Computers & Industrial Engineering 64 (2013) 482–491

Contents lists available at SciVerse ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

A framework for nonparametric profile monitoring ☆

Shih-Chung Chuang [a], Ying-Chao Hung [b], Wen-Chi Tsai [b], Su-Fen Yang [b,*]

[a] Department of Industrial Engineering and Engineering Management, National Tsing Hua University, No. 101, Sec. 2, Kuang-Fu Rd., Hsinchu 30013, Taiwan, ROC
[b] Department of Statistics, National Chengchi University, No. 64, Sec. 2, ZhiNan Rd., Wenshan District, Taipei 11605, Taiwan, ROC

ARTICLE INFO

Article history:
Received 12 November 2011
Received in revised form 19 July 2012
Accepted 19 August 2012
Available online 11 September 2012

Keywords:
Nonparametric profile monitoring
B-spline
Block bootstrap
Confidence band
Curve depth

ABSTRACT

Control charts have been widely used for monitoring the functional relationship between a response variable and some explanatory variable(s) (called profile) in various industrial applications. In this article, we propose an easy-to-implement framework for monitoring nonparametric profiles in both Phase I and Phase II of a control chart scheme. The proposed framework includes the following steps: (i) data cleaning; (ii) fitting B-spline models; (iii) resampling for dependent data using block bootstrap method; (iv) constructing the confidence band based on bootstrap curve depths; and (v) monitoring profiles online based on curve matching. It should be noted that, the proposed method does not require any structural assumptions on the data and, it can appropriately accommodate the dependence structure of the within-profile observations. We illustrate and evaluate our proposed framework by using a real data set.

1. Introduction

In many industrial applications, the quality of a process (or product) can be characterized by a functional relationship between a quality measurement and the explanatory variables. Under such circumstances, statistical process control (SPC) aims on monitoring data (called profiles) that represent such a functional relationship, instead of on monitoring a single quality measurement. A nice overview and extensive discussions of profile monitoring can be found in Noorossana, Saghaei, and Amiri (2011). In this study, we are interested in monitoring nonparametric profiles for which the underlying functional relationship cannot be reasonably described by a pre-specified model. In particular, we focus on examining profiles that are collected over time. For this type of data, it is natural to assume that the within-profile observations are correlated.

Nonparametric profile monitoring has received increased attention in recent years due to its flexibility in modeling complex data structures. We highlight some remarkable works below. Chicken, Pignatiello, and Simpson (2009) and Lada, Lu, and Wilson (2002) used the wavelet-based approaches for process fault detection. Ding, Zeng, and Zhou (2006) used a dimension-reduction method for monitoring nonlinear profiles. Colosimo and Pacella (2007) used the principal component analysis for monitoring roundness profiles of manufactured items. Zou, Tsung, and Wang (2008) used a multivariate exponentially weighted moving average (MEWMA) chart and the generalized likelihood ratio test to monitor changes of the functional relationship based on local linear smoothers. Zhang and Albin (2009) used a chi-square control chart to identify outlying profiles without requiring explicit expression of the functional relationship. Zou, Qiu, and Hawkins (2009) used a change-point model and the generalized likelihood ratio tests to detect changes of the functional relationship. Chang and Yamada (2010) used a discrete wavelet transformation and B-splines to monitor the mean shifts and shape changes in a profile. For more relevant works the readers can refer to Chicken (2011) and references therein.

Traditional profile monitoring methods often assume the within-profile data are independent. On the other hand, methods that incorporate the correlation structure into analysis are rather limited. Two remarkable works can be found in Jensen and Birch (2009) and Qiu, Zou, and Wang (2010), of which the within-profile correlation was accounted for by using nonlinear mixed models and nonparametric mixed-effects models, respectively. These two approaches are in fact semi-parametric, which are flexible and novel from a theoretical viewpoint. However, from the viewpoint of implementation, they may not be practical due to a certain amount of numerical computation on the required parameter estimation (such as the maximum likelihood estimators) and test statistics (such as the likelihood ratio tests). Recently, Hung, Tsai, Yang, Chuang, and Tseng (2012) proposed a framework for nonparametric profile monitoring in multi-dimensional data spaces. They introduced a technique called Support Vector Regression (SVR) to model the functional relationship between the response variable and explanatory variables, while the within-profile correlation is accommodated by using a resampling technique called

☆ This manuscript was processed by Area Editor H.-S. Jacob Tsao.

* Corresponding author. Tel.: +886 2 29387115; fax: +886 2 29398024.
E-mail addresses: scchuang@ie.nthu.edu.tw (S.-C. Chuang), hungy@nccu.edu.tw (Y.-C. Hung), wctsai@nccu.edu.tw (W.-C. Tsai), yang@nccu.edu.tw (S.-F. Yang).

0360-8352/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cie.2012.08.006

block bootstrap. The idea therein was closely related to our proposed framework in this study. However, it has a shortcoming that the computation for monitoring the profiles online can be very intricate, especially when the number of explanatory variables becomes large.

Our goal here is to provide an easy-to-implement and computationally cheaper framework for monitoring nonparametric profiles by taking into account the within-profile correlation. To simplify the formulation of the problem, here we discuss the cases with only one covariate. Based on a certain number of observed in-control (IC) profiles, in Phase I we establish an adequate confidence band for the underlying functional relationship without requiring strong model assumptions. This confidence band can then serve as a control chart for Phase II process monitoring. The proposed framework in Phase I is mainly divided into five steps. In Step 1, an automated approach, called the two-sided median method, is used to clean each profile data. In Step 2, an adequate B-spline model is fitted to each profile data. In Step 3, the moving block bootstrap method (MBB) is used to generate correlated samples for each profile. In Step 4, the B-spline model is fitted to each of the bootstrap sample and its corresponding curve depth is calculated. In Step 5, the B-spline curves with smaller curve depths are removed and, the resulting confidence bands of all profiles are pooled so as to obtain a simultaneous confidence band for the underlying functional relationship. In Phase II, the idea of "curve matching" is used to establish the time-dependent B-spline model for monitoring a new profile online. The remaining of this study is organized as follows. In Section 2, the framework for nonparametric profile monitoring is introduced. In Section 3, the proposed framework is illustrated by using the AIDS data collected from hospitals in Taiwan. In addition, numerical comparisons are made to show that our proposed method works well in testing untried experiments. Some concluding remarks are drawn in Section 4.

2. A framework for nonparametric profile monitoring

Suppose there are M independent profiles obtained from a typical design of IC process and the ith profile has n_i observations, $i = 1, \ldots, M$. Let y_{ij} be the measurement of the jth observation in the ith profile and x_{ij} be the corresponding explanatory variable such that $j = 1, \ldots, n_i$ for each $i = 1, \ldots, M$. With this assumption, it is clear that the data range for each profile can be different. Suppose the underlying IC model is denoted by

$$y_{ij} = f(x_{ij}) + \varepsilon_{ij}, \quad i = 1, \ldots, M, \quad j = 1, \ldots, n_i, \tag{1}$$

where $f(\cdot)$ is a general function with some degree of smoothness, and ε_{ij} are associated error terms from some unknown distribution. It should be mentioned that here we mainly focus on time-series profiles, viz., profile i is observed at time $t_i, t_1 < t_2 < \cdots < t_M$, (x_{ij}, y_{ij}) is the pair of observed quantities for the ith profile at time t_{ij}, where $t_i = t_{i1} < t_{i2} < \cdots < t_{in_i}$. Further, for modeling flexibility we do not place any structural assumptions on the errors ε_{ij} (such as i.i.d. or normal distribution). This relaxed assumption is suitable for many real-life data that present the characteristics of nonnormality and correlation over time.

Our goal here is to propose an easy-to-implement framework for constructing an overall confidence band of $f(x)$ based on the observed IC profiles. The proposed framework is sequential and outlined as follows: (i) an automated method is used to clean the profile data; (ii) an adequate B-spline model is fitted to each profile; (iii) the block bootstrap method is used to resample from each profile data; (iv) the confidence band for each profile is constructed based on the bootstrap percentiles of curve depths; and (v) the simultaneous confidence band for the desired function f is obtained by pooling the confidence bands for all IC profiles. The simultaneous confidence band then serves as a control chart for online monitoring of profiles.

