_{ij}|\boldsymbol{\eta})|\big\}. \tag{9}$$

Using either Algorithm 1 or 2 mentioned above and equation (9), we can modify the PCA.t algorithm to handle the missing data. We refer to the modified algorithm as "PCA.$\mathrm{t_m}$". The main differences between the algorithms PCA.$\mathrm{t_m}$ and PCA.t are (a) the $\widehat{\mathbb{U}}_{\boldsymbol{\eta}}$, $\widehat{\mathbb{V}}_{\boldsymbol{\eta}}$, and $\widehat{\sigma^2_{\boldsymbol{\eta}}}$ are not the direct solution of the SVD of the data matrix, and we may use either the Algorithm 1 or 2 to obtain these quantities in the second step; (b) the objective function in the third step is not the profile log-likelihood (4) based on the complete data but the profile log-likelihood (9) based on the incomplete data.

3 Functional data structure

3.1 Integrating the data transformation in functional PCA

For functional data, we still use the model (2) to integrate data transformation to PCA. The functional structure of the data suggests that each PC weight vector, i.e., each column of $\mathbb{V}$, should be evaluations of a smooth function at a grid. To ensure the smoothness of the FPC weight vector, we consider the penalized log-likelihood

$$p\ell(\mathbf{\Theta}) = \ell(\mathbf{\Theta}) - \mathrm{pen}(\mathbb{V}; \mathbf{\Omega}),$$

where $\ell(\mathbf{\Theta})$ is the log-likelihood function given in (3), and $\mathbf{\Omega}$ is a non-negative definite roughness penalty matrix.

Following Huang et al. (2008), we define the penalty term based on a common penalty parameter $\widetilde{\alpha}$ for all of the FPC weight vectors as follows:

$$p\ell(\boldsymbol{\Theta}) = \ell(\boldsymbol{\Theta}) - \widetilde{\alpha}\text{tr}(\mathbb{U}^\top\mathbb{U}\mathbb{V}^\top\boldsymbol{\Omega}\mathbb{V}). \tag{10}$$

Similar to the previous section, we estimate the transformation parameter $\boldsymbol{\eta}$ based on the penalized profile log-likelihood. Specifically, denoting $\alpha = \widetilde{\alpha}/2\sigma^2$, we maximize $p\ell(\boldsymbol{\Theta})$ in equation (10) for fixed parameters $\boldsymbol{\eta}$ and α with respect to $\mathbb{U}$ and $\mathbb{V}$. This maximization problem is equivalent to minimizing the following penalized reconstruction error criterion

$$||\mathbf{X}_{\boldsymbol{\eta}} - \mathbb{U}\mathbb{V}^\top||^2 + \alpha\text{tr}\big(\mathbb{U}^\top\mathbb{U}\mathbb{V}^\top\boldsymbol{\Omega}\mathbb{V}\big), \tag{11}$$

for a fixed $\boldsymbol{\eta}$ and α, where $\mathbf{X}_{\boldsymbol{\eta}} = f(\mathbf{Y}|\boldsymbol{\eta})$. This is the penalized low-rank approximation formulation of FPCA in Huang et al. (2008).

To see the connection between the above formulation and the maximum variance formulation of Silverman (1996), one observes that for a fixed $\mathbb{V}$, the value of $\mathbb{U}$ that minimizes the penalized reconstruction error is

$$\mathbb{U} = \mathbf{X}_{\boldsymbol{\eta}}\mathbb{V}\{\mathbb{V}^\top(\mathbf{I} + \alpha\boldsymbol{\Omega})\mathbb{V}\}^{-1}.$$

Plugging this $\mathbb{U}$ back into criterion function (11), one can see that minimizing the resulting criterion function is same as maximizing

$$\text{tr}\Big(\mathbb{V}^\top\mathbf{X}_{\boldsymbol{\eta}}^\top\mathbf{X}_{\boldsymbol{\eta}}\mathbb{V}\{\mathbb{V}^\top(\mathbf{I} + \alpha\boldsymbol{\Omega})\mathbb{V}\}^{-1}\Big),$$

with respect to $\mathbb{V}$, which is essentially the approach of Silverman (1996).

Fixing the parameters $\boldsymbol{\eta}$ and α, and using the half-smoothing technique (Silverman, 1996; Huang et al., 2008), we can rewrite the penalized reconstruction error (11) as

$$||\mathbf{X}_{\boldsymbol{\eta}}||^2 - ||\widetilde{\mathbf{X}}_{\boldsymbol{\eta}}||^2 + ||\widetilde{\mathbf{X}}_{\boldsymbol{\eta}} - \mathbb{U}\widetilde{\mathbb{V}}^\top||^2,$$

where $\widetilde{\mathbf{X}}_{\boldsymbol{\eta}} = \mathbf{X}_{\boldsymbol{\eta}}\mathbf{S}_\alpha^{1/2}$ and $\widetilde{\mathbb{V}} = \mathbf{S}_\alpha^{-1/2}\mathbb{V}$ for $\mathbf{S}_\alpha = (\mathbf{I} + \alpha\boldsymbol{\Omega})^{-1}$. Clearly the minimizer of the reconstruction error may be obtained by the rank-d truncated SVD of $\widetilde{\mathbf{X}}_{\boldsymbol{\eta}}$. Interpreting $\mathbf{S}_\alpha^{1/2}$ as a half-smoothing operator, the transformed matrix $\widetilde{\mathbf{X}}_{\boldsymbol{\eta}}$ is obtained by half-smoothing the rows of the transformed data matrix $\mathbf{X}_{\boldsymbol{\eta}}$. After $\widetilde{\mathbb{V}}$ is calculated as the first d right singular vectors of $\widetilde{\mathbf{X}}_{\boldsymbol{\eta}}$, we half-smooth it to obtain the smoothed PC function $\mathbb{V} = \mathbf{S}_\alpha^{1/2}\widetilde{\mathbb{V}}$. Note that the resulting $\mathbb{U}$ and $\mathbb{V}$ are the minimizer of the criterion (11). Therefore, by considering the truncated SVD

of $\widetilde{\mathbf{X}}_{\boldsymbol{\eta}}$ as $\mathbf{U_d}\mathbf{\Sigma_d}\widetilde{\mathbf{V}}_{\mathbf{d}}{}^{\top}$, and fixing the parameters $\boldsymbol{\eta}$ and α, the explicit forms of the penalized MLEs for the rest of the parameters are

$$\begin{aligned} \widehat{\mathbb{U}}_{\boldsymbol{\eta}} &= \mathbf{U_d}\mathbf{\Sigma_d}, & \widehat{\mathbb{V}}_{\boldsymbol{\eta}} &= \mathbf{S}_{\alpha}^{1/2}\widetilde{\mathbf{V}}_{\mathbf{d}}, \\ \widehat{\sigma^2_{\boldsymbol{\eta}}} &= \frac{1}{nm}\{||f(\mathbf{Y}|\boldsymbol{\eta}) - \widehat{\mathbb{U}}_{\boldsymbol{\eta}}\widehat{\mathbb{V}}_{\boldsymbol{\eta}}^{\top}||^2 + \alpha\mathrm{tr}(\widehat{\mathbb{U}}_{\boldsymbol{\eta}}^{\top}\widehat{\mathbb{U}}_{\boldsymbol{\eta}}\widehat{\mathbb{V}}_{\boldsymbol{\eta}}^{\top}\mathbf{\Omega}\widehat{\mathbb{V}}_{\boldsymbol{\eta}})\}. \end{aligned} \tag{12}$$

