Nonparametric multivariate control chart for numerical and categorical variables

Jiayun Jin*

Geert Loosveldt

Catholic University of Leuven, Leuven, Belgium

Abstract

Multivariate statistical process control (MSPC) was developed for the monitoring of variables that are either all numerical or all categorical. In the present paper, we describe a nonparametric control scheme that can be used to monitor a mixture of numerical and categorical variables simultaneously. It integrates Principal Component Analysis Mix (PCA Mix), a multivariate statistical tool, with the conventional Hotelling T^2 chart. To estimate the control limit for the PCA Mix based T^2 statistic, two nonparametric approaches – kernel density estimation (KDE) and bootstrap – are employed, because of the unknown nature of the underlying distribution. The simulation results demonstrate that with an appropriate number of principal components, both bootstrap and KDE exhibit convincing performance in terms of generating the same, or nearly the same, number of false alarms (ARL_0) as expected, and being able to detect process shifts efficiently (ARL_1). Compared with bootstrap, KDE is shown to work better with small sample sizes ($n < 800$) and to be slightly more sensitive to

*Contact: jiayun.jin@kuleuven.be, Catholic University of Leuven, Centre for Sociological Research, Parkstraat 45, bus 3601, 3000 Leuven, Belgium

small shifts. However, the results also show the instability of the estimated non-parametric control limit when highly imbalanced categorical variables are included, which indicates the need for further research on this topic.

Keywords: PCA Mix; Hotelling T^2 chart; Kernel density estimation; Bootstrap; Average run length

1 Introduction

Statistical Process Control (SPC) has been widely used for monitoring the performance of a process, in order to assess the process stability and ensure the product quality. Out of its available tools, the control chart is the most important and most widely used. Although initially applied in industrial fields, the control chart has found uses in many non-industrial fields, such as public health (see the review article by Thor et al. 2007), finance (e.g., Kovářık and Sarga 2014; Bodnar and Schmid 2011; Bilson et al. 2010), and even surveys (Jin and Loosveldt 2020; Jin et al. 2019; Sirkis et al. 2011). It calculates control limits using statistical equations, and graphically presents the fluctuations of the quality characteristic of a process. A process is defined as statistically "in control" if the fluctuations fall within the control limits, otherwise it is defined as "out of control."

Univariate control charts are used to monitor a process that is characterized by a single variable related to quality. In cases where a process is characterized by more than one variable, the multivariate statistical process control (MSPC) scheme is necessary in order to allow the simultaneous monitoring of multiple variables. Research into MSPC began with the pioneering work of Hotelling (1947), who introduced the T^2 statistic, which measures the distance from an observation to the multivariate mean of the sample. Based on this statistic, multivariate observations characterized by multiple variables can be plotted on a single chart, now known as the Hotelling T^2 chart (T^2 chart). Many multivariate control charts have subsequently emerged – making MSPC one of the fastest growing areas of SPC – such as the multivariate cumulative sum (MCUSUM) control chart (Crosier 1988), and the multivariate exponentially weighted moving average (MEWMA) control chart (Lowry et al. 1992). A recent review of multivariate nonparametric control charts with comparisons of their performance with each other is given by Sofikitou and Koutras (2020).

The variables that most existing multivariate control charts are designed to

monitor are either purely numerical or purely categorical. With regard to monitoring numerical variables, recent developments of multivariate control charts have been motivated by the need to deal with the characteristics inherent in the numerical variables, such as non normality (Capizzi 2015), auto-correlation (Khusna et al. 2018; Pirhooshyaran and Niaki 2015), and high dimensionality (Gunaratne et al. 2017). When treating categorical variables, most of the papers in relevant literature consider multivariate Poisson (Aslam et al. 2017; Aparisi et al. 2014; Chiu and Kuo 2007) and binomial distributions (Chiu and Kuo 2010).

To date, however, there have been surprisingly few studies exploring the use of "mixed" control charts for simultaneously monitoring a mixture of numerical and categorical variables (mixed variables), instead of treating them separately (Bersimis et al. 2018). An exception is Ding et al. (2016), who proposed a standardized-rank-based MEWMA control chart for monitoring mixed data, based on the assumption that the attribute levels of a categorical variable is determined by a latent continuous variable.

Another way to deal with the mixed data in MSPC framework is to use a multivariate tool, termed PCA Mix (Chavent et al. 2014), as a preprocessing tool to transform the mixed data into a set of numerical principal component scores. The T^2 statistic can then be calculated based on the obtained principal component scores. One problem that arises is that unlike the scores based on standard principal component analysis (PCA), the scores based on PCA Mix do not follow any known family of distribution (Ahsan et al. 2018). Among the limited research that has integrated the use of PCA Mix with the T^2 chart, Ahsan et al. (2018, 2021) employed kernel density estimation (KDE) whereas Jin and Loosveldt (2020) employed the bootstrap method to estimate a certain percentile (e.g., the 99^{th} percentile) of the PCA Mix-based T^2 distribution to use as the control limit of their control charts.

Nonparametric methods such as KDE and bootstrap enable us to avoid making any assumptions about the underlying distribution of the PCA Mix-based

T^2 statistic, but using these methods also means that the results (estimates of control limits) can be influenced by the size of the sample that is used. In the meantime, an obvious – yet unanswered – question is: which approach works better, KDE or bootstrap?

Based on the aforementioned considerations, in the current study we first present a procedure that integrates PCA Mix and nonparametric methods – KDE and bootstrap approaches – into the T^2 chart in order to simultaneously monitor mixed variables. First, simulation studies are conducted to examine the impact of the sample size on the variability of the estimated control limits for the PCA Mix-based T^2 statistic. The aim is to provide the practitioner with some guidance as to how large a sample size is reuqired in order to ensure an acceptable level of variability of the estimated control limits when non-parametric methods are used. Second, in other simulation studies, we compare the performance of the two nonparametric control limits (based on KDE and on bootstrap) with each other. To this end, we employ two indicators that have been typically used to quantify performance of control charts – the in-control average run length (ARL_0) and out-of-control average run length (ARL_1) – and conduct two simulation studies to calculate each respectively.

Essentially, run length is the number of observations that need to be plotted before the first out-of-control signal is detected by a control chart. ARL_0 is the expected run length when the process is actually *in control*. Any out-of-control signal in this situation is therefore a false alarm. What ARL_0 measures is the "time" (in terms of in-control observations) a control chart takes before the first false alarm is triggered. Ideally it reflects the false alarm rate that is specified by a user when the control chart is constructed. By comparison, ARL_1 counts the average run length given that the process is *out of control* (e.g., a shift in the process mean or process variance), and therefore evaluates how quickly the control limit is expected to be able to detect a process shift. The calculations of the two indicators are further detailed in Section 2.3.

The remainder of the current paper is organized as follows. In Section 2, we describe the PCA Mix-based T^2 chart, its control limit established based on bootstrap and KDE approaches, and the complete charting procedure. Section 3 presents the results of the simulation studies, and the final section contains our conclusions.

2 Methods

2.1 PCA Mix procedure

Suppose that we have a set of n in-control observations on k_1 continuous variables and k_2 categorical variables, comprising respectively a continuous dataset $\mathbf{X_1}(n \times k_1)$ and a categorical dataset $\mathbf{X_2}(n \times k_2)$. The continuous $\mathbf{X_1}$ is standardized to $\mathbf{Z_1}$ by scaling the variables to unit variance and centering them to zero mean. To preprocess the categorical $\mathbf{X_2}$, we build an $n \times s$ indicator matrix $\mathbf{G}$, with s being the total number of response categories for all the categorical variables. The elements of $\mathbf{G}$ are 1s if the response belongs to the corresponding category of the variable, and 0s otherwise. For example, taking the first categorical variable, if the i^{th} observation falls within the j^{th} category of the variable, $\mathbf{G_{i,j}}$ is 1, otherwise it is 0. $\mathbf{G}$ is then column-centered by subtracting the column mean from each column vector to yield matrix $\mathbf{Z_2}$. The output of the preprocessing is an $n \times (k_1 + s)$ matrix $\mathbf{Z}$, formed by merging the columns of $\mathbf{Z_1}$ and $\mathbf{Z_2}$. The next step is to transform $\mathbf{Z}$ into components, by using the Generalized Singular Value Decomposition (GSVD) technique.

The GSVD for $\mathbf{Z}$ involves assigning weights to the rows and columns of $\mathbf{Z}$. These weights are respectively expressed in two diagonal matrices, $\mathbf{N}$ and $\mathbf{M}$, in such a way that the rows of $\mathbf{Z}$ are weighted by $\frac{1}{n}$, the first k_1 columns are weighted by 1, and the last s columns are weighted by the inverse of the column average. If the rank of $\mathbf{Z}$ is r, the GSVD of $\mathbf{Z}$ is then defined as:

$$\mathbf{Z} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^{\mathbf{T}} \tag{1}$$

where $\mathbf{\Lambda}$ is the $r \times r$ diagonal matrix of the singular values of $\mathbf{ZMZ^tN}$, $\mathbf{U}$ is the $n \times r$ matrix containing the r eigenvectors of $\mathbf{ZMZ^tN}$, and $\mathbf{V}$ is the $(k_1+s) \times r$ matrix of the r eigenvectors of $\mathbf{ZMZ^tN}$. The component scores for the n observations are computed as:

$$\mathbf{Y} = \mathbf{ZMV} \tag{2}$$

where $\mathbf{Y}$ is of the size $n \times r$. The first m ($m \leqslant r$) columns of $\mathbf{Y}$ are selected to capture a sufficiently high proportion of the variance in the dataset. When $m = r$ (all principal components are selected), 100 percent of variance is retained.