2.1. Data cleaning

The identification of unusual observations (or *outliers*) for time series data is important since they can lead to intervention of analysis for the underlying process. For example, the data can be obtained in the form of signals collected by sensors with different time stamps (e.g., the Flight Data Recorder). These signals are often noisy due to inaccurate sensor readings (i.e., measurement error). Thus, in order to provide users (such as engineers or pilots) efficient analysis tools, it is necessary to extract high quality information from these noisy data.

The shortcoming of traditional methods for outlier detection is that they are usually model dependent (Chang, Tiao, & Chen, 1988; Peña, 2001). This type of approaches may not be practical since the underlying time series can be highly nonstationary, thus requiring an extremely high computational cost for model selection. To avoid the problem of model selection, here we introduce an automated data cleaning approach proposed by Basu and Meckesheimer (2007), called the *two-sided median method*. It is noted that this method is easy to implement, model independent, and computationally much cheaper than the model-based approaches. Its basic idea is described below.

Given a time series $y_1, y_2, \ldots, y_n$, the neighborhood of a particular observation y_i is defined as the set $\{y_{i-k}, \ldots, y_{i-1}, y_{i+1}, \ldots, y_{i+k}\}$, where $2k$ is the size of the neighborhood window. Compute the median for the neighborhood of y_i and denote it by $m_i^{(k)}$. Calculate the distance from y_i to $m_i^{(k)}$ and compare it to a pre-specified threshold value τ. If $|y_i - m_i^{(k)}| < \tau$, then the observation y_i is retained; otherwise y_i is identified as an outlier and replaced by $m_i^{(k)}$. With this identification rule, an observation is labeled as an outlier when it is considerably far from the median of its neighborhood. Further, by replacing the identified outliers with more reasonable values, a cleaner time series $y_1^*, y_2^*, \ldots, y_n^*$ is obtained. It should be mentioned here that in general there is no optimal rule for choosing the best values of k (window width) and τ (threshold value). In real applications, these values are often determined based on engineering knowledge. However, for practical purposes here we provide a rough rule of thumb for choosing k and τ:

> Choose first a moderate value of k (e.g., $k = 3$ or 5), then choose a rather large value of τ so that the amount of identified outliers does not exceed 5–7% of the total observations.

Note that based on the above rule of thumb, the noisy data for each profile can be reasonably cleaned, and yet, their primary information can be fairly well preserved at the same time.

Remark. The two-sided median method can also be used to estimate the missing values in a time series.

2.2. Fitting B-spline polynomial models

In order to simplify the notations, in this subsection we denote the cleaned data of a particular profile by $\{y_j, x_j\}$, where $j = 0, 1, \ldots, n$. The next step of our framework is to obtain a smooth function $\hat{f}(x)$ that represents well the relationship between y_j and x_j for each profile so that the underlying function $f(x)$ can be estimated accordingly in later stages. Note that one of the most popular and powerful techniques in nonparametric regression is *spline smoothing* (Hastie & Tibshirani, 1990; Green & Silverman, 1994; Wahba, 1990). A general solution is to choose the function $\hat{f}(x)$ so as to minimize the penalized sums of squares

$$\sum_{j=1}^{n}(y_j-\hat{f}(x))^2+\lambda\int(\hat{f}''(x))^2dx, \tag{2}$$

where λ is a *roughness penalty* (or smoothing parameter) that controls the trade-off between model fidelity and roughness.

The solution $\hat{f}(x)$ in Eq. (2) is a piecewise polynomial with the join points (called *knots*) at a unique set of the explanatory values. If $\hat{f}(x)$ is a cubic polynomial (this is the most common spline in practice) over each interval of (x_0,x_1), …, (x_{n-1},x_n) and it has continuous first and second derivatives (i.e. C^2 continuous) at the knots, then $\hat{f}(x)$ is called a *smoothing spline*. However, the natural cubic spline does not provide an explicit form for $\hat{f}(x)$. In addition, it has the disadvantage that every time when one of the control point changes, the entire curve needs to be recomputed. This means that the implementation of smoothing spline can be computationally expensive. To overcome this problem, we introduce an alternative approach based on constructing a set of basis functions, called B-spline.

Let us denote the knot vector by $\mathbf{x}=(x_0,x_1,\ldots,x_n)$ with the corresponding $n+1$ control points $y_0,\ldots,y_n$. A B-spline of degree d is a parametric curve composed of a linear combination of basis functions $B_{j,d}$, say,

$$\hat{f}(x)=\sum_{j=0}^{n}y_jB_{j,d}(x). \tag{3}$$

The basis functions can be defined and easily computed by using the Cox-de Boor recursion formula (de Boor, 1978, 2001):

$$B_{j,0}(x)=\begin{cases}1 & \text{if } x_j\leqslant x<x_{j+1},\\ 0 & \text{otherwise},\end{cases}$$
$$B_{j,d}(x)=\frac{x-x_j}{x_{j+d-1}-x_j}B_{j,d-1}(x)+\frac{x_{j+d}-x}{x_{j+d}-x_{j+1}}B_{j+1,d-1}(x), \tag{4}$$

where the convention $\frac{0}{0}=0$. Note that $B_{j,d}(x)$ is a polynomial of degree $d-1$ (which is sometimes confusing) and C^{d-2} continuous on each interval $x_j<x<x_{j+1}$. Also, the basis functions have the property that $0\leqslant B_{j,d}(x)\leqslant 1$ for all j and $\sum_{j=0}^{n}B_{j,d}(x)=1$. As can be seen, for any given x there are only d nonzero basis functions. This indicates that the B-spline depends on d nearest control points at any point x. Therefore, it is attractive in the context that, if we wish to recompute the entire spline curve after one control point is changed, then only the terms involving that point need to be removed and recomputed. Such an important feature also makes it particularly useful for real-time monitoring, especially when the number of observations becomes large.

The shape of the basis functions is clearly determined by the position of the knots. If the spacing between the knots is a constant (i.e. $x_{j+1}-x_j\equiv c$), the B-spline is referred to as a uniform B-spline. Note that the basis function for uniform B-splines can be easily calculated, and, it is equal for each polynomial segment. To illustrate, the ith polynomial segment for the most commonly used cubic uniform B-spline (i.e. $d=4$, which refers to cubic polynomials) is given by

$$S_i(x)=[x^3\ x^2\ x\ 1]\cdot\frac{1}{6}\begin{bmatrix}-1 & 3 & -3 & 1\\ 3 & -6 & 3 & 0\\ -3 & 0 & 3 & 0\\ 1 & 4 & 1 & 0\end{bmatrix}\begin{bmatrix}y_{i-1}\\ y_i\\ y_{i+1}\\ y_{i+2}\end{bmatrix} \tag{5}$$

Since equal knot spacing fails to cope with all but the most simplistic of geometries, one may consider nonuniform B-splines that allow any spacing of the knots. However, how to determine the optimal number and position of the knots remains a challenging problem (it is in fact an NP-hard problem). Some related works in literature include: the method called TURBO (Friedman & Silverman, 1989), the Delete-Knot/Cross-Validation method (DKCV) (Breiman, 1993), the reversible jump Markov chain Monte Carlo method (Denison, Mallick, & Smith, 1998), the model selection algorithm based on Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC) (Molinari, Durand, & Sabatier, 2004), and the Simulated-Annealing strategy (Lolive, Barbot, & Boeffard, 2006), just to name a few. Due to the nice computational property previously mentioned for B-splines, in this work we employ the leave-one-out Cross-Validation (CV) strategy for choosing the optimal number of knots. The idea of the leave-one-out CV strategy is introduced as follows (Geisser, 1993; Mosteller & Wallace, 1963; Stone, 1974).

First, the position of the knots can be chosen in a data-adaptive scheme. For example, given any fixed number k, the knots can be placed at suitable quantiles of $\{x_j\}_{j=0}^{n}$. The leave-one-out CV score is then defined as the following mean squared error of prediction (MSEP):

$$\mathrm{MSEP}(k)=\frac{1}{n+1}\sum_{j=0}^{n}(y_j-\hat{f}_{(-j)}(x_j))^2, \tag{6}$$

where $\hat{f}_{(-j)}$ is the obtained cubic B-spline by removing the jth observation (y_j,x_j). Consider a reasonable set of distinct knot numbers, say, $\{k_1,k_2,\ldots,k_N\}$ where $0\leqslant k_i\leqslant n+1$ for all $i=1,\ldots,N$. After calculating the MSEP for each value of k_i, the optimal number of knots is then chosen as

$$k^*=\arg\min_{k\in\{k_1,\ldots,k_N\}}\mathrm{MSEP}(k). \tag{7}$$

2.3. Block bootstrap resampling for dependent data

With all obtained residuals e_{ij} for each profile i, the next step is to construct the confidence band for $\hat{f}_i(x)$ within the data range. This can be done naturally by a *bootstrap resampling* procedure. The traditional bootstrap method, as known first proposed by Efron (1979), was designed to resample from independent data (one at a time) so that the measurement of interest can be estimated based on the empirical distribution of all sampled observations. However, since now we assume the residuals $e_{i1},\ldots,e_{in_i}$ can be correlated for each profile i, the traditional bootstrap method may overestimate (or underestimate) the desired quantity without incorporating the dependence structure of data. To overcome this problem, we introduce a simple and popular version of the bootstrap method for dependent data, called *block bootstrap*. Its basic idea is described as follows.