Consequently, the penalized profile log-likelihood function is

$$\begin{aligned} p\ell_p(\boldsymbol{\eta}) &= -\frac{1}{2\widehat{\sigma^2_{\boldsymbol{\eta}}}}\{||f(\mathbf{Y}|\boldsymbol{\eta}) - \widehat{\mathbb{U}}_{\boldsymbol{\eta}}\widehat{\mathbb{V}}_{\boldsymbol{\eta}}^{\top}||^2 + \alpha\mathrm{tr}(\widehat{\mathbb{U}}_{\boldsymbol{\eta}}^{\top}\widehat{\mathbb{U}}_{\boldsymbol{\eta}}\widehat{\mathbb{V}}_{\boldsymbol{\eta}}^{\top}\mathbf{\Omega}\widehat{\mathbb{V}}_{\boldsymbol{\eta}})\} \\ &\quad -\frac{nm}{2}\log(\widehat{\sigma^2_{\boldsymbol{\eta}}}) + \sum_{i=1}^{n}\sum_{j=1}^{m}\big\{\log|f'(y_{ij}|\boldsymbol{\eta})|\big\}. \end{aligned} \tag{13}$$

We present the functional version of PCA.t algorithm, referred to as FPCA.t, as follows:

1. Start from an initial estimate of $\boldsymbol{\eta}$, denoted by $\boldsymbol{\eta}_0$ ($t = 0$). Usually we pick the initial estimate associated with the model with no transformation.

2. Let $\mathbf{X}_t$ be the $n \times m$ matrix with the $(i,j)^{\text{th}}$ entry $f(y_{ij}|\boldsymbol{\eta}_t)$.

   (a). Use the cross-validation technique given in Section 3.2 to obtain the α for fixed $\boldsymbol{\eta}_t$.

   (b). Define $\widetilde{\mathbf{X}}_t = \mathbf{X}_t\mathbf{S}_{\alpha}^{1/2}$. Obtain the rank-$d$ truncated SVD of $\widetilde{\mathbf{X}}_t$ as $\mathbf{U_d}\mathbf{\Sigma_d}\widetilde{\mathbf{V}}_{\mathbf{d}}{}^{\top}$ and use equations (12) to obtain $\widehat{\mathbb{U}}_t$, $\widehat{\mathbb{V}}_t$, and $\widehat{\sigma^2_t}$, where $\widehat{\mathbb{U}}_t$, $\widehat{\mathbb{V}}_t$, and $\widehat{\sigma^2_t}$ are $\widehat{\mathbb{U}}_{\boldsymbol{\eta}}$, $\widehat{\mathbb{V}}_{\boldsymbol{\eta}}$, and $\widehat{\sigma^2_{\boldsymbol{\eta}}}$ at $\boldsymbol{\eta} = \boldsymbol{\eta}_t$.

3. Obtain the updated value of $\boldsymbol{\eta}_{t+1}$ via an optimization algorithm to increase the value of the penalized profile log-likelihood function $p\ell_p(\boldsymbol{\eta}_t)$ defined in (13).

4. Iterate between the last two steps until convergence is reached.

3.2 Choosing the penalty parameter

To select the penalty parameters, we adopt the cross-validation (CV) and generalized cross-validation (GCV) criteria developed in Huang et al. (2008). For a fixed parameter $\boldsymbol{\eta}$, consider the SVD of $\mathbf{X}_{\boldsymbol{\eta}}\mathbf{S}_{\alpha}^{1/2} = \mathbf{U}\mathbf{\Sigma}\widetilde{\mathbf{V}}$ and define $\mathbf{V} = \mathbf{S}_{\alpha}^{1/2}\widetilde{\mathbf{V}}$. Let $\mathbf{U_d}$ and $\mathbf{V_d}$ be the first d columns of $\mathbf{U}$ and $\mathbf{V}$ respectively, and $\mathbf{\Sigma_d}$ be a diagonal matrix formed based on the first d diagonal elements of $\mathbf{\Sigma}$ or the d largest singular values of $\mathbf{X}_{\boldsymbol{\eta}}\mathbf{S}_{\alpha}^{1/2}$. The CV and GCV criteria are defined as follows:

$$CV(\alpha) = \frac{1}{m}\sum_{j=1}^{m}\frac{[\{(\mathbf{I} - \mathbf{S}_{\alpha})(\mathbf{X}_{\boldsymbol{\eta}}^{\top}\mathbf{U_d})\}_{jj}]}{(1 - \{\mathbf{S}_{\alpha}\}_{jj})}, \qquad GCV(\alpha) = \frac{||\mathbf{V_d}\mathbf{\Sigma_d} - \mathbf{X}_{\boldsymbol{\eta}}^{\top}\mathbf{U_d}||^2/m}{\{1 - \mathrm{tr}(\mathbf{S}_{\alpha})/m\}^2}.$$

Huang et al. (2008) showed that these CV and GCV criteria can be derived from the basic idea of cross-validation and generalized cross-validation of Craven and Wahba (1979).

3.3 Missing data and the functional data structure

Missing values commonly occur in practice due to the unavailability of the subjects over time sequences, deficiency of the measurement devices in some locations, and other limits and constraints. One may ignore the functional structure and use the PCA.$\mathrm{t_m}$ algorithms in Section 2.2 to find the transformation parameter and impute the missing data, then calculate the functional PCA. This two-step procedure is however *ad-hoc*. Here, we provide a solution that uses the penalized likelihood function (10) in the presence of missing observations.

As in Section 3.1, consider the penalized reconstruction error for a fixed $\boldsymbol{\eta}$ and α, with respect to the observed-data points. The penalized reconstruction error has the following form

$$O(\mathbb{U},\mathbb{V}) = ||\mathbf{I}_m \odot \left(\mathbf{X}_{\boldsymbol{\eta}} - \mathbb{U}\mathbb{V}^\top\right)||^2 + \alpha \mathrm{tr}(\mathbb{U}\mathbb{U}^\top \mathbb{V}\boldsymbol{\Omega}\mathbb{V}^\top). \tag{14}$$

Due to the presence of missing observations, finding the $\widehat{\mathbb{U}}$ and $\widehat{\mathbb{V}}$ that minimize (14) is not straightforward. Therefore, we present two iterative algorithms to find the smoothed SVD of the incomplete data matrix. These two algorithms are extensions of Algorithms 1 and 2 to functional data.

Algorithm 3 below requires imputation of missing data and can be derived as a majorization-minimization (MM) algorithm (see Appendix for derivation). In general, the MM algorithm can be considered as a class of algorithms that contains the EM algorithm as a special case (Hunter and Lange, 2004). In the majorization step, a surrogate function has been obtained so as to be tangent to the criterion function at $(\mathbb{U}_t, \mathbb{V}_t)$, but larger than this criterion function for any other values of $(\mathbb{U}, \mathbb{V})$. The minimization step consists of minimizing the surrogate function with respect to $\mathbb{U}$ and $\mathbb{V}$ to obtain $(\mathbb{U}_{t+1}, \mathbb{V}_{t+1})$, which results in $O(\mathbb{U}_{t+1}, \mathbb{V}_{t+1}) \leq O(\mathbb{U}_t, \mathbb{V}_t)$. Repeating the iterations will lead to a local minimum; see Hunter and Lange (2004) for details.