2.2 Hotelling T^2 chart based on PCA Mix

Using the first m principal component, the mean vector μ and covariance matrix $\mathbf{S}$ of $\mathbf{Y}$ are respectively a zero vector and a diagonal matrix with entries $\lambda_1 > \lambda_2 > ... > \lambda_m$ (the diagonal elements of $\mathbf{\Lambda}$). For the individual observation vector x_i, the Hotelling T^2 statistic (T_i^2) can be calculated by the following equation:

$$\begin{aligned} T_i^2 &= (y_i - \mu)^T \mathbf{S}^{-1} (y_i - \mu) \\ &= \sum_{a=1}^{m} \frac{{y_{i,a}}^2}{\lambda_a} \end{aligned} \tag{3}$$

The control limit of the T^2 chart can be calculated by using the techniques that will be detailed in the following sections.

2.2.1 Kernel density estimation

Kernel Density Estimation (KDE) is a nonparametric density estimation method that does not assume any specific shape for the density function. The idea is to estimate the distribution of the PCA Mix based T^2 statistic using the KDE

approach, and to determine the control limit as the $100(1-\alpha)^{th}$ percentile of the estimated kernel distribution. Based on the obtained T_i^2 values for $i = 1, ..., n$, the kernel density of the T^2 statistic is:

$$\widehat{f_h}(t) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{K}[\frac{(t - T_i^2)}{\hat{h}}] \tag{4}$$

where $\mathcal{K}$ is a kernel function and $\hat{h}$ is an estimated smoothing parameter that balances fit and smoothness. We adopt the mostly-commonly used function – the standard normal density function – as $\mathcal{K}$, and estimate $\hat{h}$ using the two-stage procedure proposed by Polansky and Baker (2000). The kernel distribution function of the T^2 statistic is then obtained as:

$$\widehat{F_h}(t) = \int_0^t \widehat{f_h}(T^2)dT^2 \tag{5}$$

The control limit of T^2 based on KDE, denoted as CL_{kernel}, satisfies the equation:

$$\widehat{F_h}(CL_{kernel}) = 1 - \alpha \tag{6}$$

2.2.2 Bootstrap

In addition to KDE, another distribution-free method that is based on the bootstrap resampling technique (Efron and Tibshirani 1986) can be used to determine the control limit of the PCA Mix-based T^2 statistic. Given T_i^2 with $i = 1, ..., n$ for n in-control observations, the control limit based on the bootstrap technique is obtained by performing the following steps:

a. Resample with replacement n values from $T_1^2, T_2^2, ..., T_n^2$ for B times to generate $T_1^{2(j)}, T_2^{2(j)}, ..., T_n^{2(j)}$ for $j = 1, 2, ..., B$.

b. For each bootstrap sample, we compute the bootstrap control limit as $CL_{bootstrap}^{(j)} = T_{100(1-\alpha)}^{2(j)}$ $(j = 1, ..., B)$, where $T_{100(1-\alpha)}^{2(j)}$ is the $100(1-\alpha)^{th}$ percentile of the j^{th} bootstrap sample.

c. The control limit is then estimated as the mean of $CL^{(j)}_{bootstrap}$ ($j = 1, ..., B$):

$$CL_{bootstrap} = \frac{1}{B}\sum_{j=1}^{B} CL^{(j)}_{bootstrap} \tag{7}$$

The number of bootstrap samples B should ideally be sufficiently large, and in the current study we set it to 1000. The choice of the number of bootstrap samples B is crucial as it may affect the estimation of the bootstrap control limit. Figure 1 illustrates the bootstrap control limits using different values of the number of bootstrap samples. By using R package "SimMultiCorrDat", we consider a simplified scenario where simulated data are described by one numerical variable (zero mean, unit variance, zero kurtosis, and skewness equal to five) and one binary variable (success probability equals to 0.7). These two variables are correlated with the target correlation coefficient being 0.4. For each value of B from 100 to 2000 in incremental steps of 100, we calculate the control limit 100 times. It can be seen from Figure 1 that as expected the variability of the bootstrap control limits is relatively greater when a small number of bootstrap samples are used and tends to decrease as the number of bootstrap samples increases. The value 1000 seems to be a reasonable choice for the number of bootstrap samples, as when more samples are used the variability does not decrease much but stabilizes.

Insert Figure 1 here

2.3 A procedure for applying PCA Mix-based T^2 chart in two phases

In the previous sections, we introduced the PCA Mix procedure to transform mixed data into numerical principal components, the T^2 chart based on the obtained principal components, and two methods (KDE and bootstrap) to calculate the control limit of the T^2 chart. We now present the complete

procedure that integrates these tools in order to monitor mixed data. In line with existing literature on using control charts (e.g., Woodall 2000; Ferrer 2007; Montgomery 2009), the procedure is implemented in two phases. Phase I, also known as the retrospective phase, aims to estimate the control chart parameters for in-control data. Treating the parameters as known, the aim of Phase II is subsequently to examine whether new observations from the process fall outside the estimated control limit. Specifically, as illustrated in Figure 2, the procedure is carried out as follows:

Insert Figure 2 here

Phase I: To estimate the control chart parameters

i Transform the $n \times (k_1 + k_2)$ in-control dataset $\mathbf{X}$ (k_1 numerical variables and k_2 categorical variables) into m principal component scores using Eq.(1) and Eq.(2).

ii Compute the Hotelling T^2 values using Eq.(3).

iii Determine the control limit CL_{kernel} using Eq.(4, 5 and 6), and $CL_{bootstrap}$ using the procedure detailed in Section 2.2.2.

In Phase I, we obtain a PCA Mix model, the in-control parameters including the mean vector (which is a zero vector) and covariance matrix $\mathbf{S}$ of the derived principal components, and the control limits based on KDE and bootstrap. These are saved to be used in Phase II monitoring.

Phase II: To monitor new observations

Let the new observation from the process on the (k_1+k_2) mixed variables be x_{new}.

iv Transform x_{new} to m principal component scores y_{new} using the parameters of the obtained PCA Mix model.

v Calculate the corresponding T^2_{new} statistic using the obtained covariance matrix S as below:

$$T^2_{new} = y^T_{new} S^{-1} y_{new} \tag{8}$$

vi Compare T^2_{new} with each of the determined control limits CL_{kernel} and $CL_{bootstrap}$.

If T^2_{new} exceeds the CL_{kernel} or (and) $CL_{bootstrap}$, x_{new} is considered to be an out-of-control signal for the corresponding control limit; otherwise, it is considered to be in control for the corresponding control limit.

In the introduction, we detailed the two indicators for evaluating the performance of control charts, which are the in-control average run length (ARL_0) and out-of-control average run length (ARL_1). In order to calculate the run length, Step iv, Step v, and Step vi of the procedure are repeated until the first out-of-control signal is detected. The number of in-control observations before this first signal, is recorded as the run length. When the new observation x_{new} in Phase II is generated from the same distribution as Phase I sample **X**, the process in Phase II is in control and accordingly the recorded run length is the in-control run length; otherwise an out-of-control run length is obtained. The complete procedure is repeated 10,000 times to obtain the average values of the in-control run length (ARL_0) and the out-of-control run length (ARL_1).

3 Simulation study

3.1 Initial setup of the simulations

As nonparametric methods, neither KDE nor bootstrap makes any assumptions about the distribution of the PCA Mix-based T^2. Instead, they purely rely on the Phase I sample **X** (i.e., the historical in-control data). As a result, the estimation of the control limit may be influenced by the characteristics of the Phase I sample. One key characteristic is the sample size. Therefore, our first simu-

lation study is conducted to evaluate the variability of the bootstrap and KDE control limits when different sizes of Phase I sample are used. We then conduct two more simulation studies to compare the performance of the two competing methods, based on ARL_0 and ARL_1 respectively.

Here, we present the initial setup that is applied to all three simulations. We generate n in-control observations that are described by six numerical variables and two binary variables. For the numerical variables, in addition to the typical assumption of multivariate normal distribution, we also consider a case when they deviate from a multivariate normal distribution. The numerical variables are therefore assumed to follow a multivariate skewed-normal distribution denoted by $\mathcal{SN}(\mu, \mathbf{\Sigma}, \lambda)$ (Azzalini 2005). Two different degrees of skewness ($\lambda = 0$ and 5) are considered, in order to observe their effects on the variability and performance of the estimated control limits.

Further, to simulate the observations we use $\mu = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ and the following covariance matrix:

$$\mathbf{\Sigma} = \begin{bmatrix} 1.00 & -0.30 & -0.01 & 0.11 & 0.121 & -0.06 \\ -0.30 & 1.00 & -0.04 & 0.56 & -0.12 & -0.33 \\ -0.01 & -0.04 & 1.00 & -0.02 & -0.03 & 0.09 \\ 0.11 & 0.56 & -0.02 & 1.00 & -0.12 & -0.35 \\ 0.12 & -0.12 & -0.03 & -0.12 & 1.00 & -0.29 \\ -0.06 & -0.33 & 0.09 & -0.35 & -0.29 & 1.00 \end{bmatrix}$$

This matrix is built based on the correlations among six numerical response quality indicators for 1725 in-control interviews. These in-control interviews were obtained based on real data collected during the eighth round of the European Social Survey in an ongoing project of the authors.

The variables are of unit variance, and are weakly to moderately correlated with each other. For the simulations, the R package "sn" (Azzalini 2019, https://cran.r-project.org/web/packages/sn/sn.pdf) is used to generate multivariate skewed-normal data.

The two categorical variables, following the authors' results on the in-control interviews, are generated based on two independent Bernoulli distributions with the success probability $pr_1 = 0.7$ and $pr_2 = 0.2$ respectively. Once an in-control dataset is generated, the next step is to transform the eight variables into principal components by using the PCA Mix procedure. This analysis is carried out by using the R package "PCAmixdata" (Chavent et al. 2014). The number of principal components that are used for calculating the T^2 statistic (m) is set to five and eight, in order to represent the cases of using only a few (five) principal components and all (eight) principal components respectively. To obtain the KDE control limits, package "kerdiest" (function "PBbw") is first used to estimate the bandwidth. Package "stats" (function "density") and "pracma" (function "trapz") are then employed to estimate the kernel distribution function of the PCA Mix based T^2 statistic and compute the $100(1-\alpha)$ percentile respectively. Moreover, the R package "boot" is used to obtain the bootstrap control limits.