In the block bootstrap, data are divided into several blocks so that the original dependence structure within a block is preserved. A popular version for this type of resampling method is the *moving block bootstrap* (MBB), for which overlapping blocks of the same length are drawn randomly with replacement. It was shown in extensive studies that the MBB outperforms other methods based on subsequent values presenting high correlation in relatively short periods of observations (see Mignani & Rose, 2001, and references therein). Further, it does not require specific assumptions on the structure of the data generating process. For the MBB with a block length l, the residuals $e_{i1},\ldots,e_{in_i}$ of each profile i can be divided into n_i-l+1 blocks, viz., with block 1 being $\{e_{i1},\ldots,e_{il}\}$, block 2 being $\{e_{i2},\ldots,e_{i(l+1)}\}$,…, etc. Therefore, the bootstrap sample for each profile i is generated by the following mechanism:

$$\{y_{ij}^*,x_{ij}^*\}=\{\hat{f}_i(x_{ij})+e_{i(t+k)},x_{ij}\}, \tag{8}$$

where $j=1,2,\ldots,n_i$, t is generated independently from a uniform random variable on $\{1,2,\ldots,n_i-l+1\}$, and $k=1,2,\ldots,l$.

It is noted that the accuracy of the MBB is sensitive to the choice of block length l. In general, blocks of a shorter length can achieve a better approximation of the underlying distribution, but on the

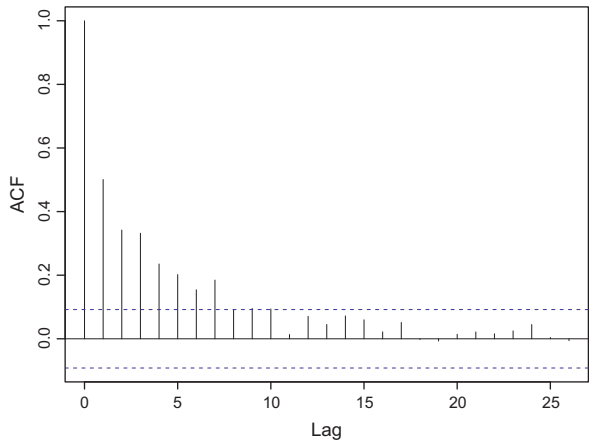

Fig. 1. The illustration of ACF for a time series with size $n = 48$.

other hand they may destroy the structure of short-range (or medium-range) dependence. There are different ways suggested in literature for choosing the optimal block length (Bühlmann & Künsch, 1999; Hall, Horowitz, & Jing, 1995; Künsch, 1989; Lahiri, 1999; Lahiri, Furukawa, & Lee, 2007). Among all the methods, two essential ones are called the plug-in methods and the empirical criterion-based methods. However, these two types of methods may not be practical in the sense that the plug-in methods often require a huge amount of work in order to obtain the theoretical expression for the optimal block length, while the criterion-based methods often require a certain amount of computation. In this work, we introduce a simple method for choosing a "suitable" block length for time series data. The idea is based on examining the diagnostic plot of the sample autocorrelation function (ACF). To illustrate, let us look at the ACF of a time series (with sample size $n = 48$) shown in Fig. 1. As can be seen from Fig. 1, the value of ACF becomes insignificant after lag 3. In this case, the suggested smallest block length is $l = 3 + 1 = 4$ (since the block length is often chosen to be substantially longer than the dependence length), which also agrees with the criterion suggested in Lahiri (1999) (i.e., $l = 4 < n^{1/2} = 48^{1/2} \approx 7$). In summary, the following guideline is suggested for choosing the block length:

> Suppose the sample size is n and $\bar{l}$ is the smallest time lag so that the ACF is not significant with all time lags greater than $\bar{l}$. The suggested block length is then:
>
> $$l^* = \min(\bar{l} + 1, \sqrt{n}).$$

Remark. Since the glue points break the property of stationarity, the data generated by the MBB are nonstationary. However, this problem can be solved by using a method called *stationary bootstrap* (if the original data are stationary), for which the block length l is randomly selected from a geometric distribution (Politis & Romano, 1994). For other methods that can be used to resample from dependent data (such as subsampling, sieve bootstrap, local bootstrap, wild bootstrap, and Markov bootstrap, etc.), the readers can refer to Chernick (1999) and Davison and Hinkley (2006).

2.4. Simultaneous confidence band based on bootstrap curve depths

To construct the simultaneous confidence band for the underlying function $f(x)$, we first establish the bootstrap percentile confidence band of $f(x)$ within the data range of each profile and then glue all the confidence bands together. Suppose for each profile i, the bootstrap sampling process is repeated N times so that a collection of N B-spline curves $C_i^N(x) = \{\hat{f}_i^1(x), \ldots, \hat{f}_i^N(x)\}$ is obtained, where $\min_{j \leqslant n_i} x_{ij} \leqslant x \leqslant \max_{j \leqslant n_i} x_{ij}$. Here the confidence band is constructed based on the method of ranking the "curve depths" in the collection $C_i^N(x)$, which was first proposed by Yeh (1996). Its basic idea, when applied to our analysis, is described as follows.

For a particular bootstrap curve $\hat{f}_i^j(x) \in C_i^N(x)$, its *curve distance* with respect to the baseline curve $\hat{f}_i(x)$ (obtained from the original profile data) can be defined as

$$d_{ij} = d(\hat{f}_i(x), \hat{f}_i^j(x)) = \int_{x_{i1}}^{x_{in_i}} (\hat{f}_i^j(x) - \hat{f}_i(x))^2 dx. \tag{9}$$

The corresponding *curve depth* is then defined as $D_{ij} = (1 + d_{ij})^{-1}$. With this definition, the smaller the curve depth is, the further the curve is located from the benchmark curve. Therefore, to obtain the bootstrap percentile confidence band we can exclude $100\alpha\%$ curves with the lowest depths from the collection $C_i^N(x)$.

Now we describe the detailed steps for constructing the confidence band of $f(x)$ within the data range of each profile i. Let $D_{i(1)} \leqslant D_{i(2)} \leqslant \cdots \leqslant D_{i(N)}$ be the sorted curve depths D_{ij} for all the bootstrap curves in $C_i^N(x)$. For any given $0 < \alpha < 1$, define the collection $C_{i,1-\alpha}^N(x) = \{\hat{f}_i^{(j)}(x) : \alpha N \leqslant j \leqslant N\}$, where $\min_{j \leqslant n_i} x_{ij} \leqslant x \leqslant \max_{j \leqslant n_i} x_{ij}$. The $100(1-\alpha)\%$ bootstrap percentile confidence band for $f(x)$ within the data range of profile i is then given by

$$B_{i,1-\alpha}^N(x) = \{(x,y) : \text{For each fixed } x, \min_j \hat{f}_i^j(x) \leqslant y \leqslant \max_j \hat{f}_i^j(x)\}, \tag{10}$$

where $\hat{f}_i^j(x) \in C_{i,1-\alpha}^N(x)$ and $\min_{j \leqslant n_i} x_{ij} \leqslant x \leqslant \max_{j \leqslant n_i} x_{ij}$. Since we assume all the profiles are independent, it is natural to pool all the confidence bands $B_{1,1-\alpha}^N(x), B_{2,1-\alpha}^N(x), \ldots, B_{M,1-\alpha}^N(x)$ so as to obtain the confidence band of $f(x)$ over the entire data space. Thus, the simultaneous confidence band of $f(x)$ is given by

$$B_{1-\alpha}(x) = \{(x,y) : \text{For each fixed } x, \min_{i,j} \hat{f}_i^j(x) \leqslant y \leqslant \max_{i,j} \hat{f}_i^j(x)\}, \tag{11}$$

where $\hat{f}_i^j(x) \in C_{i,1-\alpha}^N(x), 1 \leqslant i \leqslant M$, and $1 \leqslant j \leqslant n_i$.