Algorithm 3

1. Set $t \leftarrow 0$, and let $\mathbf{X}_t = \mathbf{I}_m \odot \mathbf{X}_{\boldsymbol{\eta}}$.

2. Repeat the following steps until convergence.

   (a). Let $\widetilde{\mathbf{X}}_t = \mathbf{X}_t \mathbf{S}_\alpha^{1/2}$. Obtain the rank-$d$ truncated SVD of the matrix $\widetilde{\mathbf{X}}_t$ as $\mathbf{U_d}\boldsymbol{\Sigma_d}\widetilde{\mathbf{V}}_{\mathbf{d}}{}^\top$. Let $\mathbf{V_d} = \mathbf{S}_\alpha^{1/2}\widetilde{\mathbf{V}}_{\mathbf{d}}$.

(b). Define $\mathbf{X}_{t+1} = \mathbf{I}_m \odot \mathbf{X}_{\boldsymbol{\eta}} + \mathbf{I}_m^c \odot (\mathbf{U_d}\boldsymbol{\Sigma}_\mathbf{d}\mathbf{V_d}^\top)$.

(c). Set $t \leftarrow t+1$.

3. After convergence record $\mathbf{U_d}$, $\mathbf{V_d}$, and $\boldsymbol{\Sigma}_\mathbf{d}$ as the output of this algorithm.

Although the above MM algorithm has a nice property of simultaneously obtaining the dominant smoothed principal components, similar to Algorithm 1 presented in Section 2.2, it may suffer from slow convergence. We thus develop an extension of Algorithm 2 as follows (see Appendix for derivation):

Algorithm 4:

1. Set $t \leftarrow 0$, and obtain the rank-d truncated SVD of the matrix $\mathbf{I}_m \odot \mathbf{X}_{\boldsymbol{\eta}}$ as $\mathbf{U_d}\boldsymbol{\Sigma}_\mathbf{d}\mathbf{V_d}^\top$.

2. Repeat the following steps until convergence.

   (a). For $k = 1, \cdots, d$,

      i. Let $\mathbf{X}_{t,k} = \mathbf{I}_m \odot (\mathbf{X}_{\boldsymbol{\eta}} - \mathbf{U}_{-k}\boldsymbol{\Sigma}_{-k}\mathbf{V}_{-k}^\top)$.

      ii. Set $\boldsymbol{u}_{t,k} = \{\mathrm{diag}(\mathbf{I}_m\mathbf{V}_{\cdot k}^2 + \alpha\mathbf{V}_{\cdot k}^\top\boldsymbol{\Omega}\mathbf{V}_{\cdot k}\mathbf{1})\}^{-1}\mathbf{X}_k\mathbf{V}_{\cdot k}$,
      and $\boldsymbol{v}_{t,k} = \{\mathrm{diag}(\mathbf{I}_m^\top\mathbf{U}_{\cdot k}^2) + \alpha\boldsymbol{\Omega}\}^{-1}\mathbf{X}_k^\top\mathbf{U}_{\cdot k}$.

      iii. Normalize the vectors $\boldsymbol{u}_{t,k}$ and $\boldsymbol{v}_{t,k}$ to have unit length.

      iv. Update $\mathbf{U}_{\cdot k} = \boldsymbol{u}_{t,k}$, $\mathbf{V}_{\cdot k} = \boldsymbol{v}_{t,k}$ and $\boldsymbol{\Sigma}_{kk} = \dfrac{(\boldsymbol{u}_{t,k}^\top\mathbf{X}_{t,k}\boldsymbol{v}_{t,k})}{(\boldsymbol{u}_{t,k}^{2\top}\mathbf{I}_m\boldsymbol{v}_{t,k}^2 + \alpha\boldsymbol{v}_{t,k}^\top\boldsymbol{\Omega}\boldsymbol{v}_{t,k})}$.

   (b). Set $t \leftarrow t+1$.

3. After convergence, record $\mathbf{U_d}$, $\mathbf{V_d}$, and $\boldsymbol{\Sigma}_\mathbf{d}$ as the output of this algorithm.

Similar to Algorithm 2, this is an extension of the power algorithm for computing singular vectors. This algorithm differs from Algorithm 2 in that (a) different weight matrices are premultiplied in Step 2(a)-ii and (b) different normalizing factors are used to obtain the singular values in Step 2(a)-iv.

Let $\mathbf{U_d}$, $\mathbf{V_d}$, and $\boldsymbol{\Sigma}_\mathbf{d}$ be the output of either Algorithm 3 or 4 in presence of missing values. Following the model (2) and the penalized log-likelihood function (10), for a fixed parameter $\boldsymbol{\eta}$, the penalized MLEs for the rest of parameters are

$$\begin{aligned} \widehat{\mathbb{U}}_{\boldsymbol{\eta}} &= \mathbf{U_d}\boldsymbol{\Sigma}_\mathbf{d}, & \widehat{\mathbb{V}}_{\boldsymbol{\eta}} &= \mathbf{V_d}, \\ \widehat{\sigma^2_{\boldsymbol{\eta}}} &= \frac{1}{N(\mathbf{I}_m)}\{||\mathbf{I}_m \odot \big(f(\mathbf{Y}|\boldsymbol{\eta}) - \widehat{\mathbb{U}}_{\boldsymbol{\eta}}\widehat{\mathbb{V}}_{\boldsymbol{\eta}}^\top\big)||^2 + \alpha\mathrm{tr}(\widehat{\mathbb{U}}_{\boldsymbol{\eta}}^\top\widehat{\mathbb{U}}_{\boldsymbol{\eta}}\widehat{\mathbb{V}}_{\boldsymbol{\eta}}^\top\boldsymbol{\Omega}\widehat{\mathbb{V}}_{\boldsymbol{\eta}})\}, \end{aligned} \tag{15}$$

The derivation of (15) is given in the Appendix. As a direct consequence of (15), we obtain the penalized profile log-likelihood function as

$$\begin{aligned} p\ell_p(\boldsymbol{\eta}) &= -\frac{1}{2\widehat{\sigma^2_{\boldsymbol{\eta}}}}\{\mathbf{I}_m \odot ||f(\mathbf{Y}|\boldsymbol{\eta}) - \widehat{\mathbb{U}}_{\boldsymbol{\eta}}\widehat{\mathbb{V}}_{\boldsymbol{\eta}}^\top||^2 + \alpha \mathrm{tr}(\widehat{\mathbb{U}}_{\boldsymbol{\eta}}^\top\widehat{\mathbb{U}}_{\boldsymbol{\eta}}\widehat{\mathbb{V}}_{\boldsymbol{\eta}}^\top\boldsymbol{\Omega}\widehat{\mathbb{V}}_{\boldsymbol{\eta}})\} \\ &\quad -\frac{N(\mathbf{I}_m)}{2}\log(\widehat{\sigma^2_{\boldsymbol{\eta}}}) + \sum_{(i,j):\mathbf{I}_m(i,j)=1}\big\{\log|f'(y_{ij}|\boldsymbol{\eta})|\big\}. \end{aligned} \tag{16}$$

The FPCA.t algorithm can then be easily modified to handle the missing data. We refer to the modified algorithm as "FPCA.$\mathrm{t_m}$". The main differences between the algorithms FPCA.$\mathrm{t_m}$ and FPCA.t are (a) equation (15) is used to compute the MLEs $\widehat{\mathbb{U}}_{\boldsymbol{\eta}}$, $\widehat{\mathbb{V}}_{\boldsymbol{\eta}}$, and $\widehat{\sigma^2_{\boldsymbol{\eta}}}$ for fixed $\boldsymbol{\eta}$; (b) equation (16) is used to compute the profile log-likelihood.