The above introduced settings are used for all of the three simulation studies. They are summarized and marked in bold in Table 1. The specific settings for each study are also shown in the table. They are detailed in each of the following sections.

Insert Table 1 here

It is notable that when the first five principal components ($m = 5$) are used to compute the PCA Mix scores, on average around 78.5 percent of the total variability can be explained in the multivariate normal case ($\lambda = 0$) and around 81.5 percent in the multivariate skewed normal case ($\lambda = 5$). Detailed information is

presented in Table A1 in Appendix A.

3.2 Simulation study 1: the variability of the estimates of control limits

3.2.1 Specific setup

In this section, we investigate the variability of the estimated control limits based on bootstrap and KDE approaches using different sizes of Phase I samples. Six numerical variables are assumed to follow $\mathcal{SN}(\mathbf{0}, \boldsymbol{\Sigma}, \lambda)$ with $\lambda = 0$ and 5, as previously mentioned. For the two Bernoulli variables – starting from $(pr_1 = 0.7, pr_2 = 0.2)$ – in this simulation we also consider two other possibilities by varying one probability parameter each time to an extreme high (or low) value, first to $(pr_1 = 0.9, pr_2 = 0.2)$ and finally to $(pr_1 = 0.9, pr_2 = 0.05)$. Together with the two possibilities that we consider when selecting the principal components ($m = 5$ and 8) and the two degrees of skewness of the numerical variables ($\lambda = 0$ and 5), there are 12 scenarios in total $(3 \times 2 \times 2)$ in this simulation study.

For each scenario, we increase the size of the Phase I sample from 100 to 5000 in incremental steps of 100. For each number of observations, the control limits are calculated 100 times respectively using bootstrap and KDE.

3.2.2 Simulation results

The variability of the control limits, which are calculated 100 times for each number of observations, in terms of standard deviation, are illustrated in Figure 3. The scatter plots of the various control limits are presented in Figure B1 and Figure B2 in Appendix B. In general, for both bootstrap and KDE methods, across the 12 scenarios, the standard deviation of the control limits decreases – although at a different pace – as the Phase I sample size increases. The decreasing trend is more obvious when the sample size increases from 100 to 500, but gradually levels off after the size reaches 1000. The KDE control limits, however,

tend to have slightly more variability than the bootstrap control limits, especially when the size of the Phase I sample is smaller than 500.

Compared with the skewness of the numerical variables (λ) and the number of principal components (m), the success probabilities of the binary variables (pr_1 and pr_2) have the most significant influence on the variability of the estimated control limits. As the probability parameters become more extreme, for example at ($pr_1 = 0.9$, $pr_2 = 0.05$), there is still a considerable amount of variability in the estimates, even when the Phase I sample is as large as 5000. This is more straightforward if we set a reasonably small arbitrary value, say 1, as the target of the standard deviation of the control limits, and compare the required sample sizes across the scenarios. As shown in Table 2, the skewness (λ) and the number of principal components (m) do not have a clear impact on the required smallest sample size. However, as the shapes of the Bernoulli variables become more extreme, more Phase I observations are required in order to guarantee a certain level of variability.

Insert Figure 3 here

Insert Table 2 here

3.3 Simulation study 2: Comparisons of in-control average run length

3.3.1 Specific setup

In this section, we compare the performance of the control limits calculated based on the bootstrap and KDE approaches using the in-control average run length (ARL_0). As noted in the introduction, ARL_0 represents the mean number of observations before the occurrence of the first false alarm (from the beginning of monitoring in Phase II) when the process is in control. *Ideally,* ARL_0 takes the value of $\frac{1}{\alpha}$, where α is the pre-specified false alarm rate. However, in

real-life situations ARL_0 may always deviate from this theoretical value. The control limit that produces an actual ARL_0 value close to the desired ARL_0 value is considered as having the better performance.

In line with the initial setup, six numerical variables and two categorical variables are generated from $\mathcal{SN}(\mathbf{0}, \mathbf{\Sigma}, \lambda)$, with $\lambda = 0$ and 5, and Bernoulli distributions with $(pr_1 = 0.7, pr_2 = 0.2)$, respectively. Five and eight principal components are selected to construct the T^2 statistic, and to compute the control limits. Thus, four scenarios altogether are considered in this simulation study.

For each of the four scenarios, new observations are generated (from the same distributions used to generate the Phase I observations), and are monitored by the control limits determined by bootstrap and KDE. For each control limit, we record the run length when the first out-of-control signal is produced. The above procedure is repeated 10,000 times to calculate ARL_0. In this study, the false alarm rate α is set to 0.01, which corresponds to a desired ARL_0 value of 100. Instead of specifying the number of Phase I observations n beforehand – for example 1000, which is a typical choice in literature (e.g., Ahsan et al. 2018; Phaladiganon et al. 2011) – we consider a series of sample sizes ranging from 50 to 2000 (i.e., 50, 80, 100, 200, 300, 500, 800, 1000, 1500, 2000). The comparisons of the ARL_0 are made for each number of Phase I observations being used.

3.3.2 Simulation results

The actual ARL_0 values for the control limits using different numbers of Phase I observations are illustrated in Figure 4.

Insert Figure 4 here

The desired ARL_0 of 100 is represented by the red horizontal lines in Figure 4. The actual ARL_0 values determined by bootstrap and KDE are shown respectively by the solid lines with circle symbols and dashed lines with triangle symbols. Their values are presented in Table C1 in Appendix C. Based on the relative

positions of the two curves to the horizontal red lines, we can see that across the four scenarios, the actual ARL_0 values first move towards the desired ARL_0 as the Phase I sample size increases, but then gradually stabilize to a certain extent (close to or at 100).

Specifically, in the multivariate normal case ($\lambda = 0$), when five out of the eight principal components are selected ($m = 5$), the actual ARL_0 values obtained by KDE are already close to the desired value when the Phase I sample size ranges around 150 ($ARL_0 = 102.82$ when $n = 150$ and $ARL_0 = 98.96$ when $n = 200$). Bootstrap only yields a performance approximately equal to KDE when the sample size increases to 800. However, when the full set of principal components are selected ($m = 8$), both KDE and bootstrap control limits tend to generate more false alarms. This is shown by the fact that the obtained values of ARL_0 are all smaller than the desired value of 100.

To summarize, in both the multivariate normal case ($\lambda = 0$) and the skewed-normal case ($\lambda = 5$), the best "strategy" depends on the Phase I sample size: when it is smaller than around 800, the KDE control limit based on five principal components provides the best performance; when the Phase I sample size becomes equal to or larger than 800, the bootstrap control limit based on five principal components has an equal or even better performance than the KDE control limit based on five principal components (especially when the Phase I sample size is larger than 1500).

3.4 Simulation study 3: Comparisons of out-of-control average run length

3.4.1 Specific setup

In the previous section, we compared the performance of the two competing methods using the in-control ARL (ARL_0). The value of ARL_0 measures the false alarm rate, as the process in Phase II – from which the new observations

are generated – is actually *in control*. In this section, we impose shifts to the process means in Phase II, so that the process turns into an *out-of-control* state in Phase II. The performance of the competing methods can then be evaluated by counting how many observations each method takes to give the first out-of-control signal, which is termed as the out-of-control ARL (ARL_1). As we want to detect the process shifts as quickly as possible, the control limit that produces a smaller value for ARL_1 is considered better.

Again, we generate Phase I observations in the same manner as described in the initial setup: six numerical variables from $\mathcal{SN}(0, \mathbf{\Sigma}, \lambda)$, where $\lambda = 0$ and 5, and two Bernoulli variables with ($pr_1 = 0.7$, $pr_2 = 0.2$). The number of principal components retained is still considered as five and eight, to construct the T^2 statistic and determine the control limits based on bootstrap and KDE. In this study, the number of Phase I observations is fixed at 1000.

In Phase II, new observations are generated after adding a shift – 0.1 unit of standard deviations *each time* – to the process mean values μ, pr_1 and pr_2. Specifically, as the variance of each numerical variable is 1, and the variance of the two Bernoulli variables are 0.21 ($pr_1 \times (1 - pr_1) = 0.7 \times (1 - 0.7)$) and 0.16 ($pr_2 \times (1 - pr_2) = 0.2 \times (1 - 0.2)$) respectively, the shift added each time for each numerical variable is $\delta_\mu = 1 \times 0.1$, and for the Bernoulli variables, $\delta_{pr_1} = \sqrt{0.21} \times 0.1$ and $\delta_{pr_2} = \sqrt{0.16} \times 0.1$. The process mean values μ, pr_1 and pr_2 are shifted to $\mu + \delta_\mu$, $pr_1 - \delta_{pr_1}$, $pr_2 + \delta_{pr_2}$ in Phase II. The direction of the shift in each Bernoulli variable is determined by the size of the probability parameter with respect to 0.5: when the parameter is larger than 0.5 – as in the case of δ_{pr_1} ($pr_1 = 0.7$) – the direction of the shift is negative (a shift to smaller values); otherwise – as in the case of δ_{pr_2} ($pr_2 = 0.2$) – it is positive (a shift to larger values). The objective is to leave more space for the process mean to shift. New observations are generated until each control limit has the corresponding run length recorded. With the false alarm rate(α) remaining as 0.01, the simulations are repeated 10,000 times to calculate ARL_1.