2.5. Algorithm for framework implementation

Note that the above procedures, which include data cleaning, fitting B-spline models, block bootstrap resampling, and construction of the simultaneous confidence band, basically constitute the Phase I analysis of profile monitoring. For implementation purpose, we summarize the steps of these procedures in the following algorithm:

Step 1: Clean each profile data by using the two-sided median method.
Step 2: Obtain the B-spline curve $\hat{f}_i(x)$ for each profile i based on the procedure introduced in Section 2.2.
Step 3: Generate N bootstrap samples $\{y_{ij}^*, x_{ij}^*\}$ for each of retained profile i based on the procedure introduced in Section 2.3.
Step 4: For each profile i, obtain the B-spline curve $\hat{f}_i^j(x)$ based on each generated bootstrap sample and compute its curve depth D_{ij} with respect to the benchmark curve $\hat{f}_i(x)$. Obtain the sorted curve depths $D_{i(j)}$ and identify the collection of curves $C_{i,1-\alpha}^N(x)$ for a given $0 < \alpha < 1$.
Step 5: Construct the $100(1-\alpha)\%$ bootstrap percentile confidence band $B_{i,1-\alpha}^N(x)$ within the data range of each profile i by using Eq. (10). Obtain the simultaneous confidence band for $f(x)$ over the entire data space by using Eq. (11).

2.6. Online monitoring based on curve matching

The confidence region $B_{1-\alpha}(x)$ in Eq. (11) can simply serve as a control chart for online monitoring of time-ordered profiles in Phase II. To do this, for any new profile we observe, at any point in time t a B-spline model has to be established based on the cur-

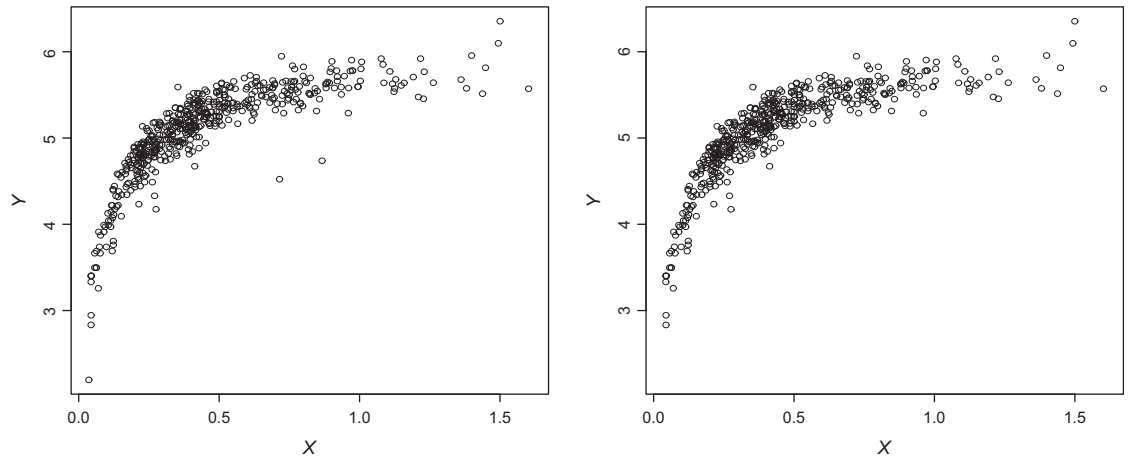

Fig. 2. (Left panel): The scatter plot for the 20 selected IC profiles. (Right panel): The scatter plot for the 20 selected IC profiles after data cleaning.

rent observations. If the obtained B-spline model (denoted by $\hat{f}(x_t)$) falls completely within the confidence region (i.e., $\hat{f}(x_t) \subset B_{1-\alpha}(x)$ for all x_t such that $\min_{i,j} x_{ij} \leqslant x_t \leqslant \max_{i,j} x_{ij}$), then the profile is considered as "in-control" at time t. Otherwise, it is considered as "out-of-control". However, such an online monitoring scheme can induce huge amounts of computation since a new search for the optimal number of knots (i.e., k^*) is necessary for obtaining the desired B-spline model (see Section 2.2) when a new data point is observed. To overcome this computational issue, we suggest choosing k^* based on the concept of "curve matching". Its basic idea is introduced below.

Let $(y_1, x_1), \ldots, (y_{n(t)}, x_{n(t)})$ be the observed profile data at any given point in time t. The IC curve that "best matches" the observed data is given by

$$i(t) = \arg\min_{i=1,\ldots,M} \sum_{j=1}^{n(t)} (y_j - \hat{f}_i(x_j))^2, \tag{12}$$

where $\hat{f}_i$ is the B-spline model for the ith IC profile obtained from the earlier stage. Let us denote the optimal value of k^* for establishing $\hat{f}_{i(t)}$ by $k^*_{i(t)}$. Since the $i(t)$th IC curve best matches the observed data up to time t, the optimal number of knots for fitting the desired B-spline model $\hat{f}(x_t)$ can be chosen based on the following rule:

If $n(t) > k^*_{i(t)}$, choose $k^* = k^*_{i(t)}$; otherwise choose $k^* = n(t)$.

It should be mentioned here that, the proposed curve matching procedure for finding k^* is computationally much cheaper than performing consecutive empirical searches of k^* over time, especially when the number of observations becomes large.

Remark. In practice, one may require a numerical way to check if a particular B-spline curve $\hat{f}(x_t)$ completely lies within the confidence band $B_{1-\alpha}(x)$. To do this, one can superimpose fine grids on the input domain of the explanatory variable and then check if the corresponding measures of response on the B-spline curve exceed the boundaries of the confidence band.

3. Performance evaluation: a real example

In this section, the performance of our proposed framework is evaluated by a real data set from hospitals in Taiwan. We first introduce the data, some numerical results are presented afterwards.

3.1. Introduction to the data

The AIDS cohort data, which were collected between January 1990 and January 2003, include the information of clinical, biochemical, serologic, and histologic parameters of 1054 HIV-in-

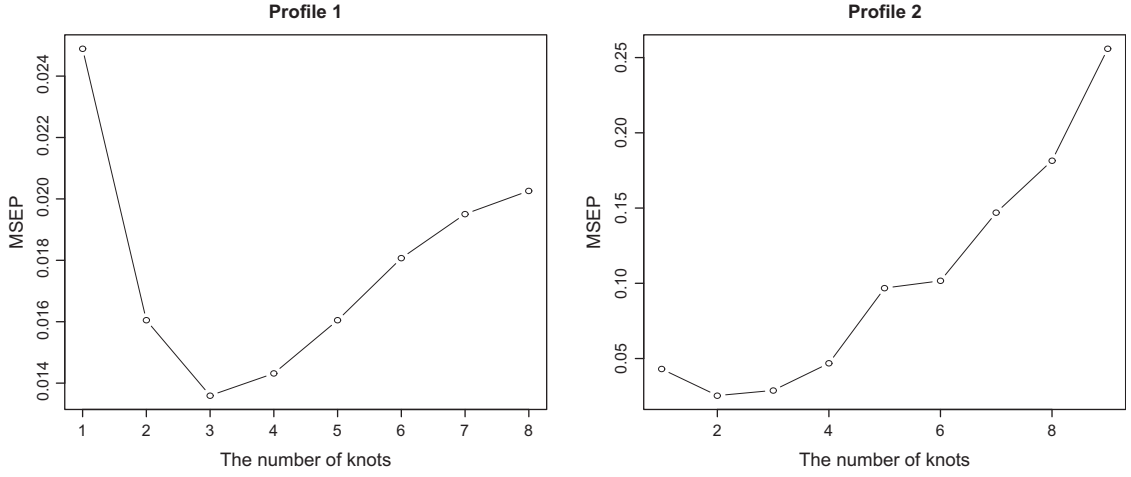

Fig. 3. The MSEP versus the selected number of knots for profiles 1 and 2.

fected patients in Taiwan. All patients were advised to return to hospital every three or four months for a follow-up diagnosis. The primary goal of collecting such a data set is to evaluate the efficacy of the highly active antiretroviral therapy (HAART), which consists of at least three anti-HIV drugs (for a detailed description of this data set, please refer to Shu & Tseng (2009)). To implement the proposed framework, we select two important variables from the data set:

Y: The CD4 cell count (per cubic millimeter of blood) in log scale.