4 Data Examples

In this section we illustrate the proposed methods using simulations and two real datasets.

4.1 Simulation

We considered the following data generating model:

$$x_{ij} = u_{i1}v_1(t_j) + u_{i2}v_2(t_j) + \epsilon_{ij}, \quad i = 1,\ldots,n;\ j = 1,\ldots,m, \tag{17}$$

where $u_{i1} \overset{\text{i.i.d}}{\sim} \pi N(\mu_1,\sigma^2) + (1-\pi)N(-\mu_1,\sigma^2)$, $u_{i2} \overset{\text{i.i.d}}{\sim} \pi N(\mu_2,\sigma^2) + (1-\pi)N(-\mu_2,\sigma^2)$, and $\epsilon_{ij} \overset{\text{i.i.d}}{\sim} N(0,\sigma^2)$. The parameters were set as $n = m = 101$, $\pi = 0.95$, $\mu_1 = 3000$, $\mu_2 = 200$, $\sigma = 10$, and the 101 grid points t_j are equally distanced from -1 to 1. We specified the underlying functional principal components as follows:

$$v_1(t) = \frac{1}{s_1}\{t + \sin(\pi t)\} \quad \text{and} \quad v_2(t) = \frac{1}{s_2}\cos(3\pi t),$$

where s_1 and s_2 are the normalizing constants to make v_1 and v_2 unit vectors. We considered five different values of β, $\{2, 1, 0.5, 0.25, 0.1\}$, in the Box-Cox transformation (6) and generated simulated datasets for each β from

$$y_{ij} = f^{-1}(x_{ij}|\beta), \quad i = 1\ldots n, \quad j = 1\ldots m. \tag{18}$$

We compared the FPCA results obtained from three methods: (a) ignoring the transformation and obtaining the FPCA results for non-transformed data using the FPCA procedure

of Huang et al. (2008); (b) a two-step procedure that first estimates the best Box-Cox transformation using raw data and then obtains the functional PCA; and (c) the proposed FPCA.t procedure.

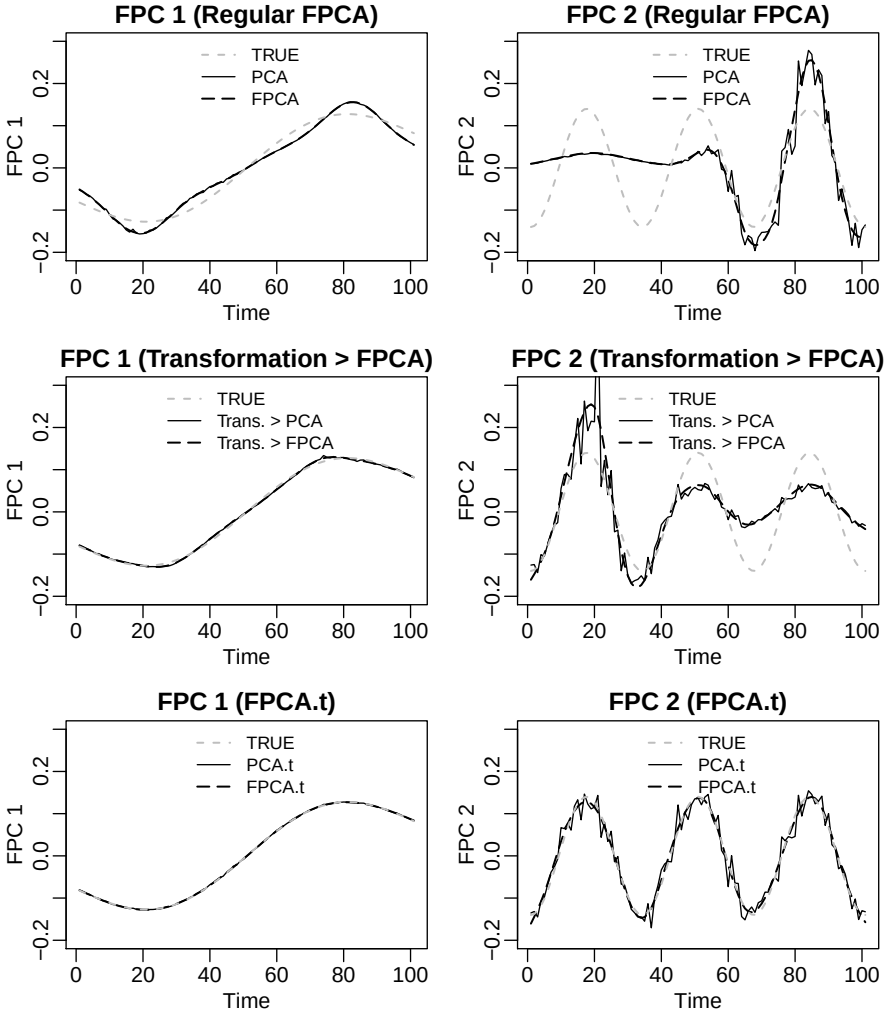

Figure 1: The estimated FPC weight functions for a simulated data set. Top row: results from the regular FPCA model; middle row: results from the two-step procedure; bottom row: results from the FPCA.t procedure. The gray dashed lines are the true FPC weight functions. The noisy black curves are the results of the PCA (PCA.t) and the smooth black dashed lines are the results of the FPCA (FPCA.t).

Figure 1 displays the first two FPC weight functions obtained by applying the three methods to a randomly selected dataset simulated from (18) with $\beta = 0.25$. The first row of the figure indicates that without data transformation, the FPC weight functions can not be recovered. The second and third rows show that both the two-step procedure and the proposed FPCA.t procedure can recover the first FPC weight function well, but the two-step procedure may miss the true structure of the second FPC weight function. The plots also suggests that the

penalization used in FPCA is necessary for obtaining a smooth FPC weight functions. In our study, we also observed that the difference of results between the two-step procedure and the proposed FPCA.t gets larger when the value of β gets closer to zero (plots not shown).

To give a more systematic comparison of the three methods, we considered two measures of the quality of estimating the FPC weight functions. The first measure is the canonical angle between the column space of $\widehat{\mathbb{V}}$ and $\mathbb{V}$, defined as maximum angle between any two vectors from the two spaces. Mathematically it can be computed as $\mathsf{angle} = \cos^{-1}(\rho) \times 180/\pi$, where ρ is the minimum singular value of the matrix $\mathbf{Q}_{\widehat{\mathbb{V}}}^{\top}\mathbf{Q}_{\mathbb{V}}$, where $\mathbf{Q}_{\widehat{\mathbb{V}}}$ and $\mathbf{Q}_{\mathbb{V}}$ are orthonormal matrices obtained by the QR decomposition of matrices $\widehat{\mathbb{V}}$ and $\mathbb{V}$, respectively (Golub and Van Loan, 2013). The second measure is the squared out-of-sample reconstruction error using the estimated FPC weight functions as the basis. Specifically, we generate a test dataset $\mathbf{X}^{(test)}$ from (17), and calculate the discrepancy between $\mathbf{X}^{(test)}$ and its projection to the subspace spanned by the FPC weight functions obtained from the training set, quantified as the sum of squared errors $\mathsf{SSE} = ||\mathbf{X}^{(test)} - \mathbf{X}^{(test)}\widehat{\mathbb{V}}(\widehat{\mathbb{V}}^{\top}\widehat{\mathbb{V}})^{-1}\widehat{\mathbb{V}}^{\top}||^2$. Smaller values of angle and SSE indicate better estimation of the FPC weight functions.