3.4.2 Simulation results

In Figure 5, the ARL_1 values determined by the bootstrap and KDE methods are respectively shown by solid curves with circle symbols and dashed curves with triangle symbols. It should be remembered that here the number of Phase I observations is fixed at 1000. Examining the overall pattern of the curves across the four scenarios shows that the value of ARL_1 decreases as the shift to the process mean increases, suggesting that as the deviation from the in-control values of the process mean becomes larger, both methods require less time to trigger a signal. Meanwhile, for the same amount of shift and skewness, ARL_1 decreases when the number of principal components increases. This means that for both methods, the ability of the control limits to detect shifts in the process is enhanced by increasing the number of principal components. This observation is more clearly indicated in Table D1 in Appendix D, where the values of ARL_1 generated by bootstrap and KDE under four combinations of skewness (λ) and the number of principal components (m) are displayed. For example, when $\lambda = 0$ and $\delta_\mu = 0.3$, the ARL_1 generated by the bootstrap and KDE approaches using five principal components are 57.73, and 56.70, respectively, but using eight principal components, they are 30.87, and 30.63 respectively. It is notable that ARL_1 is recorded as zero when the first new observation is immediately detected as an out-of-control signal after shifts are imposed to the process means in Phase II.

The above observation (the enhanced ability of the control limits to detect shifts by increasing the number of principal components) seems to be contradictory with the results found in Section 3.3.2, where the control limits based on a larger number of principal components tend to generate more false alarms. However, the information provided is actually consistent. That is, we observe that with a larger number of principal components, the average run length tends to be smaller, in both in-control case (ARL_0) and out-of-control case (ARL_1). In the in-control case, smaller average run length means more

false alarms, whereas in the out-of-control case, this means an enhanced ability of detecting shifts. Our explanation is that, too many principal components will inflate the importance of noise and accordingly results in the problem of overfitting the PCA Mix model to recognize noise in the in-control data. Therefore we recommend an appropriate number of principal components instead of too small or too large.

To conclude, in both a normal ($\lambda = 0$) and skewed normal ($\lambda = 5$) case, with the full set of principal components selected, both KDE and bootstrap are quite sensitive with regard to detecting process shifts. For example, when $\lambda = 0$ and the process mean is shifted by one standard deviation ($\delta_\mu = 1.0$), the ARL_1 values obtained by the KDE and bootstrap methods are relatively small, both being only 0.97. This means that a process shift with a magnitude of one standard deviation is expected to be detected after around only one observation. The KDE control limits exhibit a very slightly better performance than the bootstrap control limits in detecting relatively small process shifts ($\delta_\mu \leq 0.8$). As the process shifts become larger, the bootstrap control limits are shown to perform equally as well as the KDE control limits.

Insert Figure 5 here

4 Conclusions

To be able to monitor a process that is described by both numerical and categorical variables, we present a procedure integrating a multivariate statistical tool, PCA Mix (Chavent et al. 2014), with the traditional Hotelling T^2 chart. As the underlying distribution of the PCA Mix-based T^2 is unknown, two nonparametric methods – bootstrap and kernel density estimation (KDE) – are employed to estimate the percentile of the distribution and to establish the control limits.

We focus on the case of six numerical variables following a multivariate

(skewed) normal distribution and two binary variables following Bernoulli distributions. Three simulation studies are conducted based on various combinations of different levels of study parameters, including Phase I sample size (n), the skewness of the numerical variables (λ), the success probabilities of the Bernoulli variables (pr_1, pr_2), and the number of selected principal components (m). Although nonparametric methods such as bootstrap and KDE do not involve assumptions about the underlying distribution of the PCA Mix-based T^2, the size of the Phase I sample used can affect the estimation of the control limits. Therefore, we first investigate the impact of the Phase I sample size on the control limits estimated using bootstrap and KDE approaches. In order to obtain insights into which method(s) works well in which situations, the second and third study respectively compare the performance of the two non-parametric control limits in terms of the in-control average run length (ARL_0) and the out-of-control average run length (ARL_1).

Our simulation results indicate that first, the nonparametric estimates of the control limits are highly unstable when the categorical variables are extremely imbalanced (i.e., the vast majority of samples take one of the categories of the variable). Including an imbalanced Bernoulli variable, such as with the success probability $pr = 0.05$, the estimates of the control limits do not seem to converge, even when a Phase I sample as large as 5000 is used. To ensure an acceptable level of variability, for example smaller than one standard deviation, the suggested minimum sample sizes in Phase I range from 300 ($pr_1 = 0.7$, $pr_2 = 0.2$) to 1100 ($pr_1 = 0.9$, $pr_2 = 0.2$). Meanwhile, with a Phase I sample size smaller than 500, the KDE control limits show more variability than the bootstrap control limits. Second, to generate the same (or nearly the same) number of false alarms as expected, it is ideal to use the KDE control limits based on five principal components when the Phase I sample size is relatively small ($n < 800$), and to use the bootstrap control limits based on five principal components when the Phase I sample size is relatively large ($n \geq 800$). This conclusion holds for both a

normal and a skewed normal case. Third, based on eight principal components, both the bootstrap and KDE methods are highly efficient in terms of detecting shifts in the process mean. Additionally, KDE seems to be slightly more sensitive with regard to detecting small shifts ($\delta_\mu \leq 0.8$). This conclusion is also supported in both a normal and a skewed normal case.

We see a trade-off in determining the number of principal components that should be retained. Using too many principal components tends to generate more false alarms, but using too few principal components may result in not detecting shifts in the process quickly enough. In this regard, an appropriate number of principal components instead of too small or too large is recommended. Moreover, the decision concerning the number of retained principal components should also be based on the particular application, taking into account the costs of a false alarm and the costs of missing a true alarm.

In addition to their performance (in terms of ARL_0 and ARL_1), there are certainly other factors that influence the choice between bootstrap or KDE when monitoring mixed variables. Using KDE involves correctly selecting the scale of smoothing (namely the bandwidth parameter) and a kernel function. It also takes effort to perform the numerical integration when calculating the area under the estimated density curve. By contrast, bootstrap resampling is more convenient from this perspective, as no parameter specifying process is required. The disadvantage of bootstrap lies in the fact that although it should ideally be sufficiently large (with 1000 being typical), finding the optimal number of bootstrap samples is not straightforward. A minor concern associated with a sufficiently large number of bootstrap samples is the required computational capacity and time. Based on an AMD Ryzen 9 3950X CPU, we applied KDE and bootstrap to a set of 1000 generated T^2 values to calculate the KDE control limits and bootstrap control limits respectively for a 10,000 times. The average computation time of KDE is 0.0834 (with a range between 0.0622 and 0.1764), and for bootstrap with the number of bootstrap samples being 1000, the average com-

putation time is 0.1282 (with a range between 0.1119 and 0.2339). In this regard, KDE seems to offer superior computational efficiency than bootstrap.

In general, for monitoring a mixture of numerical and categorical variables, the PCA Mix-based T^2 chart with nonparametric bootstrap and KDE approaches shows convincing performance in terms of ARL_0 and ARL_1, although caution should be taken with regard to the instability of the estimated control limits when highly imbalanced categorical variables are included. Future research would benefit from exploring other techniques to deal with the imbalanced categorical variables in a multivariate statistical process control framework. On a wider level, the encouraging results of the PCA Mix-based control charts on monitoring mixed data stimulus for future research to compare them with other existing schemes, such as the control charts based on local outlier factor method (Ning and Tsung 2012) and the ones based on standardized rank (Ding et al. 2016).

References

Ahsan, M., M. Mashuri, H. Kuswanto, D. D. Prastyo, and H. Khusna. 2018. "Multivariate control chart based on PCA mix for variable and attribute quality characteristics." *Production & Manufacturing Research* 6 (1): 364–384.

———. 2021. "Outlier detection using PCA mix based T2 control chart for continuous and categorical data." *Communications in Statistics - Simulation and Computation* 50 (5): 1496–1523. doi:10.1080/03610918.2019.1586921. eprint: https://doi.org/10.1080/03610918.2019.1586921. https://doi.org/10.1080/03610918.2019.1586921.

Aparisi, F., S. Garcıa-Bustos, and E. K. Epprecht. 2014. "Optimum multiple and multivariate Poisson statistical control charts." *Quality and Reliability Engineering International* 30 (2): 221–234.

Aslam, M., G. Srinivasa Rao, L. Ahmad, and C.-H. Jun. 2017. "A control chart for multivariate Poisson distribution using repetitive sampling." *Journal of Applied Statistics* 44 (1): 123–136.

Azzalini, A. 2019. *The R package sn: The Skew-Normal and Related Distributions such as the Skew-t (version 1.5-4)*. Università di Padova, Italia. http://azzalini.stat.unipd.it/SN.

Azzalini, A. 2005. "The skew-normal distribution and related multivariate families." *Scandinavian Journal of Statistics* 32 (2): 159–188.

Bersimis, S., A. Sgora, and S. Psarakis. 2018. "The application of multivariate statistical process monitoring in non-industrial processes." *Quality Technology & Quantitative Management* 15 (4): 526–549.

Bilson, J. F., A. Kumiega, and B. Van Vliet. 2010. "Trading model uncertainty and statistical process control." *The Journal of Trading* 5 (3): 39–50.

Bodnar, O., and W. Schmid. 2011. "CUSUM charts for monitoring the mean of a multivariate Gaussian process." *Journal of Statistical Planning and Inference* 141 (6): 2055–2070.

Capizzi, G. 2015. "Recent advances in process monitoring: Nonparametric and variable-selection methods for phase I and phase II." *Quality Engineering* 27 (1): 44–67.

Chavent, M., V. Kuentz-Simonet, A. Labenne, and J. Saracco. 2014. "Multivariate analysis of mixed data: The R Package PCAmixdata." *arXiv preprint arXiv:1411.4911*.

Chiu, J.-E., and T.-I. Kuo. 2007. "Attribute control chart for multivariate Poisson distribution." *Communications in Statistics-Theory and Methods* 37 (1): 146–158.

———. 2010. "Control charts for fraction nonconforming in a bivariate binomial process." *Journal of Applied Statistics* 37 (10): 1717–1728.

Crosier, R. B. 1988. "Multivariate generalizations of cumulative sum quality-control schemes." *Technometrics* 30 (3): 291–303.