X: The ratio of CD4 cell count to CD8 cell count.

We divide the patients into two groups. The patients who took the therapy HAART are categorized as the "in-control" (IC) profiles. On the other hand, the patients who did not take the therapy HAART are categorized as the "out-of-control" (OC) profiles. To establish the confidence region for the underlying functional relationship between X and Y, we randomly select 20 IC profiles (patients) from the data. The scatter plot for these 20 selected IC profiles is given in the left panel of Fig. 2.

As can be seen from the left panel of Fig. 2, there exists a very clear functional relationship between Y and the explanatory variable X. In addition, it shows that there may be some potential outliers in this particular data set. Therefore, we utilize the two-sided median method to clean the noise. According to the rule of thumb suggested in Section 2.1, here we choose $k = 5$ and $\tau = 0.54$ in the two-sided median method. Further, the implementation of the method yields two potential outliers, $(X, Y) = (0.72, 4.52)$ and $(0.87, 4.74)$. With the two potential outliers being replaced by the medians in the neighborhood, the cleaned data are shown in the right panel of Fig. 2.

3.2. Results of framework implementation – Phase I

We next establish the B-spline model for each of the 20 cleaned IC profiles. In order to obtain an adequate B-spline model for each of the profiles, we consider a set of possible numbers of knots $\{0, 1, \ldots, 10\}$ and compute the corresponding MSEP for each number using the leave-one-out CV. To make the results amenable to visualization, the diagnostic plots for two illustrative profiles (profiles 1 and 2) are shown in Fig. 3. As can be seen from Fig. 3, the optimal numbers of knots (i.e., k^*) that minimize the MSEP for profile 1 and profile 3 are given by 3 and 2, respectively. Based on all the diagnostic plots, the optimal number of knots for all the 20 IC profiles are given by 3, 1, 2, 2, 6, 1, 1, 1, 1, 9, 2, 5, 11, 2, 3, 1, 1, 2, 1, and 1, respectively. The resulting fitted B-spline curves (i.e. $\hat{f}_i(x)$) based on the best selected number of knots for profiles 2, 5, 7, and 10 are shown in Fig. 4.

We next examine the plot of ACF for the residuals based on the fitted B-spline curve for each IC profile. The goal of this step is to choose a suitable block size so as to conduct an adequate bootstrap sampling procedure for each profile data. Note that Fig. 1 in fact shows the plot of ACF for profile 11. For additional illustrations, the plots of ACF for profiles 2, 5, 7, and 10 are given in Fig. 5. As we can see from Fig. 5, the residuals of all profiles reveal a small range of dependence over time (in fact this is true for all profiles). In particular, the ACFs for profiles 2 and 10 are insignificant for all

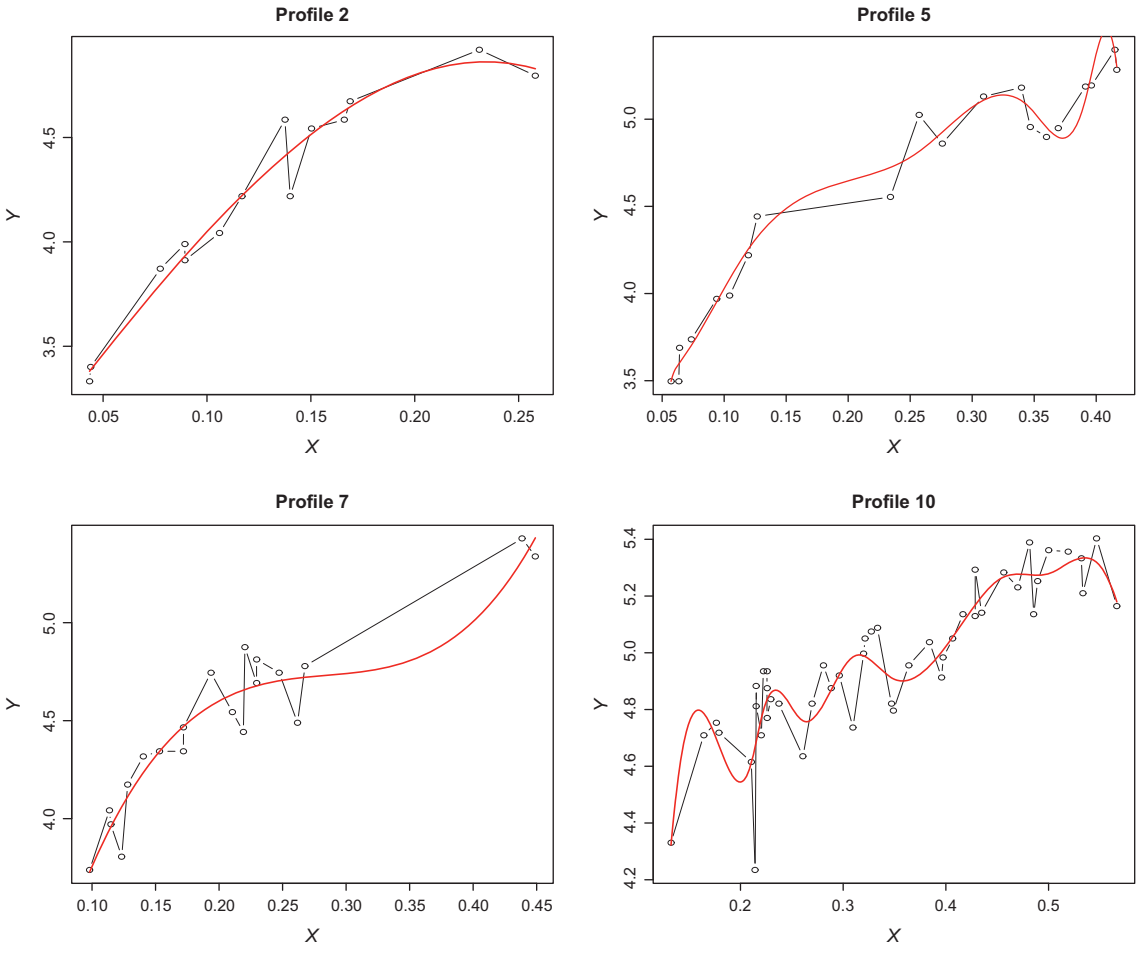

Fig. 4. The illustration of fitted B-spline curves for profiles 2, 5, 7, and 10.