Table 1 reports the summary statistics of using the above two performance measures to compare the three methods on 100 simulated datasets for each of the five different values of β. For the case $\beta = 1$ when the data transformation is not needed, the proposed FPCA.t method estimates β to be close to 1 and performs similarly to FPCA. In all other cases, the FPCA.t works consistently better than other two methods, producing smaller values of angle and SSE. When the value of β gets smaller, the data distribution is more skewed—this is the reason why application of the regular FPCA without data transformation gets worse as β decreases. While the FPCA.t estimates β very well, the two-step procedure usually gives biased estimation. A good illustration of the bias problem of the two-step procedure is the case of $\beta = 1$, where estimated β has the mean value of 0.4068. The bias of two-step procedure can be explained by that the transformation step focuses on the marginal distribution which contains information from both the signal and noise. Because of the biased estimation of β, the two-step procedure can not provide good estimation of FPCA weight functions, as confirmed by the bigger values of angle and SSE shown in Table 1. (Note that it is not an error that the numbers in some columns are the exactly the same; there are indeed small variations that cannot be seen in the reported significant digits.)

The simulation study was conducted on a computer with Core i7-2600 processor @ 3.4 GHz speed with 16 GB of RAM. The summary statistics reported in Table 1 on computation time suggests that the time of running the proposed FPCA.t method is significantly longer than the regular FPCA procedure. Our tracking of the timing of the algorithm reveals that the bottleneck of the FPCA.t algorithm is the repeated application of the half-smoothing operation; when no penalization is used, the run time of the PCA.t algorithm is no greater than double of the run time of the regular PCA procedure.

Table 1: Comparison of three FPCA methods (the regular FPCA, the two-step procedure of transformation followed by FPCA, and the FPCA.t) under different values of β, based on 100 simulation runs on each of five values of β. Reported are the mean and standard errors (in parentheses) of the principal angle, SSE, and the run time (in seconds).

	FPCA			transformation → FPCA				FPCA.t			
True Par.	angle	SSE	time	$\hat{\beta}$	angle	SSE	time	$\hat{\beta}$	angle	SSE	time
$\beta = 2.00$	29.2	1700.2	0.1967	0.8133	33.9	1907.3	0.3679	2.0062	2.8	1020.9	3.3871
	(0.3)	(12.6)	(0.0036)	(9e-04)	(0.3)	(13.7)	(0.0045)	(1e-03)	(0.1)	(0.5)	(0.0650)
$\beta = 1.00$	2.8	1021.0	0.1810	0.4067	33.9	1907.0	0.3665	1.0031	2.8	1020.9	3.459
	(0.1)	(0.50)	(0.0016)	(4e-04)	(0.3)	(13.7)	(0.0041)	(7e-04)	(0.1)	(0.5)	(0.1638)
$\beta = 0.50$	33.9	2235.4	0.2029	0.2034	33.9	1906.6	0.3660	0.5015	2.8	1020.8	4.8038
	(0.3)	(24.0)	(0.0032)	(2e-04)	(0.3)	(13.7)	(0.0038)	(3e-04)	(0.1)	(0.5)	(0.3271)
$\beta = 0.25$	47.0	5244.4	0.2218	0.1017	33.9	1906.1	0.3730	0.2508	2.8	1020.8	5.7202
	(0.1)	(42.0)	(0.0037)	(1e-04)	(0.3)	(13.7)	(0.0043)	(2e-04)	(0.1)	(0.5)	(0.4391)
$\beta = 0.10$	52.4	13262.6	0.2237	0.0407	33.8	1905.2	0.3720	0.1003	2.8	1020.8	6.2781
	(0.1)	(59.5)	(0.0037)	(5e-05)	(0.3)	(13.7)	(0.0044)	(1e-04)	(0.1)	(0.5)	(0.4781)

Next, we evaluate the performance of FPCA.t$_m$ in obtaining the FPC weight functions in the presence of missing observations. For each simulated dataset used in producing Table 1, we randomly removed either 10% or 25% of the observations in two separate runs. We applied the following three methods: (a) impute the missing data using the method of Hastie et al. (1999) (i.e., Algorithm 1) followed by applying the FPCA algorithm (imputation → FPCA); (b) impute the missing data using the method of Hastie et al. (1999) followed by implementing the FPCA.t algorithm (imputation → FPCA.t); and (c) use the proposed FPCA.t$_{\mathrm{m}}$ procedure (with Algorithm 4 for computing the profile likelihood). Method (a) does not consider data transformation at all. method (b) considers data transformation not in the missing data imputation step but in the FPCA step. Our method (c) contains missing data handling, data

transformation, and FPCA in one unified procedure. Table 2 shows that the FPCA.t_m method consistently outperforms other two methods, excepts for the case of $\beta = 1$ where method (a) and method (b) work equally well. On the other hand, when data transformation is needed, by ignoring data transformation entirely, method (a) does not produce good results. method (b) works comparably well with the proposed method for some β but its performance deteriorates quickly when β gets closer to 0. This is due to the fact when β approaches zero, the observed y_{ij}'s obtained from (18) converge to the log-normal distribution; hence imputing the missing values using the method of Hastie et al. (1999) (i.e., Algorithm 1) becomes less reliable for smaller βs, since Algorithm 1 essentially maximizes the Gaussian likelihood (see Appendix).

Table 2: Comparison of three FPCA methods for dealing with missing data ("imputation → FPCA", "imputation → FPCA.t", and FPCA.t_m) under five different values of β and two different missing rates. Reported values are the mean and standard errors (in parentheses) of the principal angle, SSE, and the run time (in seconds).