Ding, D., F. Tsung, and J. Li. 2016. "Rank-based process control for mixed-type data." *IIE Transactions* 48 (7): 673–683.

Efron, B., and R. Tibshirani. 1986. "Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy." *Statistical Science* 1 (1): 54–77.

Ferrer, A. 2007. "Multivariate statistical process control based on principal component analysis (MSPC-PCA): Some reflections and a case study in an autobody assembly process." *Quality Engineering* 19 (4): 311–325.

Gunaratne, N. G. T., M. A. Abdollahian, S. Huda, and J. Yearwood. 2017. "Exponentially weighted control charts to monitor multivariate process variability for high dimensions." *International Journal of Production Research* 55 (17): 4948–4962.

Hotelling, H. 1947. "Multivariate quality control—illustrated by the air testing of sample bombsights." In *Techniques of Statistical Analysis,* edited by C. Eisenhart, M. Hastay, and W. Wallis, 111–184. New York: McGraw-Hill.

Jin, J., and G. Loosveldt. 2020. "Assessing Response Quality by Using Multivariate Control Charts for Numerical and Categorical Response Quality Indicators." Smaa012, *Journal of Survey Statistics and Methodology* (July). ISSN: 2325-0984. doi:10.1093/jssam/smaa012.

Jin, J., C. Vandenplas, and G. Loosveldt. 2019. "The Evaluation of Statistical Process Control Methods to Monitor Interview Duration During Survey Data Collection." *SAGE Open* 9 (2). doi:10.1177/2158244019854652.

Khusna, H., M. Mashuri, D. D. Prastyo, M. Ahsan, et al. 2018. "Multioutput least square SVR based multivariate EWMA control chart." In *Journal of Physics: Conference Series,* 1028:012221. 1. IOP Publishing.

Kovářık, M., and L. Sarga. 2014. “Implementing control charts to corporate financial management.” *WSEAS Transactions on Mathematics.*

Lowry, C. A., W. H. Woodall, C. W. Champ, and S. E. Rigdon. 1992. “A multivariate exponentially weighted moving average control chart.” *Technometrics* 34 (1): 46–53.

Montgomery, D. C. 2009. *Introduction to Statistical Quality Control.* John Wiley & Sons (New York).

Ning, X., and F. Tsung. 2012. “A density-based statistical process control scheme for high-dimensional and mixed-type observations.” *IIE transactions* 44 (4): 301–311.

Phaladiganon, P., S. B. Kim, V. C. Chen, J.-G. Baek, and S.-K. Park. 2011. “Bootstrap-based T 2 multivariate control charts.” *Communications in Statistics—Simulation and Computation* 40 (5): 645–662.

Pirhooshyaran, M., and S. T. A. Niaki. 2015. “A double-max MEWMA scheme for simultaneous monitoring and fault isolation of multivariate multistage auto-correlated processes based on novel reduced-dimension statistics.” *Journal of Process Control* 29:11–22.

Polansky, A. M., and E. R. Baker. 2000. “Multistage plug—in bandwidth selection for kernel distribution function estimates.” *Journal of Statistical Computation and Simulation* 65 (1-4): 63–80.

Sirkis, R., M. Jans, J. Dahlhamer, R. Gindi, and B. Duffey. 2011. “Using statistical process control to understand variation in computer-assisted personal interviewing data.” In *Joint Statistical Meetings,* 8:477–489. 2.

Sofikitou, E. M., and M. V. Koutras. 2020. "Multivariate Nonparametric Control Charts Based on Ordered Samples, Signs and Ranks." In *Distribution-Free Methods for Statistical Process Monitoring and Control,* edited by M. V. Koutras and I. S. Triantafyllou, 57–105. Cham: Springer International Publishing.

Thor, J., J. Lundberg, J. Ask, J. Olsson, C. Carli, K. P. Härenstam, and M. Brommels. 2007. "Application of statistical process control in healthcare improvement: systematic review." *BMJ Quality & Safety* 16 (5): 387–399. ISSN: 1475-3898. doi:10.1136/qshc.2006.022194.

Woodall, W. H. 2000. "Controversies and contradictions in statistical process control." *Journal of Quality Technology* 32 (4): 341–350.

A The percentage of the total variability explained by the first five principal components using the PCA Mix procedure

Insert Table A1 here

B Supplementary results of Simulation 1

Insert Figure B1 here

Insert Figure B2 here

C Supplementary results of Simulation 2

Insert Table C1 here

D Supplementary results of Simulation 3

Insert Table D1 here

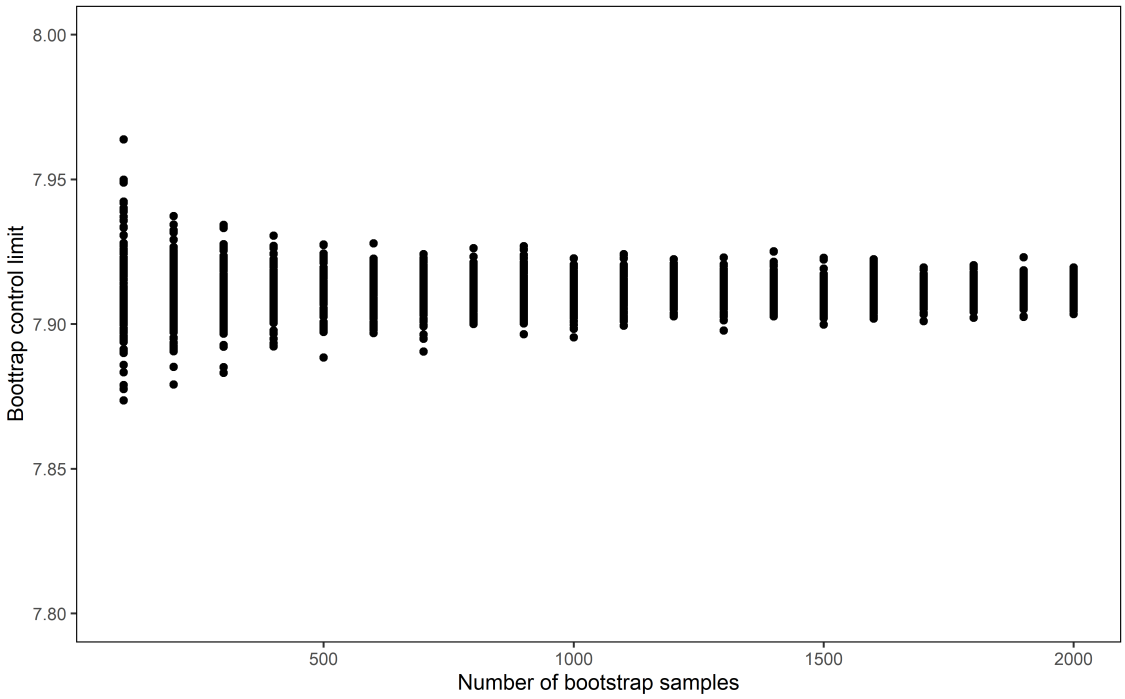


Figure 1: Bootstrap control limits (calculated 100 times) with different number of bootstrap samples.