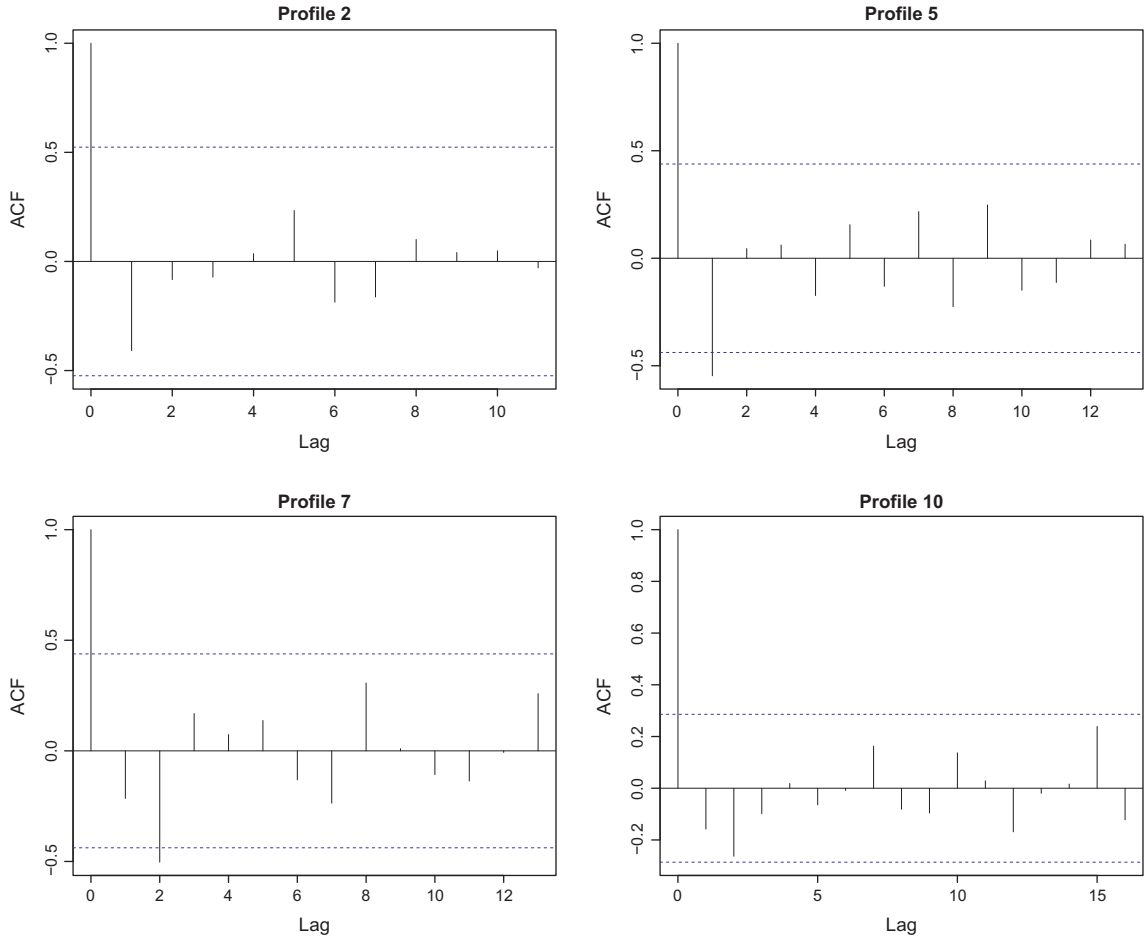

Fig. 5. The ACF plots of residuals for profiles 2, 5, 7, and 10.

lags greater than zero, whereas the ACFs for profiles 5 and 7 become insignificant after lag 1 and 2, respectively (note that the ACF always has the value one at lag zero). These suggest that the suitable block size for profiles 2 and 10 is one (which then reduces to the primary bootstrap method), while the suggested block sizes for profiles 5 and 7 are 2 and 3, respectively. Based on all the diagnostic plots, the suggested block sizes for all the 20 IC profiles are given by 4, 1, 1, 1, 2, 1, 3, 1, 5, 1, 4, 2, 6, 1, 1, 1, 1, 2, 2, and 1, respectively.

Based on the obtained block sizes, the MBB method (or bootstrap method) is then used to resample from the residuals of each profile. Thus, for each generated bootstrap sample we can obtain a fitted B-spline curve that represents the relationship between X and Y. Fig. 6 shows 20 B-spline curves obtained from the MBB method for profiles 2, 5, 7, and 10. As can be seen from Fig. 6, each experiment reveals to have a fairly uniform and tight confidence band (in fact this is true for all profiles), except for the locations where data are sparse or near the boundaries (this might be due to the feature of fitting B-spline models). The result also provides a strong evidence that each of the obtained B-spline models represents fairly well the underlying functional relationship within the corresponding data range.

To construct the confidence band $B^N_{i,1-\alpha}(x)$ for each profile, the MBB method is repeated 10^4 times (i.e. $N = 10^4$) so that 10^4 bootstrap curves are obtained. Eq. (9) is then used to compute the curve depth for each bootstrap sample. By deleting 5% of curves based on the sorted curve depths for each profile, the resulting 95% simultaneous confidence bands $B_{0.95}(x)$ over the entire data space are given in Fig. 7a. Note that for comparison purpose, the simultaneous confidence band based on the traditional bootstrap method (i.e., block size $\equiv 1$) is shown in Fig. 7b.

We summarize some remarkable findings in Fig. 7. First, as can be seen from panel (a), the confidence band is promising since all the B-spline curves for the 20 IC profiles completely lie within the boundaries (i.e., coverage probability = 1.00 for these 20 IC profiles). Second, the simultaneous confidence band is not particularly smooth (especially for the locations where two or more confidence bands are pooled together) and does not have homogeneous widths over the entire data range. This is likely due to inconsistency of the correlation structures for different profiles. Last, panel (b) shows that the confidence band obtained by the traditional bootstrap method has a very similar shape to that obtained by our method, of which the result is likely due to the small block sizes chosen in the MBB method. In fact, numerical results show that the confidence band obtained by the traditional bootstrap method is a bit wider in the area of $0.5 < X < 0.8$ and a bit narrower in the area of $0.8 < X < 1.4$, within which the corresponding profile residuals are negatively correlated and positively correlated, respectively.

3.3. Online profile monitoring – Phase II

To evaluate the effectiveness of our proposed framework, the online monitoring procedure is performed for 50 randomly selected IC and OC profiles from the original data set. At any point in time t, a B-spline model is fitted to the observed profile data

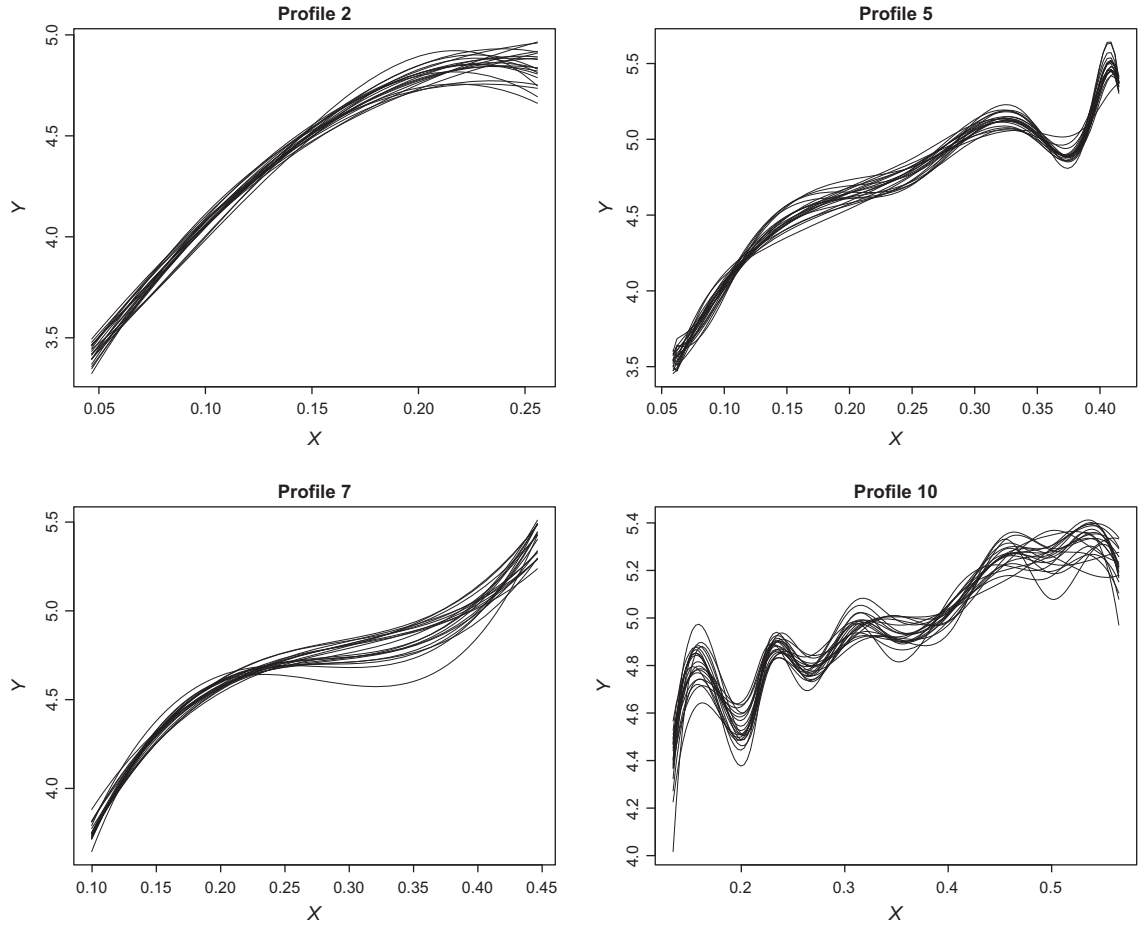

Fig. 6. The illustration of 20 bootstrap B-spline curves for profiles 2, 5, 7, and 10.