Rate of Missing	True Par.	imputation → FPCA			imputation → FPCA.t				FPCA.t_m			
		angle	SSE	time	$\hat{\beta}$	angle	SSE	time	$\hat{\beta}$	angle	SSE	time
10%	$\beta = 2.00$	30.021	1724.0	0.2436	2.0080	4.444	1028.0	3.5259	2.0081	4.443	1027.7	5.2407
		(0.348)	(12.8)	(0.0115)	(1e-03)	(0.127)	(0.9)	(0.2759)	(2e-03)	(0.126)	(0.9)	(0.2781)
	$\beta = 1.00$	4.521	1028.3	0.2441	1.0041	4.443	1027.7	3.7358	1.0041	4.443	1027.7	6.1054
		(0.126)	(0.90)	(0.0036)	(8e-04)	(0.126)	(0.9)	(0.0919)	(8e-04)	(0.126)	(0.9)	(0.1363)
	$\beta = 0.50$	33.843	2252.8	0.2542	0.5020	4.447	1027.7	4.1414	0.5020	4.443	1027.7	6.1434
		(0.335)	(21.8)	(0.0041)	(4e-04)	(0.128)	(0.9)	(0.1855)	(4e-04)	(0.126)	(0.9)	(0.2404)
	$\beta = 0.25$	47.127	5297.5	0.2710	0.1046	17.694	1705.0	5.3535	0.2510	4.443	1027.7	6.7732
		(0.069)	(35.0)	(0.0036)	(2e-03)	(0.337)	(14.3)	(0.3808)	(2e-04)	(0.126)	(0.9)	(0.3698)
	$\beta = 0.10$	52.245	13483	0.4280	0.0054	33.526	2807.3	5.1642	0.1004	4.443	1027.7	7.1433
		(0.162)	(48.5)	(0.0044)	(5e-04)	(0.482)	(16.0)	(0.4314)	(1e-04)	(0.126)	(0.9)	(0.5833)
25%	$\beta = 2.00$	29.827	1701.5	0.2226	2.0345	7.011	1066.2	3.3910	2.0043	6.486	1042.3	5.2706
		(0.394)	(14.1)	(0.0025)	(3e-02)	(0.406)	(21.3)	(0.2058)	(2e-03)	(0.203)	(1.9)	(0.1034)
	$\beta = 1.00$	6.594	1043.3	0.2313	1.0022	6.487	1042.3	3.2043	1.0022	6.486	1042.3	5.6459
		(0.201)	(1.90)	(0.0018)	(1e-03)	(0.203)	(1.9)	(0.0347)	(1e-03)	(0.203)	(1.9)	(0.1012)
	$\beta = 0.50$	34.034	2243.0	0.2460	0.5005	6.772	1053.4	3.7602	0.5011	6.486	1042.3	6.5995
		(0.347)	(21.6)	(0.0022)	(8e-04)	(0.316)	(4.1)	(0.1596)	(5e-04)	(0.203)	(1.9)	(0.3471)
	$\beta = 0.25$	47.324	5308.2	0.3916	0.1101	21.672	1825.4	5.2073	0.2506	6.487	1042.3	7.2528
		(0.122)	(36.1)	(0.0078)	(1e-03)	(0.489)	(16.4)	(0.3499)	(3e-04)	(0.203)	(1.9)	(0.5509)
	$\beta = 0.10$	52.47	13715	0.4484	0.0285	40.507	3383.1	7.1208	0.1002	6.487	1042.3	9.1735
		(0.168)	(51.9)	(0.0036)	(7e-03)	(0.576)	(296)	(0.6831)	(1e-04)	(0.203)	(1.9)	(0.8433)

4.2 Real data

We applied the proposed methods to two datasets from the literature. While sensible data transformations were decided manually in previous work before further analysis, our methods obtained suitable transformations automatically while performing PCA/FPCA.

4.2.1 Fruit Fly Mortality Data

To illustrate and assess the performance of the proposed approach in the presence of missing data, we considered the "fruit fly mortality" (FFM) data (Zimmerman and Núñez Antón, 2009). The dataset contains age-specific measurements of mortality for 112 cohorts of a common fruit fly, Drosophila melanogaster. Every day, dead flies were counted for each cohort, and these counts were pooled into 11 five-day intervals. The raw mortality rate was recorded as $-\log(N(t+1)/N(t))$, where $N(t)$ is the number of flies alive in the cohort at the beginning of time $t(t = 0, 1, \ldots, 10)$. For unknown reasons, 22% of the data are missing. We target estimating the data transformation while performing PCA and systematically imputing the missing values.

Considering that the 22% of the data are missing, one can use either (a) a two-step procedure that imputes the missing data using the method in Hastie et al. (1999) (i.e., Algorithm 1) and then applies the regular PCA; or (b) the proposed method PCA.t_m (with Algorithm 2 for computing the profile likelihood). When we applied method (a), however, Algorithm 1 did not work well for $d > 1$, specifically some imputed values were inflated towards infinity in magnitude. In contrast, PCA.t_m consistently performs reasonable data imputation regardless of the d value. To make a square comparison of the two methods, we only present here the results for $d = 1$. From Figure 2, it is clear that the distribution of imputed values obtained from method (a) does not follow the distribution of the observed values, in particular at the right end of the time interval. The PCA.$\mathrm{t_m}$ procedure obtained the estimate of $\hat{\beta} = 0.0147$, which is very close to the logarithmic transformation that was suggested by Zimmerman and Núñez Antón (2009). We also observed that the residuals obtained from the PCA.$\mathrm{t_m}$ algorithm behave closer to a normal distribution with constant variability.

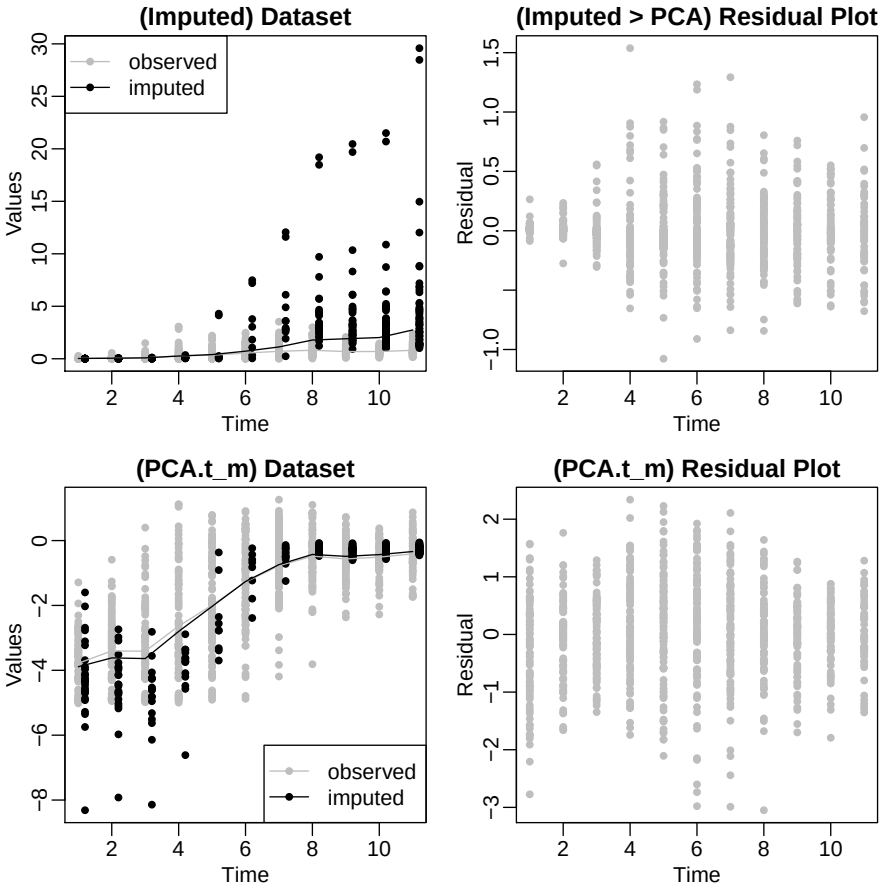

Figure 2: FFM data. Left panels: imputation of missing values in the original scale (top row) and using the PCA.t$_m$ method (bottom row); the gray and dark lines indicate the pointwise means for the observed data and after including the imputed data. Right panels: residual plot for the two-step procedure (top row) and for the PCA.t$_m$ method.

4.2.2 Call Center Data

The source of the second dataset is a small call center for an anonymous bank in Israel (Brown et al., 2005). This dataset provides the exact time of the calls that were connected to the call center between January 1 and December 31 in the year 1999. We would like to study the trend and variability of call volumes over time of the day. To do so we aggregate the data into time intervals to obtain a data matrix. More precisely, the i^{th}, row j^{th} column of our data matrix contains the call volume during the j^{th} time interval on day i. Brown et al. (2005) used the square-root transformation to stabilize the variance and make the distribution close to normal. The same transformation was also used by Huang et al. (2008), Shen and Huang (2008) prior to application of PCA and SVD in the analysis of a different dataset of call arrival volumes.