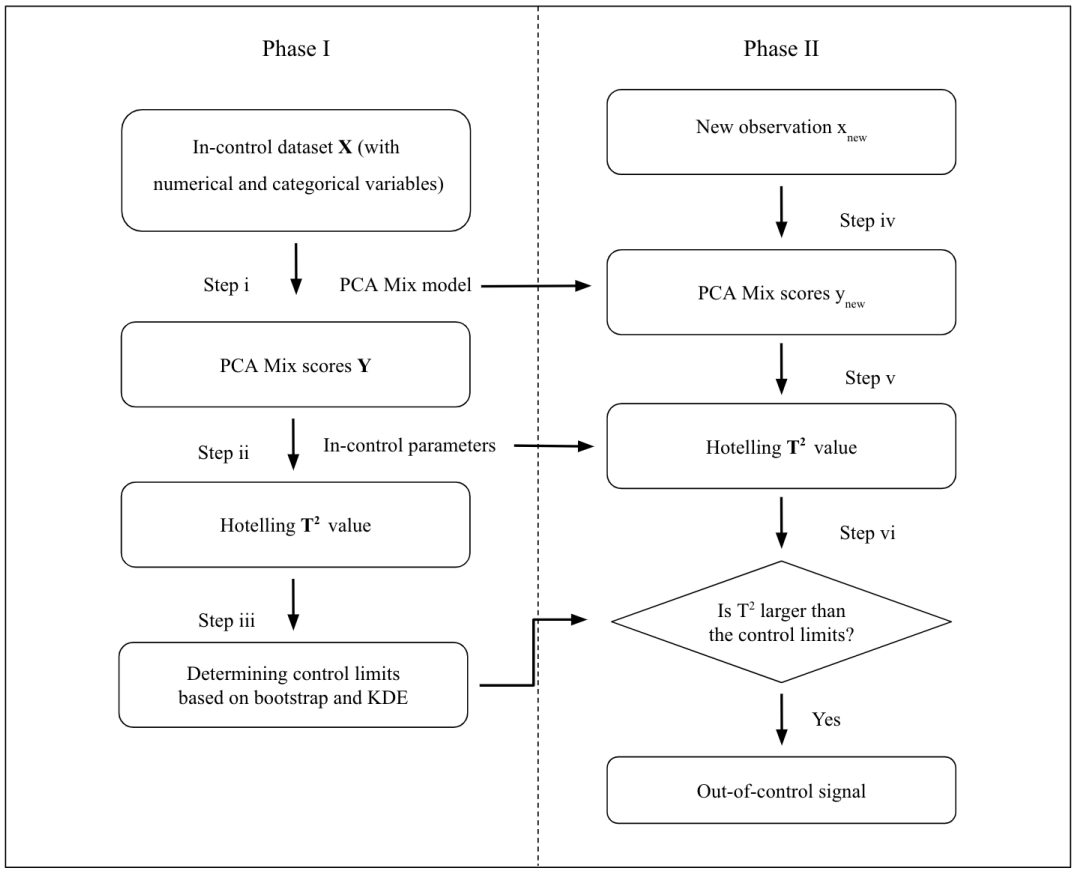


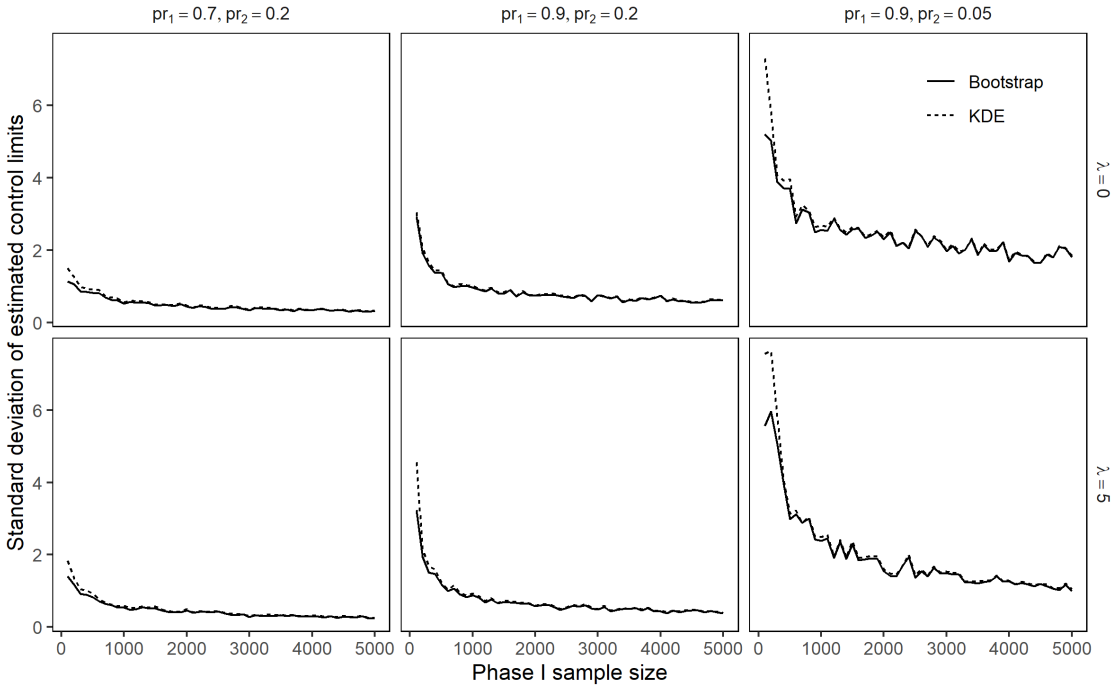
Figure 2: The PCA Mix based Hotelling T^2 chart in Phase I and Phase II

Table 1: Parameters and values used for each simulation study

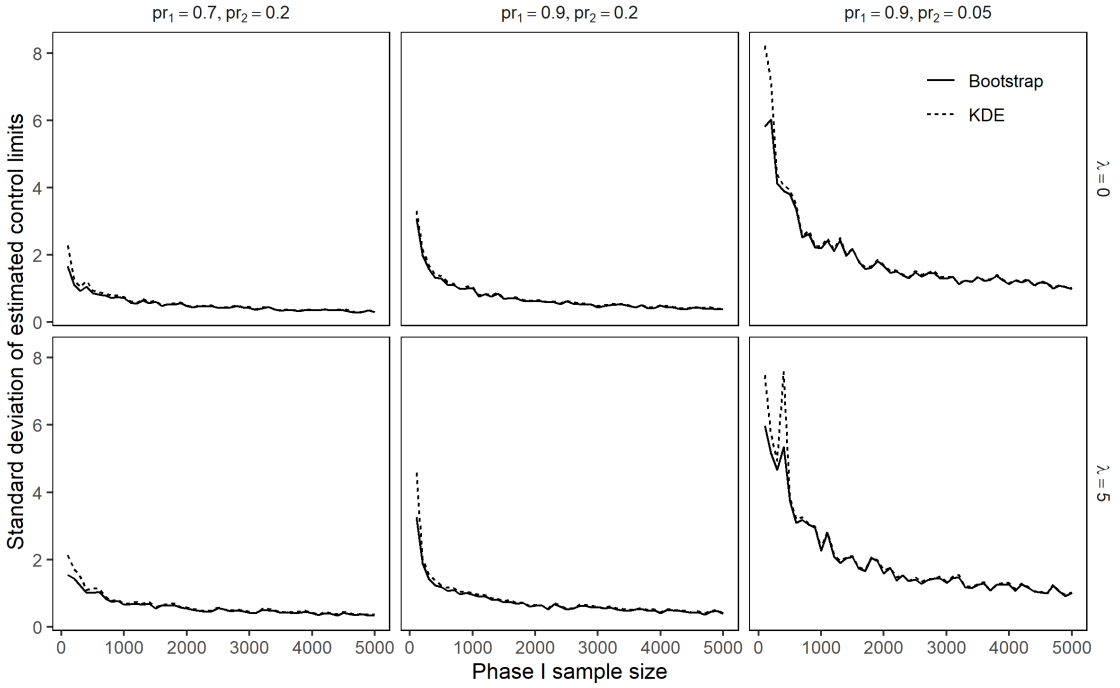
Parameter	Simulation 1	Simulation 2	Simulation 3
Phase I sample size (n)	100 to 5000 by 100	a series ranging from 50 to 2000	1000
Distribution properties			
skewness of numerical variables (λ)	**0; 5**	**0; 5**	**0; 5**
success probabilities of categorical variables (pr_1, pr_2)	(**0.7,0.2**) ; (0.9, 0.2); (0.9, 0.05)	(**0.7,0.2**)	(**0.7,0.2**)
number of principal components (m)	**5; 8**	**5; 8**	**5; 8**

Table 2: The required sample size to guarantee a smaller-than-1 standard deviation.

		$pr_1 = 0.7$ $pr_2 = 0.2$		$pr_1 = 0.9$ $pr_2 = 0.2$		$pr_1 = 0.9$ $pr_2 = 0.05$	
		Bootstrap	KDE	Bootstrap	KDE	Bootstrap	KDE
$\lambda = 0$	$m = 5$	300	300	1000	1100	>5000	>5000
	$m = 8$	500	500	1100	1100	5000	>5000
$\lambda = 5$	$m = 5$	300	500	800	800	5000	>5000
	$m = 8$	700	700	1000	1000	>5000	>5000



(a) Number of principal components (m) = 5



(b) Number of principal components (m) = 8

Figure 3: The standard deviation of control limits (calculated 100 times) with different numbers of Phase I observations.

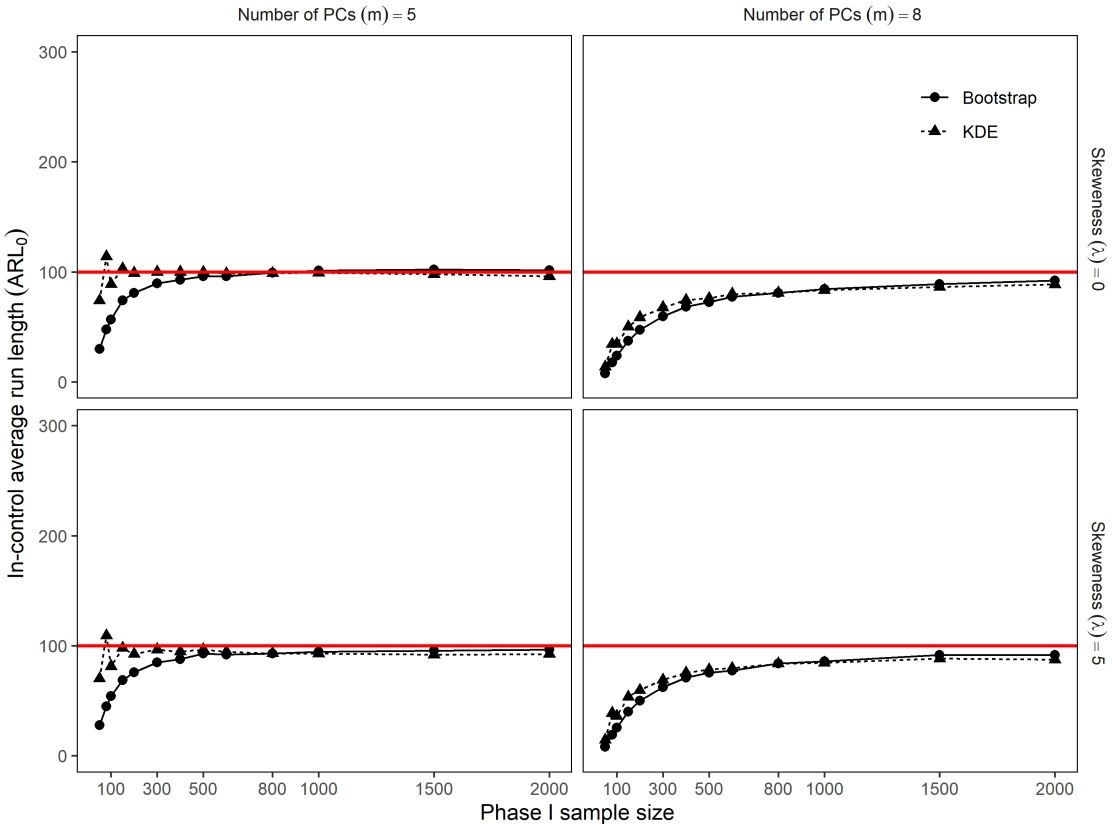


Figure 4: ARL_0 from control limits established by using bootstrap and KDE approaches from 10,000 replications