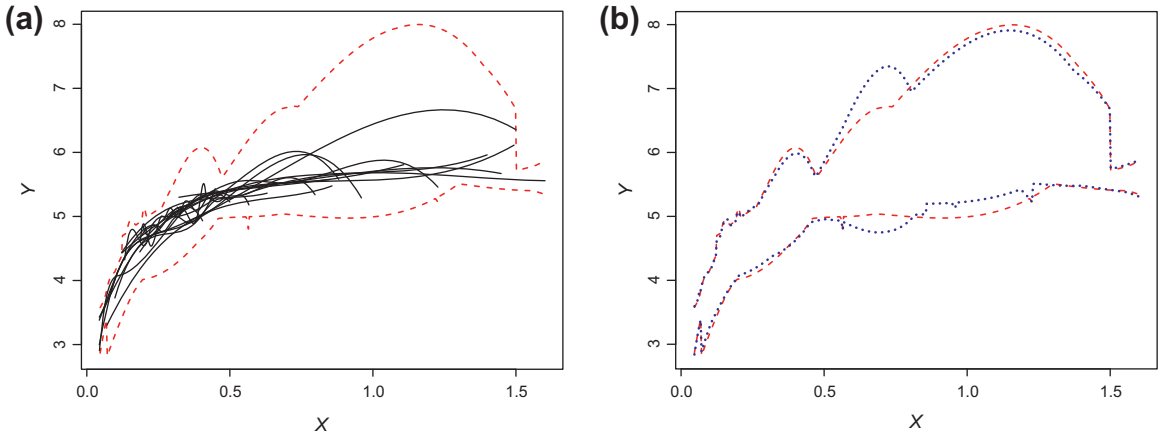

Fig. 7. (a) The simultaneous confidence band $B_{0.95}(x)$ (dash lines) obtained by the MBB method based on the curve distance in Eq. (9) along with the B-spline curves (solid lines) for the 20 IC profiles. (b) The simultaneous confidence bands $B_{0.95}(x)$ obtained by the MBB method (red dash lines) and by the traditional bootstrap method (blue dotted lines) based on the curve distance in Eq. (9). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

with the number of knots chosen based on the rule suggested in Section 2.6. For visualization purpose, Fig. 8 illustrates the results for monitoring four profiles using the previously obtained confidence band based on Eq. (9) (i.e., the confidence band in panel (a) of Fig. 7), of which two are identified as IC profiles and two are identified as OC profiles. It is noted that an OC signal here indicates the detection of HIV infection in high-risk patients.

We next compare the performance of our proposed framework with other methods in terms of the following measures: (i) Average Run Length (ARL) to false alarm, i.e., the average number of observed IC profiles before an OC signal is generated; (ii) ARL to true alarm, i.e., the average number of observed OC profiles before an OC signal is generated; (iii) Standard Deviation of Run Length (SDRL), the standard deviation for the number of observed profiles

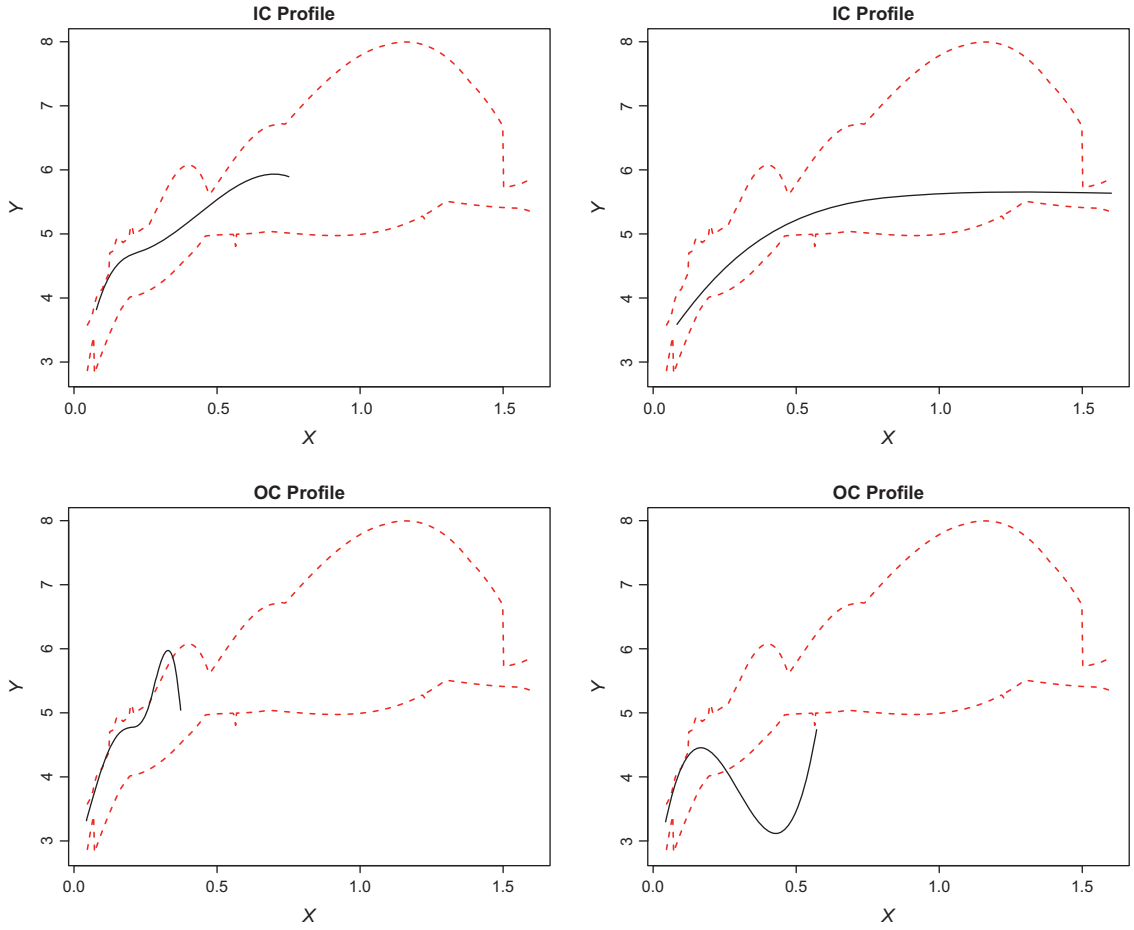

Fig. 8. An illustration of four fitted B-spline curves (solid lines) along with the confidence band $B_{0.95}(x)$ (dash lines) based on the curve distance in Eq. (9). Note that the curves on the top left and top right panels are identified as IC profiles, whereas the curves on the bottom left and bottom right panels are identified as OC profiles.

Table 1
The performance measures for the online monitoring of 50 randomly selected IC and OC profiles based on three different methods.

Performance measures	Our method	The method by Zou et al.	The method by Jensen and Birch
ARL (to false alarm)	7.14	1.25	8.33
ARL (to true alarm)	1.85	1.19	4.55
SDRL	3.38	1.60	2.40
ATS	7.44	7.00	7.36

to true and false alarm; and (iv) Average Time to Signal (ATS), the average number of observations to signal an OC profile. Due to limited computational resources, here we consider two benchmark methods to carry out the numerical comparisons. The first method to be compared is the nonparametric regression approach introduced by Zou et al. (2008), wherein a standard Gaussian kernel function is selected to construct the local linear smoother and the error terms e_{ij} are assumed to be iid normal random variables (thus the within-profile correlation is not taken into account). The second method to be compared is the nonlinear mixed (NLM) models introduced by Jensen and Birch (2009), wherein a logistic model is selected and correlation within the profile is also incorporated. The numerical results based on the 50 randomly selected IC and OC profiles are summarized in Table 1.

The numerical results in Table 1 reveal several interesting findings. First, the method by Jensen and Birch classifies well the IC profiles (ARL to false alarm = 8.33) but misclassifies a large proportion of the OC profiles (ARL to true alarm = 4.55). This may be due to the fact that the method incorporates inadequate correlation structures into analysis so that a rather "conservative" (large) confidence region is obtained. Second, the method by Zou et al. classifies fairly well the OC profiles (ARL to true alarm = 1.19) but misclassifies a large proportion of the IC profiles (ARL to false alarm = 1.25). This may be due to the fact that the method totally ignores the correlation structure within the profile so that a rather "tight" (small) confidence region is obtained. Third, our method classifies fairly well both the IC profiles (ARL to false alarm = 7.14) and the OC profiles (ARL to true alarm = 1.85). It is known that a good monitoring scheme should result in a large ARL to false alarm and a small ARL to true alarm. Therefore, if the ARLs to both false alarm and true alarm are considered, our method appears to have the best overall performance among all the methods in this study (though the associated SDRL is shown to be a bit larger than the other two methods). Finally, our method results in a larger value of ATS (i.e., slower on generating OC signals) for this particular data set, which indicates a tradeoff between the computational speed and accuracy.