We applied the FPCA.t algorithm to automatically find the appropriate data transformation

and to obtain the FPC weight functions. We considered 10 different time interval lengths: 6, 8, 10, 12, 15, 20, 30, 45, 60, and 90 minutes, and studied consistency of the FPC weight for different levels of aggregation. As comparison, we also applied the regular FPCA without transformation and the two-step procedure that first finds a data transformation and then applies the regular FPCA to the transformed data.

Table 3: Estimated transformation parameter by the two-step procedure ("transformation → FPCA") and FPCA.t for different level of data aggregation.

Time intervals in min	6	8	10	12	15	20	30	45	60	90
transformation → FPCA	0.13	0.15	0.16	0.16	0.17	0.18	0.19	0.21	0.21	0.23
FPCA.t	0.19	0.23	0.25	0.27	0.29	0.31	0.35	0.38	0.39	0.42

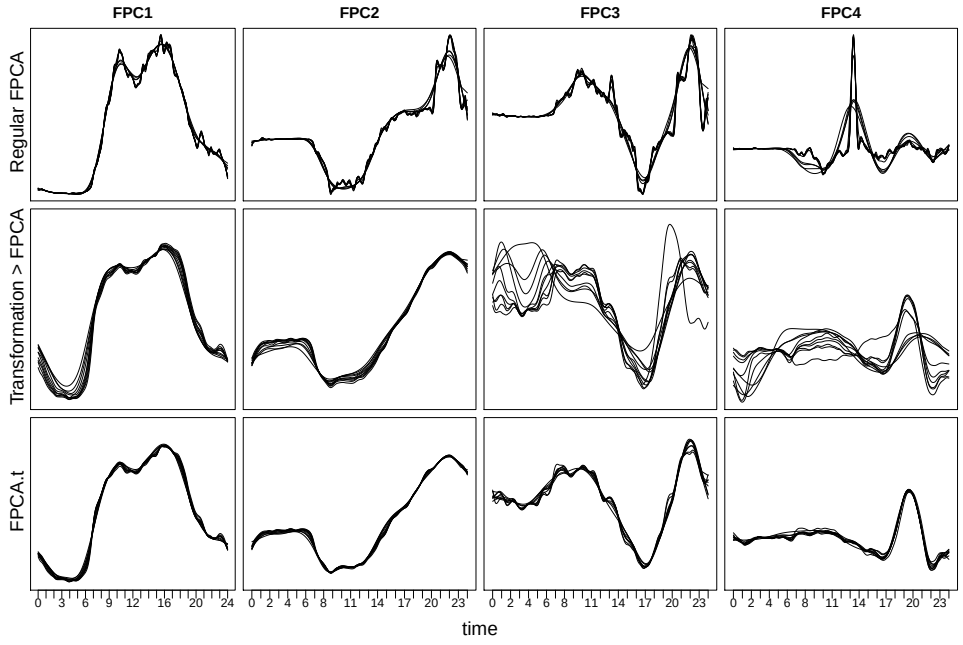

Figure 3: Call center data: The first four FPC weight functions obtained by three methods: the regular FPCA (first row), the two-step procedure (middle row), and the FPCA.t method (last row). In each panel, different lines correspond to different level of data aggregation.

Table 3 shows that FPCA.t estimated transformation parameter $\hat{\beta}$ is bigger than that produced by the two-step procedure but is still smaller than 0.5, the value used in the square-root transformation. The average of $\hat{\beta}$ over different level of data aggregation from FPCA.t is 0.31 with the standard error 0.02, while this average for the two-step procedure is 0.18 with the standard error 0.01. Furthermore, we have seen that the FPC weight obtained from FPCA.t

resemble the associated FPC weight from the square-root of the call volume (as suggested in literature by Brown et al., 2005; Huang et al., 2008; Shen and Huang, 2008) better than the two-step procedure. Figure 3 displays the first four FPC weight functions obtained by the three methods for each of the 10 different data aggregation levels. The first row of the figure indicates that when applying the regular FPCA to the untransformed data, the resulting FPCA weight functions are not very smooth. This may be contributed to the right-skewness of the data—there are many big values that influence the results. The data transformation helps improve the smoothness of the FPCA weight functions, as shown in the second and third row of the figure. On the other hand, while the two-step procedure manifests large variability particularly for the last two FPC weight functions, the proposed FPCA.t procedure produces more consistent estimates of FPCA weight functions across different data aggregation levels. Figure 4 suggests that with the same number of FPCs, the proposed FPCA.t method tends to explain more percentage of variance than other two methods.

5 Discussion

We propose a model-based approach to simultaneously obtain an appropriate data transformation and perform principal components analysis/functional principal components analysis. The model assumes that the transformed data matrix has a signal-plus-noise representation where the signal part is a low-rank matrix and the noise part contains independent zero-mean normally distributed random variables. In our approach, the normality of distribution only serves as a working assumption and its use is motivated by the consideration that after data transformation, the noise terms should have a less skewed distribution and close to constant variance. A similar normality assumption was used in maximum likelihood estimation of the Box-Cox transformation (Box and Cox, 1964). Our model-based approach is also convenient for handling missing data while performing principal components analysis together with data transformation.

Our method estimates the transformation parameter by the maximum profile likelihood and is very general to incorporate any parametric monotone transformations. We focus on the Box-Cox transformation in our presentation because it is commonly used in practice. The proposed method is most relevant when the variables in consideration are of similar nature (e.g., functional data) as in our two real data examples, and so the same transformation applied to

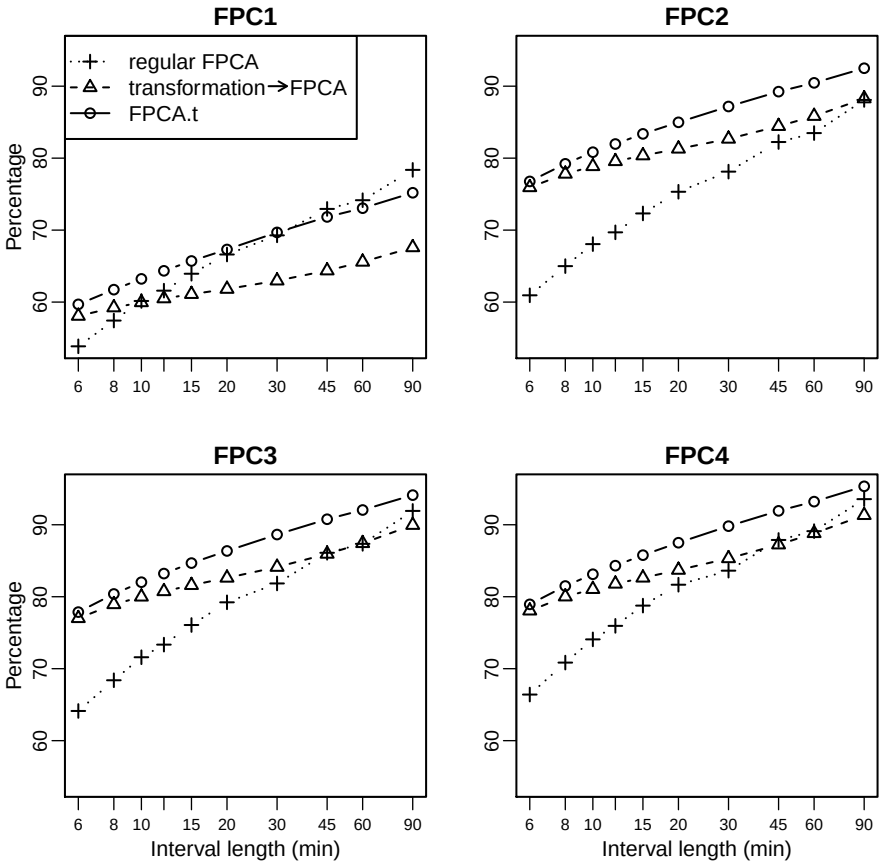

Figure 4: Call center data. Cumulative percentage of explained variance by the first four FPCA functions for three methods: the regular FPCA on untransformed data, the two-step procedure ("transformation → FPCA"), and FPCA.t_m for different level of data aggregation (note that the x-axis is in log scale).

all variables is desirable. In principle, it is possible to use different transformations to different variables but a careful investigation is beyond the scope of this paper and left for future research.