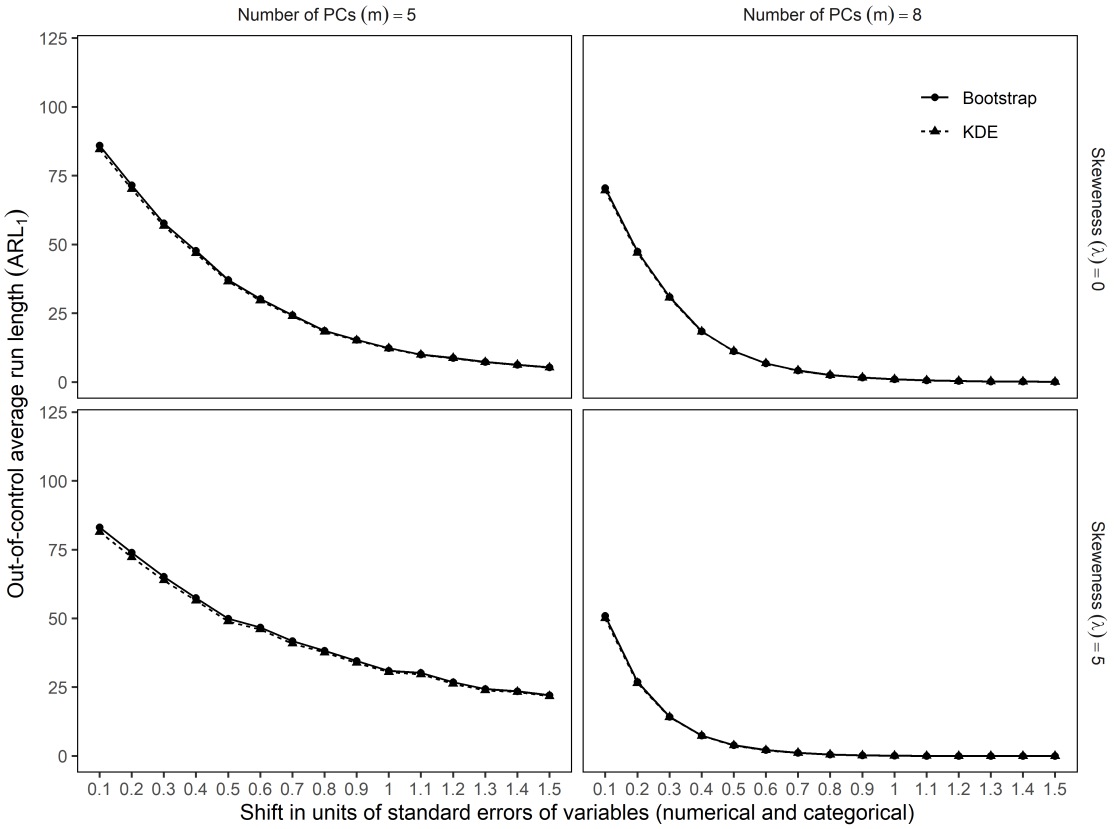
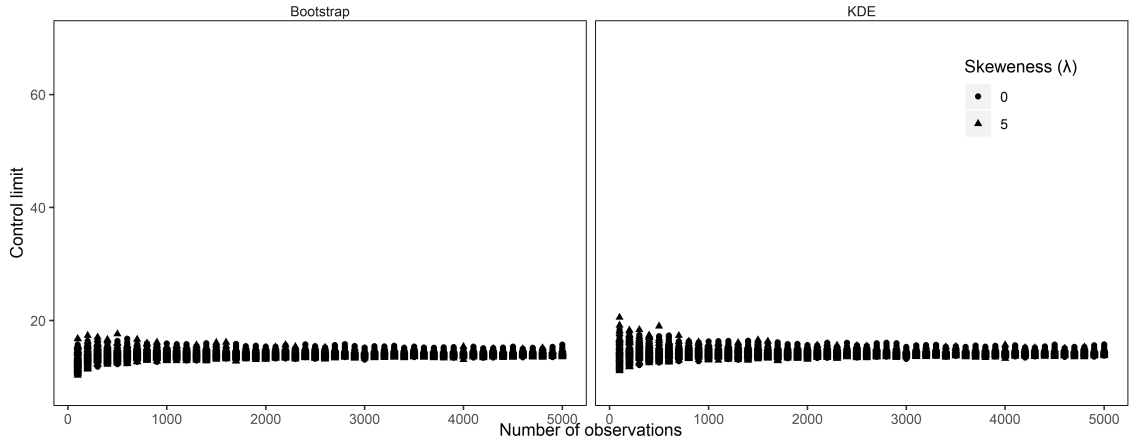


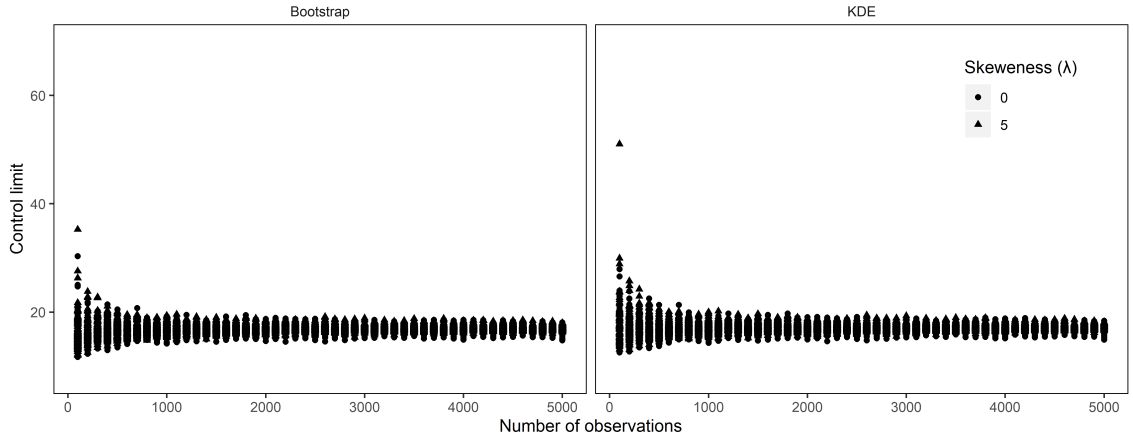
Figure 5: ARL_1 from control limits established from 10,000 replications by using the bootstrap and KDE approaches (Phase I sample size $n = 1000$)

Table A1: The average percentage of variability explained by the first five principal components from 10,000 replications ($n = 1000$)

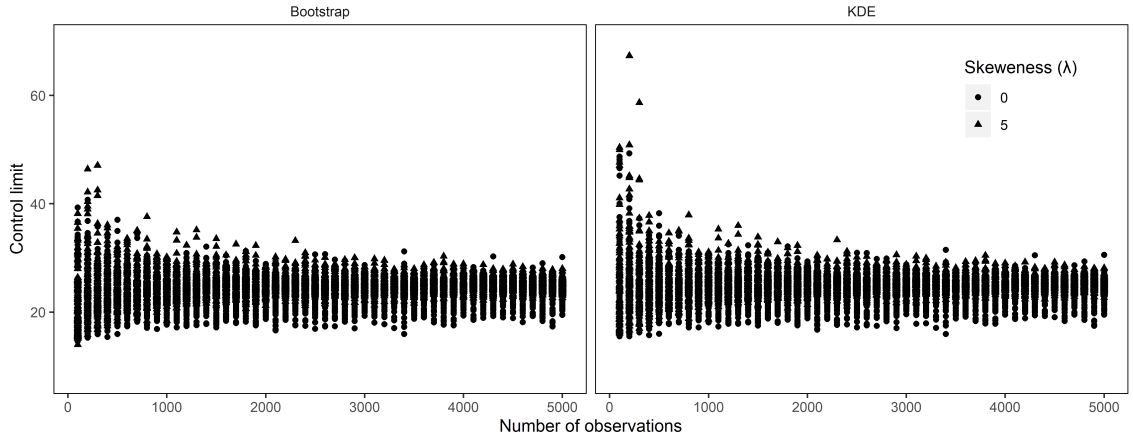
skewness of numerical variables (λ)	success probabilities of categorical variables (pr_1, pr_2)	percentage of variability
0	(0.7,0.2)	78.512
5	(0.7,0.2)	81.485
0	(0.9,0.2)	78.514
5	(0.9,0.2)	81.478
0	(0.9,0.05)	78.511
5	(0.9,0.05)	81.482