4. Summary and concluding remarks

We proposed a general framework for monitoring nonparametric profiles in both Phase I and Phase II. The framework is mainly divided into five steps in Phase I. In Step 1, an automated approach (called the two-sided median method) is used to clean each profile data. In Step 2, an adequate B-spline model is fitted to each profile

data. In Step 3, the moving block bootstrap method is used to generate dependent samples for each profile. In Step 4, the B-spline model is fitted to each of the bootstrap sample and its corresponding curve depth is calculated. In Step 5, the B-spline curves with smaller curve depths are removed and, the resulting confidence bands of all profiles are pooled so as to obtain a simultaneous confidence band of the underlying functional relationship over the entire data space. The obtained confidence band in Phase I can be used to monitor nonparametric profiles in Phase II. The numerical results show that our framework is effective in detecting unobserved profiles in terms of average run length (ARL), compared to two benchmark methods shown in literature.

Here we highlight some potential problems for future research studies. First, in practice any nonparametric modeling technique with adequately chosen tuning parameters can be applied in Step 2 of our proposed framework. However, one needs to take into account the computational cost (especially when the number of profiles/observations is large) and how to best compare the confidence bands obtained from different modeling techniques. Second, we can develop other control charts for monitoring nonparametric profiles in real time based on the obtained simultaneous confidence band. For example, we can consider the centerline of the confidence band as a baseline function (i.e. an estimate of $f(x)$) and then monitor the change of residuals (or average curve distances) for a new profile over time. However, how to best incorporate the dependence structure of the observations into the development of such a control chart needs to be further investigated. Third, the proposed framework can be properly extended to monitor nonparametric profiles in high dimensional data spaces. However, one may require a more efficient modeling technique to best describe the underlying functional relationship. Finally, incorporating common-cause variation between profiles into the control chart scheme is sometimes necessary in real applications. This is obviously a more challenging task since one needs a new model (parametric or semi-parametric model) that takes into account both the intra-profile and inter-profile correlations.

References

Basu, S., & Meckesheimer, M. (2007). Automatic outlier detection for time series: An application to sensor data. *Knowledge and Information Systems, 11*, 137–154.

Breiman, L. (1993). Fitting additive models to regression data. *Computational Statistics and Data Analysis, 15*, 13–46.

Bühlmann, P., & Künsch, H. R. (1999). Block length selection in the bootstrap for time series. *Computational Statistics and Data Analysis, 31*, 295–310.

Chang, I., Tiao, G. C., & Chen, C. (1988). Estimation of time series parameters in the presence of outliers. *Technometrics, 30*, 193–204.

Chang, S. I., & Yamada, S. (2010). Statistical process control for monitoring nonlinear profiles using wavelet filtering and B-spline approximation. *International Journal of Products Research, 48*, 1049–1068.

Chernick, M. R. (1999). *Bootstrap methods. A practitioner's framework*. Wiley Series in Probability and Statistics.

Chicken, E. (2011). Nonparametric nonlinear profiles. In R. Noorossana, A. Saghaei, & A. Amiri (Eds.), *Statistical analysis of profile monitoring*. Wiley Series in Probability and Statistics.

Chicken, E., Pignatiello, J. J., Jr., & Simpson, J. R. (2009). Statistical process monitoring of nonlinear profiles using wavelets. *Journal of Quality Technology, 41*, 198–212.

Colosimo, B. M., & Pacella, M. (2007). On the use of principal component analysis to identify systematic patterns in roundness profiles. *Quality and Reliability Engineering International, 23*, 707–725.

Davison, A. C., & Hinkley, D. (2006). *Bootstrap methods and their application* (8th ed.). Cambridge: Cambridge Series in Statistical and Probabilistic Mathematics.

de Boor, Carl (1978). *A practical guide to splines*. Springer-Verlag.

de Boor, Carl (2001). *A practical guide to splines* (revised ed.). Springer-Verlag.

Denison, D. G. T., Mallick, B. K., & Smith, A. F. (1998). Automatic Bayesian curve fitting. *Journal of the Royal Statistical Society, Series B, 60*, 333–350.

Ding, Y., Zeng, L., & Zhou, S. (2006). Phase I analysis for monitoring nonlinear profiles in manufacturing processes. *Journal of Quality Technology, 38*, 199–216.

Efron, B. (1979). Bootstrap methods: Another look at the jackknife (with discussion). *Annals of Statistics, 7*, 1–26.

Friedman, J. H., & Silverman, B. W. (1989). Flexible parsimonious smoothing and additive modeling (with discussion). *Technometrics, 31*, 3–39.

Geisser, S. (1993). *Predictive inference*. New York: Chapman and Hall.

Green, P. J., & Silverman, B. W. (1994). *Nonparametric regression and generalized linear models: A roughness penalty approach*. Chapman and Hall.

Hall, P., Horowitz, J. L., & Jing, B-Y. (1995). On blocking rules for the bootstrap with dependent data. *Biometrika, 82*, 561–574.

Hastie, T. J., & Tibshirani, R. J. (1990). *Generalized additive models*. Chapman and Hall.

Hung, Y. C., Tsai, W. C., Yang, S. F., Chuang, S. C., & Tseng, Y. K. (2012). Nonparametric profile monitoring in multi-dimensional data spaces. *Journal of Process Control, 22*, 397–403.

Jensen, W. A., & Birch, J. B. (2009). Profile monitoring via nonlinear mixed models. *Journal of Quality Technology, 41*, 18–34.

Künsch, H. R. (1989). The jackknife and the bootstrap for general stationary observations. *Annals of Statistics, 17*, 1217–1241.

Lada, E. K., Lu, J.-C., & Wilson, J. R. (2002). A wavelet-based procedure for process fault detection. *IEEE Transactions on Semiconductor Manufacturing, 15*, 79–90.

Lahiri, S. N. (1999). Theoretical comparisons of block bootstrap methods. *Annals of Statistics, 27*, 386–404.

Lahiri, S. N., Furukawa, K., & Lee, Y.-D. (2007). A nonparametric plug-in rule for selecting optimal block lengths for block bootstrap methods. *Statistical Methodology, 4*, 292–321.

Lolive, D., Barbot, N., & Boeffard, O. (2006). Melodic contour estimation with B-spline models using a MDL criterion. In *Proceedings of the 11th international conference on speech and computer (SPECOM), Saint Petersburg, Russia* (pp. 333–338).

Mignani, S., & Rose, R. (2001). Markov Chain Monte Carlo in statistical mechanics: The problem of accuracy. *Technometrics, 43*, 347–355.

Molinari, N., Durand, J. F., & Sabatier, R. (2004). Bounded optimal knots for regression splines. *Computational Statistics & Data Analysis, 45*, 159–178.

Mosteller, F., & Wallace, D. L. (1963). Inference in an authorship problem. *Journal of the American Statistical Association, 58*, 275–309.

Noorossana, R., Saghaei, A., & Amiri, A. (2011). *Statistical analysis of profile monitoring*. Wiley Series in Probability and Statistics.

Peña, D. (2001). Outliers, influential observations, and missing data. In *A course in time series analysis* (pp. 136–170). New York: Wiley.

Politis, D. N., & Romano, J. P. (1994). The stationary bootstrap. *Journal of the American Statistical Association, 89*, 1303–1313.

Qiu, P., Zou, C., & Wang, Z. (2010). Nonparametric profile monitoring by mixed effects modeling. *Technometrics, 52*, 265–277.

Shu, K. N., & Tseng, Y. K. (2009). A semiparametric extended hazard model with time dependent covariates. In *Proceedings of the joint statistical meetings, Washington, DC* (pp. 831–843).

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society, Series B, 36*, 111–147.

Wahba, G. (1990). *Spline models for observational data*. Philadelphia: SIAM.

Yeh, A. B. (1996). Bootstrap percentile confidence bands based on the concept of curve depth. *Communications in Statistics – Simulation and Computation, 25*, 905–922.

Zhang, H., & Albin, S. (2009). Detecting outliers in complex profiles using a χ^2 control chart method. *IIE Transactions, 41*, 335–345.

Zou, C., Qiu, P., & Hawkins, D. M. (2009). Nonparametric control chart for monitoring profiles using the change point formulation and adaptive smoothing. *Statistica Sinica, 19*, 1337–1357.

Zou, C., Tsung, F., & Wang, Z. (2008). Monitoring profiles based on nonparametric regression methods. *Technometrics, 50*, 512–526.