Acknowledgment

We would like to thank an associate editor and two anonymous referees for their constructive and thoughtful comments which helped us tremendously in revising the manuscript. Maadooliat and Hu were partially supported by the National Science Foundation (grants DMS-0706818), the National Institutes of Health (grants R01GM080503-01A1, R21CA129671), and the National Cancer Institute (grant CA97007). Huang was partially supported by the National Science Foundation (grants DMS-0606580, DMS-0907170). Huang and Maadooliat were partially supported by King Abdullah University of Science and Technology (grant KUS-CI-016-04).

Supplementary Materials

Appendices: Technical Appendices A-D. (webAppendix.pdf; pdf file)

Software: R code to implement FPCA.t, along with the call center data, fruit fly mortality data and the simulation study. Tested for R version 3.0.1; see readme.txt in the base directory for instructions on use. (FPCA.t_code&data.zip, zip file)

References

Avriel, M. (1976), *Nonlinear Programming: Analysis and Methods*, Englewood Cliffs, New Jersey: Prentice-hall.

Beckers, J. M. and Rixen, M. (2003), "EOF Calculations and Data Filling from Incomplete Oceanographic Datasets*," *Journal of Atmospheric and Oceanic Technology*, 20, 1839–1856.

Box, G. E. P. and Cox, D. R. (1964), "An Analysis of Transformations," *Journal of the Royal Statistical Society. Series B (Methodological)*, 26, pp. 211–252.

Brown, L., Gans, N., Mandelbaum, A., Sakov, A., Shen, H., Zeltyn, S., and Zhao, L. (2005), "Statistical Analysis of a Telephone Call Center: A Queueing-Science Perspective," *Journal of the American Statistical Association*, 100, 36–50.

Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C. (1994), "A Limited Memory Algorithm for Bound Constrained Optimization," *SIAM Journal on Scientific Computing*, 16, 1190–1208.

Craven, P. and Wahba, G. (1979), "Smoothing noisy data with spline functions. Estimating the correct degree of smoothing by the method of generalized cross-validation," *Numer. Math.*, 31, 377–403.

Croux, C. and Ruiz-Gazen, A. (2005), "High breakdown estimators for principal components: the projection-pursuit approach revisited," *Journal of Multivariate Analysis*, 95, 206 – 226.

Daniels, M. J. and Hogan, J. W. (2008), *Missing data in longitudinal studies: strategies for Bayesian modeling and sensitivity analysis; electronic version*, Monographs on Statistics and Applied Probability, Hoboken, NJ: Taylor & Francis Ltd.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977), "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, 39, pp. 1–38.

Golub, G. and Van Loan, C. (2013), *Matrix Computations*, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, 4th ed.

Hastie, T., Tibshirani, R., Sherlock, G., Eisen, M., Brown, P., and Botstein, D. (1999), "Imputing Missing Data for Gene Expression Arrays," Tech. rep., Stanford Statistics Department.

Higuchi, I. and Eguchi, S. (2004), "Robust Principal Component Analysis with Adaptive Selection for Tuning Parameters," *J. Mach. Learn. Res.*, 5, 453–471.

Hotelling, H. (1933), "Analysis of a Complex of Statistical Variables with Principal Components," *Journal of Educational Psychology*, 24, 498–520.

Hu, J., Wright, F. A., and Zou, F. (2006), "Estimation of Expression Indexes for Oligonucleotide Arrays Using the Singular Value Decomposition," *Journal of the American Statistical Association*, 101, 41–50.

Huang, J. Z., Shen, H., and Buja, A. (2008), "Functional principal components analysis via penalized rank one approximation," *Electronic Journal of Statistics 2008, Vol. 2, 678-695.*

Hubert, M., Rousseeuw, P., and Verdonck, T. (2009), "Robust PCA for skewed data and its outlier map," *Computational Statistics & Data Analysis*, 53, 2264 – 2274, the Fourth Special Issue on Computational Econometrics.

Hubert, M., Rousseeuw, P. J., and Verboven, S. (2002), "A fast method for robust principal components with applications to chemometrics," *Chemometrics and Intelligent Laboratory Systems*, 60, 101 – 111.

Hunter, D. R. and Lange, K. (2004), "A Tutorial on MM Algorithms," *The American Statistician*, 58.

Jolliffe, I. T. (2002), *Principal Component Analysis*, Springer, 2nd ed.

Locantore, N., Marron, J., Simpson, D., Tripoli, N., Zhang, J., Cohen, K., Boente, G., Fraiman, R., Brumback, B., Croux, C., Fan, J., Kneip, A., Marden, J., and P, D. (1999), "Robust

principal component analysis for functional data," *TEST: An Official Journal of the Spanish Society of Statistics and Operations Research*, 8, 1–73.

Maronna, R. (2005), "Principal Components and Orthogonal Regression Based on Robust Scales," *Technometrics*, 47, 264–273.

Pearson, K. (1901), "On Lines and Planes of Closest Fit to Systems of Points is Space," *Philosophical Magazine*, 2, 559–572.

Ramsay, J. and Silverman, B. (2005), *Functional Data Analysis (2nd Edition)*, Springer - Verlag.

Rice, J. A. and Silverman, B. W. (1991), "Estimating the mean and covariance structure nonparametrically when the data are curves," *Journal of the Royal Statistical Society. Series B (Methodological)*, 233–243.

Shen, H. and Huang, J. Z. (2008), "Interday Forecasting and Intraday Updating of Call Center Arrivals," *Manufacturing & Service Operations Management*, 10, 391–410.

Silverman, B. W. (1996), "Smoothed functional principal components analysis by choice of norm," *The Annals of Statistics*, 24, 1–24.

Stewart, G. W. (1993), "On the early history of the singular value decomposition," *SIAM Rev.*, 35, 551–566.

Tipping, M. E. and Bishop, C. M. (1999), "Probabilistic Principal Component Analysis," *Journal of the Royal Statistical Society, Series B*, 61, 611–622.

Trefethen, L. N. and Bau, D. (1997), *Numerical Linear Algebra*, SIAM: Society for Industrial and Applied Mathematics.

Wu, K. and Simon, H. (2000), "Thick-Restart Lanczos Method for Large Symmetric Eigenvalue Problems," *SIAM J. Matrix Anal. Appl.*, 22, 602–616.

Zimmerman, D. L. and Núñez Antón, V. (2009), *Antedependence Models for longitudinal Data*, Chapman & Hall / CRC Press, New York.