(a) $\mathrm{pr}_1 = 0.7$, $\mathrm{pr}_2 = 0.2$, $\mathrm{m} = 5$

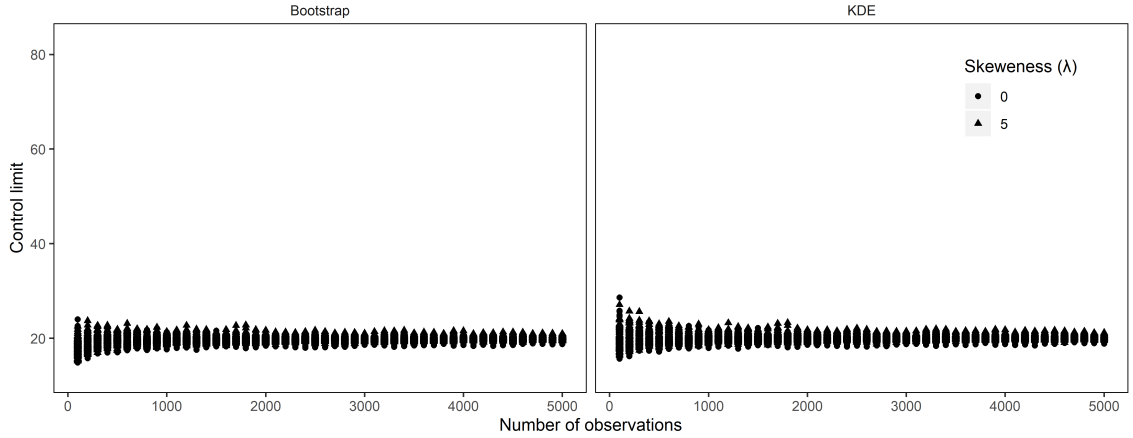


(b) $\mathrm{pr}_1 = 0.9$, $\mathrm{pr}_2 = 0.2$, $\mathrm{m} = 5$

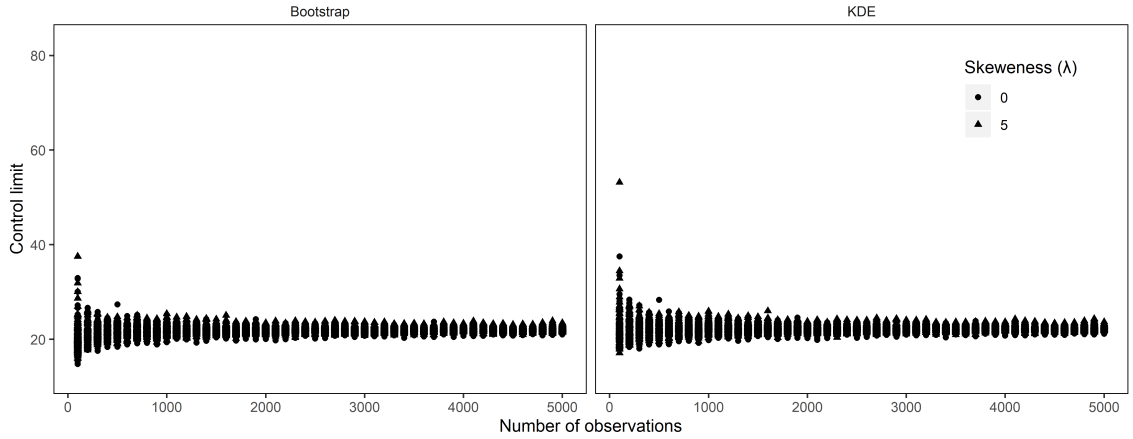


(c) $\mathrm{pr}_1 = 0.9$, $\mathrm{pr}_2 = 0.05$, $\mathrm{m} = 5$

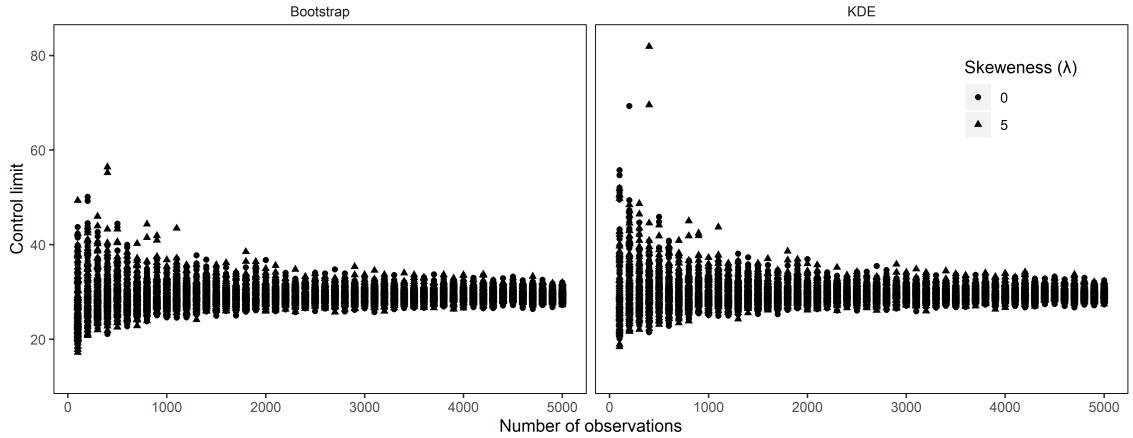
Figure B1: Control limits established by the bootstrap and KDE methods with different numbers of observations (control limits are calculated 100 times for each number of observation)



(a) $\mathrm{pr}_1 = 0.7$, $\mathrm{pr}_2 = 0.2$, $\mathrm{m} = 8$



(b) $\mathrm{pr}_1 = 0.9$, $\mathrm{pr}_2 = 0.2$, $\mathrm{m} = 8$



(c) $\mathrm{pr}_1 = 0.9$, $\mathrm{pr}_2 = 0.05$, $\mathrm{m} = 8$

Figure B2: Control limits established by the bootstrap and KDE methods with different numbers of observations (control limits are calculated 100 times for each number of observation)

Table C1: The ARL_0 for the control limits established by using bootstrap and KDE methods under various combinations of skewness parameter(λ), number of principal components (m), and Phase I sample size (n)

	$\lambda = 0$				$\lambda = 5$			
	$m = 5$		$m = 8$		$m = 5$		$m = 8$	
n	Bootstrap	KDE	Bootstrap	KDE	Bootstrap	KDE	Bootstrap	KDE
50	29.87	73.90	7.72	13.71	27.87	70.22	8.10	14.65
	(11.88)	(13.54)	(15.87)	(17.81)	(11.90)	(13.58)	(16.15)	(18.17)
80	47.62	113.83	17.58	34.27	44.98	109.25	19.04	38.90
	(12.72)	(14.14)	(17.09)	(18.70)	(12.70)	(14.11)	(17.48)	(19.27)
100	56.65	88.75	23.76	34.50	54.23	81.40	25.62	36.08
	(13.07)	(13.90)	(17.53)	(18.61)	(13.03)	(13.86)	(18.00)	(19.08)
150	74.14	102.82	37.30	50.06	68.91	97.91	40.01	53.50
	(13.57)	(14.16)	(18.19)	(18.92)	(13.51)	(14.09)	(18.77)	(19.51)
200	80.93	98.96	47.42	58.63	75.72	92.42	49.99	59.67
	(13.75)	(14.13)	(18.49)	(19.01)	(13.68)	(14.06)	(19.10)	(19.60)
300	89.52	99.87	59.46	67.65	84.97	96.56	62.41	68.97
	(13.99)	(14.21)	(18.80)	(19.12)	(13.90)	(14.11)	(19.45)	(19.74)
400	92.68	99.87	68.15	74.33	87.96	94.65	70.88	75.48
	(14.12)	(14.26)	(18.97)	(19.17)	(13.98)	(14.11)	(19.63)	(19.81)
500	96.09	99.80	72.54	75.99	93.01	96.58	75.43	78.42
	(14.19)	(14.27)	(19.07)	(19.20)	(14.03)	(14.11)	(19.74)	(19.85)
600	95.87	98.41	77.44	79.99	92.13	94.66	77.47	79.66
	(14.24)	(14.28)	(19.14)	(19.22)	(14.06)	(14.10)	(19.82)	(19.88)
800	99.11	98.78	80.79	80.90	92.85	93.10	83.91	83.77
	(14.28)	(14.27)	(19.23)	(19.24)	(14.10)	(14.09)	(19.90)	(19.89)
1000	101.02	99.24	84.24	83.28	94.68	92.86	85.73	84.49
	(14.33)	(14.30)	(19.29)	(19.26)	(14.12)	(14.08)	(19.97)	(19.91)
1500	102.14	97.93	88.93	86.26	95.59	91.95	91.80	88.38
	(14.37)	(14.28)	(19.35)	(19.27)	(14.14)	(14.05)	(20.05)	(19.94)
2000	101.40	95.99	92.18	88.58	96.46	92.43	91.54	87.37
	(14.39)	(14.28)	(19.40)	(19.29)	(14.16)	(14.05)	(20.09)	(19.95)

Note: $\alpha = 0.01$. The estimated control limits are presented in parenthesis.

Table D1: The ARL_1 for the control limits established by using bootstrap and KDE methods under various combinations of skewness parameter(λ), number of principal components (m), and shifts in the process mean (δ_μ, δ_{pr_1}, δ_{pr_2}) (Phase I sample size $n = 1000$)

shift			$\lambda = 0$				$\lambda = 5$			
			$m = 5$		$m = 8$		$m = 5$		$m = 8$	
δ_μ	δ_{pr_1}	δ_{pr_2}	Bootstrap	KDE	Bootstrap	KDE	Bootstrap	KDE	Bootstrap	KDE
0	0	0	101.02	99.24	84.24	83.28	94.68	92.86	85.73	84.49
0.1	0.0458	0.04	85.92	84.55	70.52	69.65	83.11	81.40	50.90	50.04
0.2	0.0917	0.08	71.52	70.16	47.43	46.89	73.87	72.22	26.87	26.49
0.3	0.1375	0.12	57.73	56.70	30.87	30.63	65.14	63.83	14.21	14.10
0.4	0.1833	0.16	47.70	46.78	18.42	18.29	57.43	56.48	7.39	7.32
0.5	0.2291	0.20	37.09	36.51	11.23	11.16	49.86	48.91	3.88	3.83
0.6	0.2750	0.24	30.18	29.59	6.80	6.78	46.72	45.99	2.12	2.09
0.7	0.3208	0.28	24.29	23.97	4.16	4.13	41.71	40.84	1.10	1.09
0.8	0.3666	0.32	18.62	18.24	2.51	2.49	38.24	37.61	0.51	0.50
0.9	0.4124	0.36	15.33	15.12	1.57	1.57	34.51	33.82	0.21	0.21
1.0	0.4583	0.40	12.33	12.17	0.97	0.97	30.96	30.47	0.05	0.05
1.1	0.5041	0.44	10.00	9.87	0.61	0.61	30.19	29.68	0.01	0.01
1.2	0.5499	0.48	8.76	8.66	0.37	0.37	26.76	26.23	0.00	0.00
1.3	0.5957	0.52	7.32	7.21	0.21	0.21	24.29	23.83	0.00	0.00
1.4	0.6416	0.56	6.28	6.20	0.12	0.12	23.54	23.26	0.00	0.00
1.5	0.6874	0.60	5.32	5.24	0.06	0.06	22.10	21.73	0.00	0.00
control limit			14.33	14.30	19.29	19.26	14.12	14.08	19.97	19.91

Note: $\alpha = 0.01$.